



HAL
open science

Études des stratégies et conception d'outils pour la production de supports illustratifs d'interaction

Axel Antoine

► **To cite this version:**

Axel Antoine. Études des stratégies et conception d'outils pour la production de supports illustratifs d'interaction. Interface homme-machine [cs.HC]. Université de Lille, 2021. Français. NNT : 2021LILUI012 . tel-03366786

HAL Id: tel-03366786

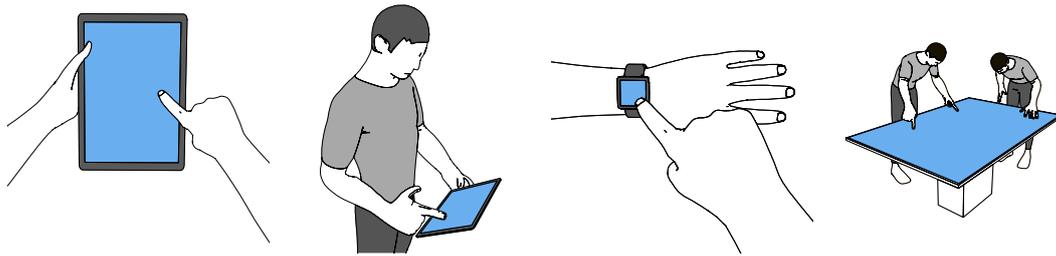
<https://theses.hal.science/tel-03366786>

Submitted on 5 Oct 2021

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

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

Études des stratégies et conception d'outils pour la production de supports illustratifs d'interaction



Axel ANTOINE

Thèse dirigée par Géry CASIEZ et encadrée par Sylvain MALACRIA

Thèse présentée et soutenue le 29 janvier 2021 pour obtenir le grade de Docteur de l'Université de Lille dans la spécialité Informatique.

PRÉSIDENT	: Christian DURIEZ	Inria
RAPPORTEURS	: Laurence NIGAY	Université de Grenoble
	Theophanis TSANDILAS	Université Paris-Saclay, Inria
EXAMINATEURS	: Emmanuel DUBOIS	Université de Toulouse
	Anne ROUDAUT	University of Bristol

Univeristé de Lille



Inria Lille - Nord Europe



Remerciements

Ma thèse, bien qu'étant une épreuve de longue haleine, s'est trouvée être une expérience enrichissante, finalement courte – j'avouerais ne pas toujours avoir eu ce discours – et surtout bien accompagnée.

Je tiens en premier lieu à remercier Géry Casiez, mon directeur, et Sylvain Malacria, mon encadrant, pour leur soutien sans failles. Cela fait maintenant sept ans que je travaille avec vous, et c'est toujours un pur plaisir rempli aussi de plusieurs succès : ForceEdge, TurboMouse et le stage chez Google en sont des bons exemples. Je tiens à vous dire que ma thèse n'aurait pu avoir lieu sans vous, car hormis le sujet passionnant que vous m'avez proposé, vous avez été tous les deux une raison supplémentaire – et nécessaire – à ma motivation. Je vous remercie pour votre supervision hors pair et le partage de votre expérience et de votre dévouement à la recherche même si je ne suis pas ce même chemin. C'est plus qu'un honneur de continuer à travailler deux ans de plus avec vous.

Je remercie grandement Nicolai Marquardt, qui a collaboré avec nous depuis Londres sur le projet Esquisse et la taxonomie. Ce fut un plaisir de travailler avec toi et de profiter de ta grande expérience dans le domaine de l'illustration.

Je remercie les rapporteurs de cette thèse, Laurence Nigay et Theophanis Tsandilas, les examinateurs, Emmanuel Dubois et Anne Roudaut, et le président, Christian Duriez, d'avoir consacré du temps à lire, évaluer et assister à ma soutenance de thèse.

Je remercie mes proches, ma famille, mes amis, d'avoir été là, de m'avoir donné du courage dans les moments difficiles (via la consommation – non-optionnelle – de rhum). Promis, je paie bientôt des tournées.

Enfin, je te remercie toi, Salomé, qui même à l'autre bout de la France, m'a porté avec ses petits bras de future chirurgienne. Je te ferais remarquer néanmoins, que je suis docteur avant toi, alors que je n'ai jamais touché un bistouri.

Études des stratégies et conception d'outils pour la production de supports illustratifs d'interaction

FR L'utilisation de systèmes interactifs est une situation complexe dans laquelle un ou plusieurs utilisateurs interagissent avec un ou plusieurs systèmes. Pour bien comprendre l'interaction, il est idéal d'en faire l'expérience ou, à défaut, de pouvoir la voir en action, ce qui n'est pas toujours possible. En effet, le principal médium de communication de la recherche en Interaction Homme-Machine (IHM) reste les articles scientifiques, généralement au format PDF et, de ce fait, de plus en plus de figures illustrant de nouvelles interactions sont produites. L'objectif de cette thèse est d'étudier les stratégies et les outils permettant de créer de telles figures. Tout d'abord, nous enrichissons les connaissances théoriques actuelles sur la création de figures illustrant l'interaction, en proposant une nouvelle taxonomie des attributs conceptuels et visuels qui composent 795 figures extraites d'articles de recherche en IHM, et un ensemble de stratégies d'illustration identifiées via l'analyse des relations entre ces attributs. Ensuite, nous nous intéressons à l'aspect pratique de la création de ces figures et soutenons l'hypothèse que l'agencement de scènes 3D est une alternative intéressante pour les créer. Nous présentons alors *Esquisse*, un outil proposant des effets de rendu et des techniques d'interaction spécifiques qui permettent à des utilisateurs novices et experts de produire efficacement des figures illustrant l'interaction. Enfin, cette thèse présente une étude sur les rotations d'objets en 3D, une tâche qui a été qualifiée de complexe lors de l'agencement de scènes 3D. Nous étudions les techniques de sphères virtuelles de contrôle et montrons que certains facteurs, comme la taille de la sphère, ont une influence sur le comportement des utilisateurs. Les travaux présentés dans cette thèse fournissent ainsi un premier cadre théorique et pratique pour informer les futures recherches et outils portant sur la production de figures illustrant des scénarios interactifs.

Study of strategies and design of tools for creating illustrations representing interaction scenarios

EN Interaction with interactive systems is a complex situation in which one or more users interact with one or more systems. To fully understand an interaction, it is ideal to experience it or, at least, to be able to see it in action, which is not always possible. However, the main medium to communicate research in Human-Computer Interaction (HCI) remains research papers, generally in PDF format, and as a result, more and more figures illustrating new interactions are being produced. The objective of this thesis is to study the strategies and tools used to create such figures. First of all, we enrich the current theoretical knowledge on the creation of figures illustrating the interaction by proposing a new taxonomy of the conceptual and visual attributes that compose 795 figures extracted from research articles in HCI. We also present a set of illustration strategies identified through the analysis of the relationships between these attributes. Then, we look at the practical aspect of the creation of these figures and support the hypothesis that 3D scenes staging is an interesting alternative to create figures representing interaction. We then present *Esquisse*, a tool offering specific rendering effects and interaction techniques that allow both novice and expert users to efficiently produce figures illustrating the interaction. Finally, this thesis presents a study on 3D object rotations, a task that has been described as complex when staging 3D scenes. We study the techniques of virtual control spheres and show that some factors, such as the size of the sphere, have an influence on the behavior of users during rotation tasks. The work presented in this thesis thus provides a first theoretical and practical framework to inform future research and tools for the production of figures illustrating interactive scenarios.

Table des matières

Remerciements	2
Résumés	3
Table des matières	5
Introduction	8
1 Axes de recherche	10
2 Contributions de la thèse	11
3 Organisation du document	12
1 État de l’art	14
Introduction	15
1 Illustration de scénarios interactifs	15
1.1 Taxonomies existantes sur la représentation visuelle	16
1.2 Lien entre les taxonomies et les théories sur l’illustration	17
2 Création de figures à base de tracés	24
2.1 Méthodes basées sur des modèles	24
2.2 Méthodes basées sur la mise en scène	27
Conclusion	29
2 Étude des stratégies d’illustration de scénarios interactifs	30
Introduction	31
1 Taxonomie des scénarios interactifs	33
1.1 Construction du jeu de données	33
1.2 Procédure de codage	34
1.3 Statistiques globales	37
2 Stratégies d’illustration de scénarios interactifs	42
2.1 Procédure	42
2.2 Stratégies de structure	44
2.3 Stratégies d’interaction	54

2.4	Stratégies de brevets	64
3	Outils pour le codage, la visualisation et l’exploration de la taxonomie	66
3.1	Outil de codage	66
3.2	Outil de visualisation	67
3.3	Outil d’exploration	68
	Discussion	69
	Discussion sur la méthodologie	69
	Limitations de l’étude	71
	Conclusion	71
3	Esquisse : un outil de production d’illustrations reposant sur de l’agencement de scènes 3D	73
	Introduction	74
1	Enquête sur les processus de création de figures à base de tracés	76
2	Esquisse : processus créatif et interaction	78
2.1	Choisir son modèle de scène et ses objets	80
2.2	Ajout d’interfaces utilisateur	80
2.3	Faciliter la manipulation des objets 3D	81
2.4	Représentation du mouvement	86
3	<i>Esquisse</i> : algorithme de rendu vectoriel	89
3.1	État de l’art sur les techniques de rendu non photoréaliste vectoriel	89
3.2	Algorithme de rendu non-photorealiste vectoriel	90
4	Étude : illustration de techniques d’interaction	95
4.1	Méthode d’expérimentation	95
4.2	Résultats	97
	Discussion	100
	Limitations dues à l’utilisation de Blender	100
	Modifier les pratiques actuelles	100
	Conclusion	101
4	Étude des techniques de rotation d’objets 3D	102
	Introduction	103
1	État de l’art sur les techniques de rotation 3D via des sphères virtuelles	105
1.1	Interaction sur la sphère virtuelle	105
1.2	Interaction en dehors de la sphère virtuelle	106
1.3	Comparaisons des sphères virtuelles	107
2	Étude : influence de la taille de la sphère de contrôle	108
2.1	Méthode d’expérimentation et dispositif	108
2.2	Procédure d’expérimentation	109
2.3	Résultats	111

Discussion	116
Discussion des résultats	116
Stratégies de rotation	117
Conclusion	119
Conclusion	120
Travaux futurs	122
Pistes d'amélioration de la taxonomie	122
Pistes d'amélioration d' <i>Esquisse</i>	122
Pistes d'amélioration des sphères virtuelles	124
Vers un système idéal de création de figures?	124
Annexes	126
Annexe A Exemples d'illustrations extraites des articles de recherche	127
Annexe B Version détaillée de la taxonomie	129
Liste des acronymes	133
Glossaire	134
Bibliographie	135

L'interaction avec les systèmes interactifs est une situation complexe dans laquelle un ou plusieurs utilisateurs interagissent avec un ou plusieurs systèmes interactifs. Nous appelons une telle situation un scénario interactif. Lorsque l'on s'intéresse à un scénario interactif, la façon idéale de le comprendre est bien entendu d'en faire l'expérience ou, à défaut, de pouvoir le voir en action. Malheureusement, interagir avec un système interactif n'est pas toujours possible, par exemple si le système repose sur un système d'exploitation incompatible, ou bien s'il nécessite du matériel spécifique et difficilement disponible.

De plus, le principal medium de communication de la recherche en Interaction Homme-Machine (IHM) reste les articles de recherche, qui utilisent des formats non dynamiques, à savoir des PDF ou des pages web HTML statiques. Comme « *une image vaut mille mots* », les auteurs de ces articles conçoivent et représentent souvent des illustrations graphiques statiques pour expliquer ces scénarios interactifs, la plupart du temps associées à des légendes, pour former une figure complète [45]. En outre, un nombre croissant de revues recommandent d'ajouter des résumés graphiques [66] résumant les soumissions (par exemple, les revues Elsevier suggèrent d'inclure des résumés visuels [42]). Ces résumés visuels doivent donner une « *représentation claire du travail décrit dans l'article* », « *résumer le contenu* » pour aider les lecteurs à « *comprendre le message principal de l'article* », « *encourager la navigation* », « *promouvoir* » et « *identifier* » les articles de recherche¹.

Les illustrations ne sont pas seulement un moyen utile de décrire des scénarios interactifs dans les documents de recherche. Les illustrations sont largement utilisées pour apporter un support visuel aux présentations, idées ou systèmes développés et peuvent avoir plusieurs formes. Dans la communauté IHM, les illustrations sont souvent utilisées en tant que supports visuels pour décrire et représenter de nouvelles techniques d'interaction, des systèmes et des données de manière statique (par exemple à travers des images, des diagrammes ou des graphiques) et de manière dynamique (par exemple à travers des figures animées ou interactives et des vidéos). À titre d'exemple, les 951 articles des actes

1. « *A graphical abstract should allow readers to quickly gain an understanding of the main take-home message of the paper and is intended to encourage browsing, promote interdisciplinary scholarship, and help readers identify more quickly which papers are most relevant to their research interests.* » [42]

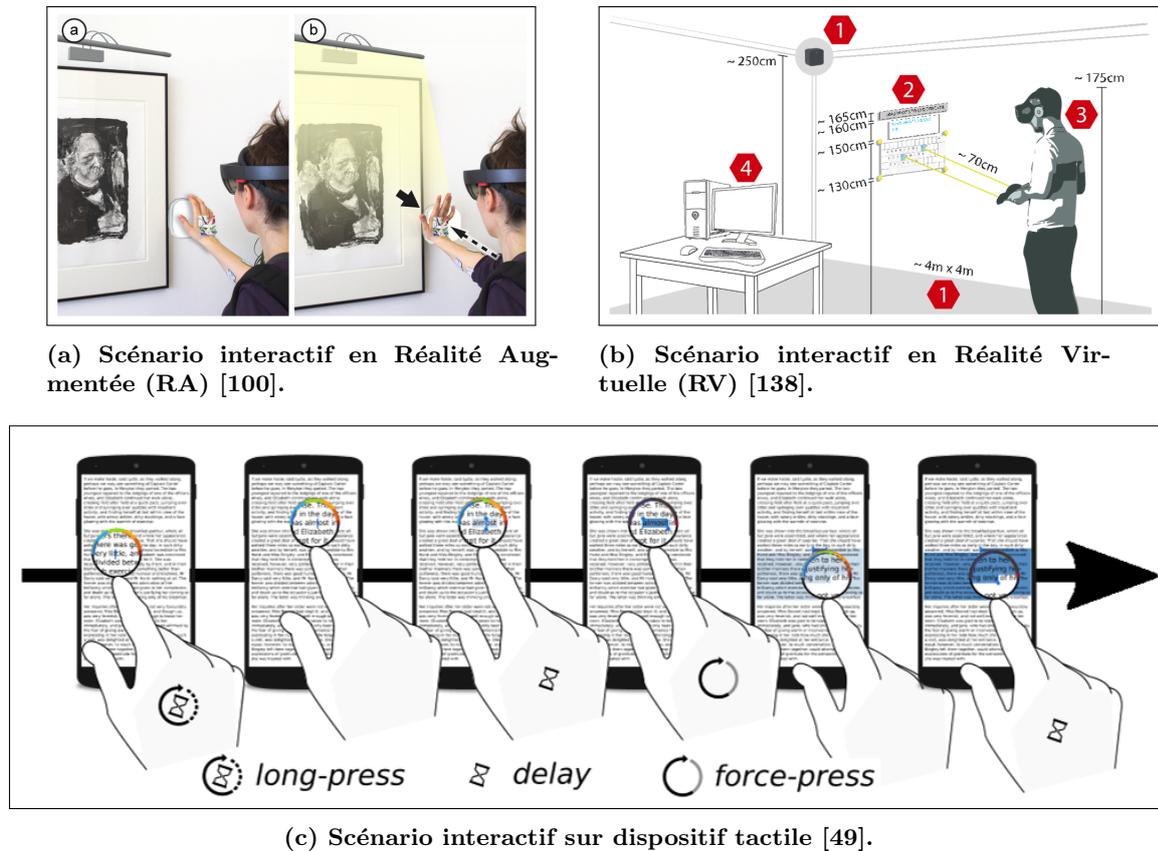


Figure 1 – Exemples d’illustrations de scénarios interactifs. (a) Photographies illustrant l’enchaînement d’actions avec un interrupteur d’une utilisatrice en RA. (b) Représentation du pointage à distance d’un utilisateur sur un clavier via des contrôleurs en RV. (c) Dessin d’une séquence d’action de la main d’un utilisateur sur un dispositif tactile dans une tâche de sélection de texte.

2018 des conférences ACM Conference on Human Factors in Computing Systems (CHI), ACM Symposium on User Interface Software and Technology (UIST), ACM Conference on Computer Supported Cooperative Work (CSCW) et ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp) ont inclus un total de 7243 illustrations dont 1893 graphiques, 617 diagrammes, 1774 tables et 2841 autres figures (dessins ou photographies). Si le contenu d’une illustration peut varier, sa structure quant à elle, peut tout aussi varier, en utilisant par exemple une ou plusieurs vignettes et être complétée par des titres et des étiquettes. Les légendes jouent également un rôle prépondérant, à tel point qu’elles peuvent également être considérées comme faisant partie intégrante de l’illustration [45]. Bien qu’elles soient omniprésentes et très utilisées, on sait peu de choses sur les différentes approches et les outils permettant de créer de telles illustrations statiques. Cette thèse se focalise principalement sur une catégorie de ces illustrations, celles qui illustrent des scénarios interactifs, c’est-à-dire, les figures qui illustrent une situation complexe dans laquelle un ou plusieurs utilisateurs interagissent avec un ou plusieurs systèmes. La

figure 1 montre trois exemples d'illustrations de scénarios interactifs. Trois axes principaux de recherche sont étudiés et présentés dans cette thèse.

1 Axes de recherche

Dans le premier axe de recherche, nous nous intéressons à l'aspect théorique de la création des figures qui illustrent un scénario interactif, c'est-à-dire, à l'analyse du contenu de ces illustrations et des stratégies utilisées par les chercheurs pour illustrer l'interaction. Cette analyse requiert une méthodologie spécifique pour construire un jeu de données reposant sur un codage rigoureux de 795 figures extraites d'articles de recherche selon les éléments de conception représentés, c'est-à-dire, les attributs conceptuels et visuels des illustrations utilisés pour coder l'interaction [109]. Nous proposons ainsi une nouvelle taxonomie qui classe les éléments de conception identifiés dans les figures statiques illustrant l'interaction. Celle-ci permet d'avoir un aperçu des styles et des techniques utilisés par la communauté en IHM dans les figures pour représenter des scénarios interactifs. En analysant les corrélations entre les codes identifiés dans les figures, c'est-à-dire, les caractéristiques visuelles utilisées et le type de scénario interactif représenté, nous proposons ensuite un ensemble de stratégies potentielles utilisées par les utilisateurs pour produire ces illustrations.

Dans le deuxième axe de recherche, nous nous intéressons à l'aspect pratique de la création des figures illustrant un scénario interactif, c'est-à-dire, aux outils actuels utilisés pour créer ces figures, et plus précisément, les figures à base de tracés. Les figures à base de tracés sont des figures illustratives statiques créées pour capturer l'essence d'une situation, supprimer les détails non nécessaires en limitant la représentation graphique aux contours les plus importants des objets et personnes représentés. Alors que prendre une photographie ou une capture d'écran est une tâche qui peut être effectuée en quelques secondes en cliquant sur un bouton physique ou virtuel, le processus de création d'une figure à base de tracés est quant à lui beaucoup plus complexe. De manière générale, nous montrons que les techniques actuelles pour produire de telles figures reposent sur un processus de création long et fastidieux (*p.ex.* le décalquage de photographies). Nous proposons alors un nouvel outil, *Esquisse*, qui repose sur l'agencement de scènes 3D pour générer des figures illustrant un scénario interactif. En s'appuyant sur les stratégies d'illustration identifiées pour produire ces figures dans le premier axe de recherche, *Esquisse* facilite le processus créatif en proposant des effets de rendu et des techniques d'interaction qui permettent à des utilisateurs novices et experts de produire efficacement des figures à base de tracés.

Enfin, dans le troisième axe de recherche, nous nous intéressons à la rotation des objets en 3D. En effet, nous observons dans le deuxième axe de recherche, au travers de l'étude que nous avons menée pour évaluer *Esquisse*, que certains participants ont rencontré des difficultés à manipuler des objets 3D, et plus précisément, à contrôler leur

rotation dans l'espace. Dans cet axe de recherche, nous établissons tout d'abord la liste des techniques de rotation actuellement utilisées sur ordinateur de bureau pour orienter des objets. Nous présentons ensuite une étude contrôlée, réalisée auprès de 24 participants dans le but d'identifier les différents facteurs qui peuvent influencer les performances et le comportement des utilisateurs dans une tâche de rotation d'un objet 3D. En analysant le comportement des utilisateurs lors de ces tâches, il est alors possible d'en déduire des stratégies de rotation, et de proposer une nouvelle technique d'interaction permettant de faciliter la rotation d'objets 3D.

2 Contributions de la thèse

Cette thèse propose les contributions suivantes :

1. L'établissement d'une taxonomie des éléments de conception qui composent les figures représentant des scénarios interactifs, établie sur un large jeu de figures extraites de conférences en IHM. Cette taxonomie unifie les taxonomies actuelles de l'état de l'art, les étend, et synthétise un large éventail de techniques et d'approches parmi les nombreux scénarios interactifs illustrés dans la communauté IHM.
2. Une étude exploratoire sur les stratégies d'illustration utilisées pour représenter un scénario interactif. Cette étude propose un ensemble de stratégies de *structure* et *d'interaction* identifiées dans les figures existantes et utilisées pour représenter des aspects spécifiques de l'interaction.
3. Trois outils logiciels open-source : une application pour faciliter le long et fastidieux processus de codage de figures et deux outils en ligne pour explorer la taxonomie que nous avons créée et identifier les stratégies. Ces outils peuvent être utilisés pour reproduire notre méthodologie avec des jeux de données différents ou avec des objectifs différents, afin d'extraire de nouvelles stratégies.
4. La conception d'*Esquisse*, un outil se basant sur l'agencement de scènes 3D pour générer des figures à base de tracés vectoriels. L'agencement de scènes s'appuie sur des techniques d'interaction spécifiques qui facilitent la manipulation des objets 3D, tels qu'un système d'ancrage ou de téléportation de caméra.
5. Le développement d'une extension open-source pour Blender qui intègre les fonctionnalités d'*Esquisse*.
6. Un algorithme de rendu innovant, qui s'appuie sur des techniques de rendu de l'état de l'art, permettant de générer des figures vectorielles à partir de scènes 3D illustrant des contextes d'interaction. Cet algorithme permet de rendre des scènes qui illustrent un contenu dynamique grâce au rendu de flèches ou d'un effet stroboscopique.
7. Une étude utilisateur sur l'évaluation de l'utilisation de l'agencement de scènes 3D comme alternative aux outils actuels pour la production de figures à base de tracés

vectoriels, pour illustrer l'interaction. Cette étude compare l'utilisation d'*Esquisse* et des outils actuels lors d'une production de figures.

8. Une étude préliminaire sur la manipulation d'objets 3D via une technique de sphère virtuelle de contrôle et l'influence de la taille de la sphère de contrôle et de la zone d'interaction sur les performances et le comportement des utilisateurs.

3 Organisation du document

Le document est structuré en quatre chapitres.

Le chapitre 1 présente deux parties d'état de l'art. La première partie présente les taxonomies existantes sur la représentation de scénarios interactifs en IHM [34, 110] et leurs limitations, telles que la représentation d'un sous-ensemble des interactions en IHM. Nous présentons aussi d'autres travaux qui ont classifié les interactions sans pour autant étudier l'impact sur leur représentation [103, 148] ou ne sont pas liés au domaine de l'IHM [5, 119, 142]. Enfin, nous lions les taxonomies existantes avec des travaux d'autres domaines comme la bande dessinée [111] ou les manuels d'instructions [5], afin d'en extraire les propriétés et stratégies qui pourraient être pertinentes dans le cadre de l'illustration de scénarios interactifs. La seconde partie présente les techniques et outils existants qui peuvent être utilisés pour créer des figures qui illustrent des scénarios interactifs et plus particulièrement les figures à base de tracés. Ces techniques sont principalement regroupées en deux catégories, celles basées sur l'utilisation d'agencement et celles basées sur la démonstration. De manière générale, nous montrons que les techniques actuelles pour produire des figures à base de tracés sont soit longues et fastidieuses, ou alors non adaptées à la représentation d'un scénario interactif.

Le chapitre 2 présente notre travail d'élaboration d'une nouvelle taxonomie de la représentation de l'interaction. Cette taxonomie a notamment été élaborée à l'aide d'un long travail d'annotation de 795 figures extraites de la littérature. Nous présentons ensuite une analyse de ces figures annotées, basée sur l'exploration de ce jeu de données créé grâce à un outil disponible en ligne. Cette analyse permet d'identifier un ensemble de stratégies utilisées par les chercheurs de la communauté IHM pour illustrer des scénarios interactifs.

Le chapitre 3 défend l'hypothèse qu'une technique d'agencement de scènes 3D est une alternative intéressante pour faciliter la production de figures à base de tracés illustrant des scénarios interactifs. Ce chapitre présente *Esquisse*, un outil open-source implémentant un processus d'agencement 3D pour produire des tracés vectoriels et distribué sous forme d'extension pour le logiciel de modélisation 3D Blender. Nous discutons des techniques d'interaction proposées pour faciliter la manipulation d'objets 3D. Nous présentons également le pipeline de rendu vectoriel non-photométrique innovant basé sur des techniques de l'état de l'art. Nous présentons enfin l'évaluation d'*Esquisse* auprès d'un groupe de chercheurs en IHM.

Le chapitre 4 présente une étude préliminaire sur l'influence de la taille de la sphère virtuelle de contrôle sur les performances des utilisateurs lors de la manipulation de la rotation d'un objet 3D via une technique de *trackball* sur ordinateur de bureau. Nous montrons que bien que la taille de la sphère n'a pas un impact direct sur les performances des utilisateurs, l'analyse des mouvements du curseur de souris indique que les participants emploient différentes stratégies pour manipuler les objets 3D.

Sommaire

Introduction	15
1 Illustration de scénarios interactifs	15
1.1 Taxonomies existantes sur la représentation visuelle	16
1.2 Lien entre les taxonomies et les théories sur l'illustration	17
2 Création de figures à base de tracés	24
2.1 Méthodes basées sur des modèles	24
2.2 Méthodes basées sur la mise en scène	27
Conclusion	29

Introduction

Les deux premiers axes de recherche étudiés dans cette thèse s'intéressent à la création de figures illustratives d'interaction, d'abord d'un point de vue théorique, puis d'un point de vue pratique. Nous présentons dans ce chapitre l'état de l'art¹ lié à ces deux axes de recherche et nous mettons en perspective leurs limites.

La première partie de l'état de l'art s'intéresse à l'aspect théorique de la création de figures, notamment, nous cherchons à savoir s'il existe dans la littérature des connaissances déjà définies, des concepts ou des techniques d'illustration permettant de représenter l'interaction dans des figures statiques. Ces connaissances peuvent permettre alors aux chercheurs novices qui souhaitent créer leur figure ou aux chercheurs expérimentés qui souhaitent s'améliorer, d'avoir un aperçu des stratégies d'illustration existantes pour produire des figures illustrant un scénario interactif, et de s'en servir comme exemple. En outre, nous présentons dans cette partie un certain nombre de connaissances qui existent dans la littérature sur la manière de structurer d'une illustration, l'apparence des figures, la représentation de la dynamique (*p.ex.* représenter le mouvement) ou encore leurs annotations.

Le deuxième partie de l'état l'art, s'intéresse quant à elle à l'aspect pratique de la création de figures. Sachant qu'il existe des stratégies d'illustration spécifiques, identifiées dans la première partie, pour représenter l'interaction, il convient ici d'étudier les techniques et outils utilisés pour produire ces illustrations. Notamment, nous cherchons à savoir s'il existe des techniques spécifiques permettant de représenter facilement les éléments qui constituent les scénarios interactifs, à savoir les utilisateurs, les dispositifs système utilisés et les interactions entre ces deux types d'entités. Nous présentons dans cette partie un ensemble de techniques existantes, plus ou moins adaptées à la création d'illustrations représentant un scénario interactif, regroupées en deux catégories : les techniques basées sur la démonstration et celles basées sur des modèles.

1 Illustration de scénarios interactifs

La taxonomie présentée dans cette thèse s'appuie sur des taxonomies existantes qui classifient des éléments de conception visuels, c'est-à-dire, les attributs conceptuels et visuels des illustrations utilisés pour représenter l'interaction [109]. Ces travaux, tous avec des motivations différentes et spécifiques, explorent soit l'impact des illustrations sur le comportement des apprenants [45], le développement de nouveaux produits [119] ou encore les représentations gestuelles [109]. Dans cette section, nous résumons les taxonomies visuelles existantes les plus pertinentes, et leur relation avec les travaux présentés dans cette thèse.

1. L'état de l'art correspondant au troisième axe de recherche sera pour sa part discuté au chapitre 4.

1.1 Taxonomies existantes sur la représentation visuelle

Pei *et al.* ont proposé une taxonomie des représentations visuelles dans le contexte du développement de produits [119]. Ils discutent notamment du fait que dans le domaine de la conception de produits, le style de dessin peut être différent en fonction du cycle de vie d'un produit illustré ou de l'objectif véhiculé par l'illustration. En utilisant un jeu de données construit par des designers et des ingénieurs, leur taxonomie divise les représentations visuelles du domaine du design en quatre catégories principales : les *sketchs*, les *dessins*, les *modèles* et les *prototypes*. Bien que cette taxonomie se concentre sur des représentations spécifiques au domaine du design, nous pensons que leur travail est intéressant dans notre contexte d'illustrations de scénarios interactifs, car les systèmes interactifs peuvent également nécessiter différents types de représentations, par exemple si elles décrivent un prototype ou un produit fini. De plus, Pei *et al.* ont proposé une sous-catégorie spécifique dans leur taxonomie : *Représentations de design visuel en 2D* → *Dessins* → *Dessins de design industriel* → *Scénario et storyboard*, où l'objectif de ces illustrations est de « suggérer une interaction entre l'utilisateur et le produit et de représenter son utilisation dans le contexte d'artefacts, de personnes et de relations »².

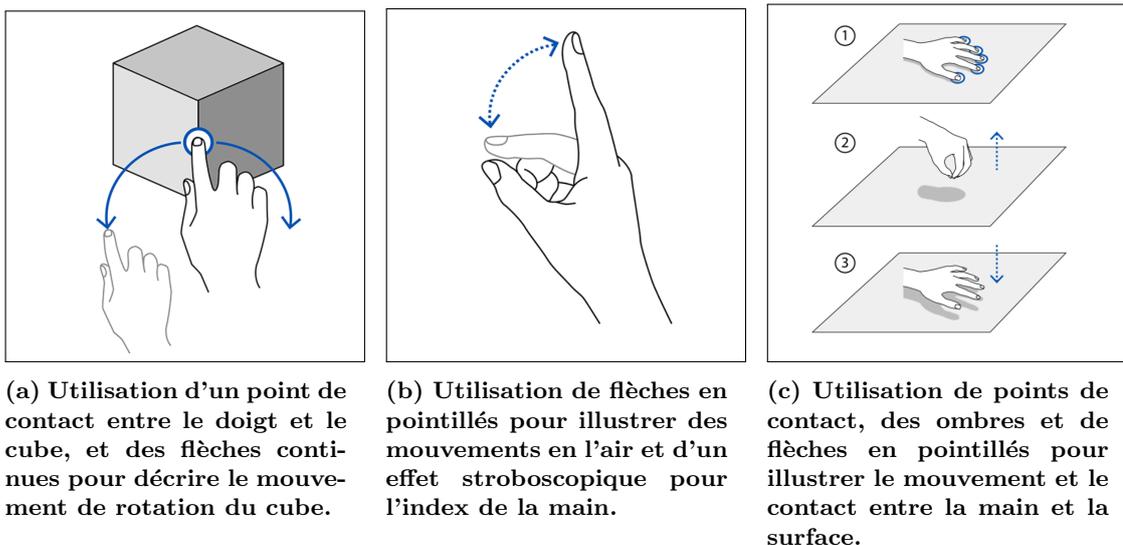


Figure 1.1 – Exemples de recommandations proposées par McAweeney *et al.* [109] pour illustrer des interactions gestuelles (extraits des figures 8, 9 et 10 de la publication).

Une autre taxonomie peut être trouvée dans le travail de McAweeney *et al.* [109]. Ils ont d'abord mené une étude d'élicitation avec des concepteurs et des chercheurs pour comprendre les processus utilisés pour créer des représentations gestuelles. Leur étude a montré qu'il n'existait pas de lignes directrices pour aider les chercheurs à concevoir

2. « *Scenarios and storyboards are 2D visual design representations to suggest user and product interaction, and to portray its use in the context of artefacts, people and relationships.* » [119, p23]

des représentations gestuelles. Ensuite, les auteurs ont constitué un jeu de données à partir de 30 articles publiés dans des conférences de l'ACM (CHI, ISS et Ubicomp), comprenant des figures à base de tracés, des photographies, des rendus par ordinateur, des lignes abstraites, des points et des textes. En utilisant les informations issues de leur élicitation, ils ont codé le jeu de données, en se basant sur une approche empirique qui vise à développer une théorie à partir de données ou d'observations empiriques lorsque la littérature existante est limitée [89, section 11.4.1.1]. Ils ont alors identifié et classé les éléments de conception qui composent les figures en six dimensions regroupées en deux catégories principales : *structure* et *détails*. Les dimensions de type *structure* (*perspective*, *vignettage* et *coloration*) sont décrites comme des dimensions « nécessaires pour concevoir toute représentation »³ tandis que les dimensions de type *détails* (*contexte corporel*, *contexte environnemental* et *éléments du geste*) sont décrites comme des dimensions « optionnelles » couramment utilisées pour « étendre/enrichir la représentation structurelle du geste »⁴. Ces catégories sont utilisées comme éléments graphiques pour composer les cinq aspects d'une représentation gestuelle : la *temporalité*, la *position*, la *posture*, le *mouvement* et le *toucher*. Cette taxonomie fournit des conseils et des lignes directrices précises axées sur l'illustration du mouvement (tel que « utiliser une flèche en forme de ligne pointillée pour illustrer le mouvement en l'air », cf. figure 1.1) mais se limite aux représentations de gestes.

Enfin, bien qu'éloignée des figures illustrant des scénarios d'interaction, la taxonomie de Fleming [45] peut s'avérer intéressante dans l'établissement d'une nouvelle taxonomie. Fleming a analysé 787 illustrations extraites de 40 manuels scolaires de quatre disciplines : anglais, histoire, mathématiques et sciences [45]. L'objectif de cette étude était d'observer l'impact du contenu des illustrations sur la capacité d'apprentissage des apprenants. Pour ce faire, il a établi une taxonomie des illustrations pédagogiques en attribuant à chaque illustration des attributs regroupés selon 11 échelles pour un total de 107 catégories. Les échelles sont les suivantes : la *zone*, le *vignettage*, la *forme*, la *position*, les *éléments*, la *chrominance*, la *non-chrominance*, le *style d'encodage*, le *moyen d'encodage*, le *niveau d'information* et l'*unification*. Ces échelles ont permis d'éclairer notre schéma de classification présenté dans la suite du document.

1.2 Lien entre les taxonomies et les théories sur l'illustration

Comme expliqué ci-dessus, il n'existe malheureusement que très peu de taxonomies spécifiquement axées sur les illustrations. Elles sont limitées, que ce soit en termes de contexte d'interaction, de types de figures ou de domaines d'intérêt. Néanmoins, ces taxonomies contiennent des informations sur lesquelles nous pouvons nous baser pour former

3. « We grouped Perspective, Frame, and Color as structural elements that are necessary for the design of any representation. » [109, p4]

4. « Gesture elements (as well as Body context and Environmental context) is grouped as details of a representation, and is added to structure to further communicate aspects of the gesture. » [109, p4]

une taxonomie d'illustration de scénarios interactifs plus large et plus globale. Dans les sections suivantes, nous relierons les taxonomies présentées dans la section précédente à d'autres ouvrages de la littérature, généralement des livres ou des articles de design, afin d'analyser comment les éléments de conception présents dans les figures peuvent soutenir la représentation de scénarios interactifs. L'analyse suivante est présentée dans un ordre qui reflète le processus de création d'une figure, c'est-à-dire de sa structure à son apparence.

1.2.1 Vignettage des illustrations de scénarios interactifs

Une première caractéristique des illustrations réside dans la façon dont elles sont structurées, par exemple le nombre d'images utilisées et le nombre ou la disposition des vignettes avec lesquelles elles sont composées.

Typiquement, Bill Buxton a écrit dans *Sketching User Experiences* [26] que « *les dessins pour l'expérience et la conception d'interaction [...] diffèrent des dessins conventionnels, car ils doivent composer avec le temps, le phrasé et le ressenti* »⁵. Ainsi, la représentation d'un scénario d'interaction doit respecter différents attributs comme par exemple gérer la *temporalité* (puisque l'interaction peut être continue), utiliser un « *vocabulaire clair* » et des « *gestes distincts* » pour ne représenter que les éléments essentiels qui composent l'interaction. Enfin, il recommande de créer une figure avec des « *détails minimaux* » et une « *précision appropriée* » pour minimiser la perturbation du lecteur. Ces attributs se rapportent à des concepts similaires mentionnés dans la taxonomie des représentations gestuelles de McAweeney *et al.* [109] où les auteurs mentionnent cinq aspects importants lors de l'illustration d'un geste : la *temporalité*, la *position*, la *posture*, le *mouvement* et le *toucher*.

De même, alors que les dimensions de type *détails* de McAweeney *et al.* [109] décrivent l'ajout de détails précis pour comprendre les différents aspects du geste représenté, les dimensions de type *structure* sont nécessaires pour concevoir tout type d'illustration d'interaction. Plus précisément, la dimension *vignettage* (aussi présentée dans la taxonomie de Fleming [45]) mentionne que les illustrations peuvent avoir une ou plusieurs vignettes pour structurer la représentation et illustrer une progression dans le temps [109].

Une notion similaire de « *vignette* » est abordée dans *Understanding Comics*, l'ouvrage phare de Scott McCloud qui explore les aspects formels de la bande dessinée et leur vocabulaire fondamental [111]. Plus précisément, McCloud aborde la notion de « *panneaux* » qui peuvent être utilisés pour « *fracturer à la fois le temps et l'espace pour offrir un rythme irrégulier de moments non connectés* »⁶ et de « *caniveaux* » (*c.-à-d.* espaces blancs entre les panneaux) comme un moyen de relier ces moments. Ces transitions entre les différentes

5. « *One thing that we know is that sketches for experience and interaction design will likely differ from conventional sketching since they have to deal with time, phrasing, and feel—all attributes of the overall user experience.* » [26, p135]

6. « *Comics panels fracture both time and space, offering a jagged, staccato rhythm of unconnected moments.* » [111, p67]

vignettes d'un *storyboard* de bande dessinée peuvent prendre différentes formes (p.70) [111], dont certaines peuvent être pertinentes dans une illustration de scénario interactif. McCloud introduit notamment le *moment-à-moment* comme une transition de vignettes entre deux moments très proches dans le temps (qui, dans le contexte de l'IHM, pourrait être pertinent pour illustrer les transitions entre deux captures d'un système), « *action-à-action* » comme une transition de vignettes entre deux actions proches dans le temps d'un même sujet (*p.ex.* illustrer un utilisateur pointant sur une cible spécifique), *sujet-à-sujet* comme une transition de vignettes entre différents sujets (*p.ex.* montrer un transfert de données entre deux utilisateurs qui collaborent) et *aspect-à-aspect* comme une transition de vignettes entre différents aspects du contexte représenté (*p.ex.* donner différents points de vue des utilisateurs d'un système).

Greenberg *et al.* ont également identifié l'importance des vignettes et de leurs transitions dans la description d'un contexte interactif, car ce sont elles qui « déclenchent un changement d'état » et peuvent être « déclenchées par une ou plusieurs actions de l'utilisateur » [51]. Un bon exemple est celui des transitions des interfaces utilisateur, qui sont souvent représentées sur différents supports en IHM pour illustrer l'interaction étape par étape avec une interface.

Enfin, une dimension de structure intéressante est la *perspective*, une dimension codant le point de vue de la représentation. McAweeney *et al.* ont énuméré cinq points de vue différents : 1^{ère} *personne*, 3^{ème} *personne*, *miroir*, *vue aérienne* et *vue de côté* [109]. Notez que les notions de *vignette* et de *perspective* sont dépendantes, car la perspective englobe le point de vue utilisé dans une vignette.

1.2.2 Illustrer des interactions dynamiques dans des images statiques

La section précédente montre que le choix d'une structure de figure spécifique dépend de ce qui est représenté. Plus précisément, le nombre de vignettes et les points de vue peuvent différer si le contexte interactif représenté est statique ou dynamique. Cependant, des techniques alternatives pour illustrer des séquences dynamiques ont été identifiées dans la taxonomie de McAweeney *et al.* [109]. Par exemple, des flèches peuvent être utilisées pour représenter la trajectoire ou le chemin d'un mouvement en utilisant un style de ligne continue pour un mouvement sur une surface ou en pointillés pour une gestuelle en l'air. Des « *points de touché* » [109] peuvent être utilisés pour illustrer les contacts physiques entre les doigts et une surface. Enfin, un effet « *fantôme* » [109] (*c.-à-d.* un effet stroboscopique) peut être utilisé pour illustrer un mouvement où les états/positions précédents sont dessinés de différentes couleurs ou suivant différents styles de lignes. McAweeney *et al.* rapportent une utilisation de 44% de flèches, 25% d'effets stroboscopiques, 12% de lignes pointillées dans les 150 représentations de gestes produites par leurs participants dans leur deuxième étude. Il est intéressant de noter que les flèches et les effets stroboscopiques ont déjà été identifiés comme les meilleurs candidats pour illustrer la dynamique par Cutting

dans son livre sur la représentation du mouvement dans les images statiques [37], et des outils logiciels aidant à la production de figures permettent également d'illustrer ces deux méthodes [35].

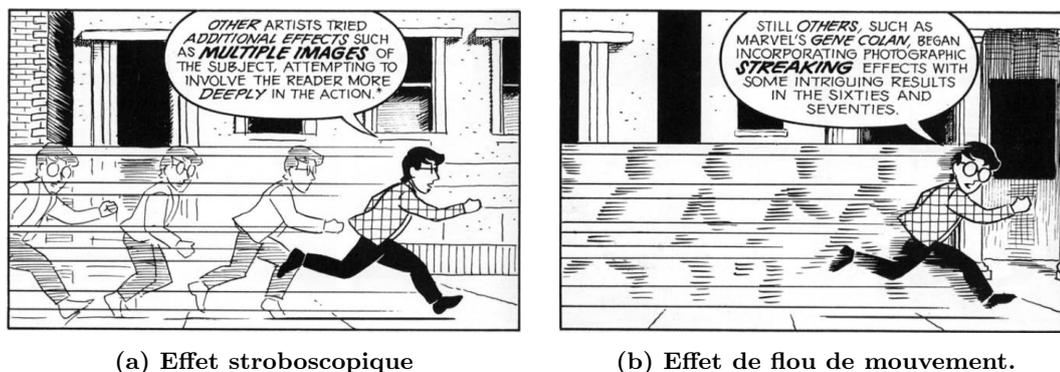


Figure 1.2 – Exemples de techniques permettant d'illustrer le mouvement. Les figures suivantes sont extraites du livre de Scott McCloud [111] (p112) et illustrent le déplacement d'un personnage à l'aide d'un effet stroboscopique (a) ou d'un effet de flou de mouvement (b).

Des variantes de flèches et d'effets stroboscopiques sont également souvent utilisées pour illustrer la dynamique dans les bandes dessinées [111] : des symboles d'ondes pour décrire la propagation du son, des lignes d'action pour illustrer les mouvements ou les trajectoires de mouvements, des images multiples (tel un effet stroboscopique), un effet de traînée (*c.-à-d.* utiliser une ombre/version très simplifiée du modèle) ou un effet de flou (*cf.* figure 1.2). Certains de ces effets peuvent par ailleurs être appliqués sur l'objet ou sur le fond, selon que la caméra se déplace avec l'objet ou non [111].

Notons que dans les illustrations représentant la dynamique, les mouvements peuvent être explicites (un objet se déplace à l'intérieur d'une image) ou implicites (un objet se déplace entre deux images différentes) [111], en particulier lorsqu'il s'agit de représenter des opérations pas-à-pas qui sont généralement illustrées sous forme de storyboards ou de séquences d'interaction. La conception d'un storyboard consiste à structurer une figure avec plusieurs vignettes, dépendantes les unes des autres, et indiquant un flux continu d'événements dans le temps et/ou l'espace pour représenter le comportement attendu. Greenberg *et al.* [51, 52] ont dit que les storyboards « *communiquent des informations sur le lieu où se déroule l'interaction, présentent les personnes comme des personnalités et fournissent des détails sur les actions et choses que les personnes font pendant leur interaction* »⁷ (*cf.* figure 1.3). Les opérations à plusieurs étapes peuvent également être structurées comme une seule illustration, en utilisant des éléments de structure clés pour indiquer l'ordre du processus (généralement des nombres). Dans le contexte des manuels

7. « *They [storyboards] communicate information about the location where the interaction takes place, present the people as personalities, and provide details about the other actions and things people are doing as they interact.* » [51, 52]

d'instructions, il a été démontré que les instructions étape par étape sont préférées par les utilisateurs à une seule figure complexe (telle qu'une vue éclatée), pour comprendre des instructions de montage [5], confirmant ainsi l'utilité des storyboards pour présenter des informations séquentielles.



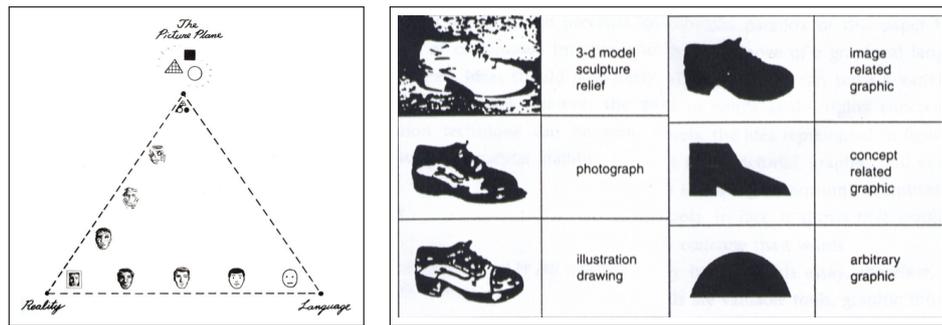
Figure 1.3 – Exemple de storyboard extrait du livre *Sketching User Experience - The Workbook* [51, 52, p67], relatant à travers plusieurs vignettes la séquence d'actions de la numérisation d'un QR Code par un utilisateur via son téléphone.

1.2.3 Apparence d'une illustration

Outre la structure d'une illustration, son apparence générale peut également être influencée par le contenu affiché et la manière dont il est présenté. Généralement, l'aspect est directement influencé par le type de source utilisé pour les illustrations : une photographie, une capture d'écran ou un dessin. Comme mentionné précédemment, le style du dessin peut différer dans le domaine de la conception de produits en fonction du cycle de vie du produit illustré ou de l'objectif véhiculé par l'illustration, et appartient à l'une des quatre grandes catégories différentes : *sketchs*, *dessins*, *modèles* ou *prototypes* [119]. Dans le contexte spécifique des dessins de design automobile, Tovey a étudié le processus de dessin des étudiants de troisième cycle et des designers professionnels [142]. Leurs résultats ont montré différentes catégories de lignes pour représenter un objet telles que les *lignes de formes* (c.-à-d. les lignes définissant les contours de l'objet), les *lignes de sommets* (c.-à-d. les lignes utilisées pour ajouter du volume à l'objet) ou les *lignes de zone* (c.-à-d. lignes définissant les ombres de l'objet).

Une approche plus souple est suggérée par Scott McCloud qui a établi que le style, l'apparence des bandes dessinées, pouvait être positionné dans un triangle qu'il appelle le « *vocabulaire pictural* » des dessins [111] (chapitre 2, p51-53), et dont les sommets sont *réalité*, *langage* et *plan de l'image*, correspondants respectivement aux styles *réaliste*, *iconique* et *abstrait*. Ainsi, chaque dessin peut être positionné à l'intérieur de ce triangle en fonction du style utilisé par l'auteur pour raconter son récit (cf. figure 1.4a).

Nous observons la même philosophie avec Wileman (cf. figure 8.6 dans [56]) qui décrit comment représenter un objet d'un concept abstrait jusqu'au dessin réaliste (cf. figure 1.4b). Ces notions de style et de complexité du dessin sont également représentées par



(a) Triangle du « vocabulaire pictural » de Scott McCloud [111]

(b) Échelle de réalisme selon Wileman [56] allant d'un modèle 3D réaliste d'une chaussure à une simple icône.

Figure 1.4 – Représentations du style d'un dessin selon (a) Scott McCloud [111] où chaque objet dessiné peut être placé dans le triangle représentant le « vocabulaire pictural » de l'illustration et selon (b) Wileman [56] où chaque objet est dessiné selon une échelle de réalisme allant d'une simple icône à un dessin réaliste.

Buxton [26] à travers l'élément clé *détails minimaux* ou par Fleming [45] à travers l'échelle du *style d'encodage* – visuel –, ce qui reflète l'hypothèse que lorsque l'on utilise des illustrations pour illustrer un discours autre que le dessin, le style doit être simpliste. C'est notamment le cas des figures à base de tracés, où la complexité du dessin est très faible puisque seuls les contours des objets représentés sont dessinés.

Enfin, dans le contexte spécifique de l'illustration de l'interaction, l'apparence d'une illustration peut différer pour des éléments spécifiques. McAweeney *et al.* recommandent d'utiliser le style de ligne comme moyen d'exprimer des lignes de mouvement spécifiques telles que les lignes pointillées pour l'interaction en l'air ou les lignes continues pour exprimer un mouvement sur une surface [109].

Un autre facteur important pour l'apparence d'une illustration est l'utilisation éventuelle de couleurs. Pour McAweeney *et al.* [109], la dimension *Coloration* est simplement une valeur booléenne représentant la présence ou non de la couleur dans l'illustration. Les auteurs considèrent la couleur comme un élément clé et la classent dans la dimension *structure*. Pour Fleming, cependant, l'utilisation des couleurs est décrite par les catégories *chrominance* et *non-chrominance* qui impliquent plus de détails tels que la transparence, la teinte, la nuance et des images de type noir et blanc ou en niveaux de gris.

Notez que par rapport aux dessins ou aux photographies classiques, les couleurs peuvent jouer un rôle essentiel dans le contexte de l'illustration de l'interaction. Généralement, elles peuvent être utilisées comme un moyen de mettre en évidence des aspects spécifiques d'une illustration, par exemple en soulignant le mouvement ou en mettant en évidence les points de contact du toucher (avec des cercles colorés) ou une zone spécifique [109]. Enfin, la couleur peut également être utilisée comme schéma de codage où différentes couleurs peuvent regrouper (respectivement distinguer) les éléments d'un même type (respectivement différent) [34].

1.2.4 Ajouter des annotations et des informations aux illustrations

Fleming considère que chaque figure est composée de trois types d'éléments distincts [45] : les éléments de type *pictural*, *verbal* et de *conception*. Les éléments de type *pictural* sont les illustrations, c'est-à-dire ce qui est représenté comme des formes géométriques, des dessins schématiques ou des graphiques ; les éléments de type *verbal* sont des éléments de texte ajoutés à la figure comme des étiquettes, des légendes ou des chiffres ; enfin, les éléments de type *conception* sont des éléments géométriques ajoutés à la figure en superposition comme des flèches, des lignes ou des zones colorées. Ces informations supplémentaires ajoutées au dessin/à la photographie font partie de l'illustration et sont considérées comme utiles pour aider à la diffusion d'informations importantes.

Hansen a exploré ce langage graphique et a identifié des outils graphiques pour aider à la visualisation [56]. Il les a classés en six familles : *cercle ou curvoïde*, *carré et coins*, *triangle*, *ligne*, *point* et *flou ou idée floue*. Ces catégories impliquent que des codes visuels peuvent être spécifiquement utilisés pour trier et catégoriser les informations présentées.

Pour simplifier la visualisation de toute représentation et aider l'utilisateur à identifier rapidement les éléments clés parmi tous les différents éléments structurant une illustration, certaines techniques d'annotation peuvent être utilisées. Chi, dans sa thèse sur la conception d'outils qui reposent sur la démonstration physique pour produire des illustrations de mouvements humains [34], a énuméré de telles techniques d'annotation. Les *mises en évidence* sont utilisées pour attirer l'attention par des formes (*p.ex.* marquer une zone) et des couleurs (*p.ex.* utiliser une couleur différente de celle du fond, des couleurs vives). Les *agrandissements* sont utilisés pour présenter une vue détaillée tout en préservant le contexte (connus sous le nom de loupes ou zooms). Les *annotations textuelles* sont utilisées pour fournir des informations utiles supplémentaires qui ne peuvent pas être dessinées. Les *icônes* peuvent être utilisées pour indiquer une situation générale (par exemple, une situation valide ou non). Enfin, les *chiffres* peuvent être utilisés pour indiquer l'ordre d'une opération à plusieurs étapes (ce qui peut être une alternative à un effet stroboscopique).

1.2.5 Lien avec les figures de brevets

On trouve également des illustrations de contextes interactifs dans les figures utilisées dans les brevets décrivant de nouveaux dispositifs ou systèmes interactifs. Les illustrations sont fortement recommandées pour ces brevets, car elles fournissent un contexte visuel pour s'assurer que les idées derrière les aspects techniques du brevet sont claires. La création d'une figure de brevet suit des normes [12, 145], de son organisation à son apparence. Un aspect important de l'organisation d'une figure de brevet est sa mise en page : les marges de la figure, le type de papier, la taille de la figure et du texte sont recommandés.

En ce qui concerne l'apparence de l'illustration, il doit s'agir d'un dessin en noir et blanc, mais des exceptions pour les couleurs et les photographies peuvent être acceptées

dans des cas spécifiques. Le dessin doit être aussi clair que possible, mais il est recommandé d'utiliser un « *jeu d'ombres* » pour donner une perception 3D au dessin si nécessaire. Enfin, les annotations supplémentaires sont très utilisées dans les figures de brevets, notamment pour relier les éléments visuels des illustrations aux descriptions textuelles correspondantes. À cette fin, des lignes doivent être utilisées pour relier l'élément visuel illustré et le texte correspondant à l'aide d'un identifiant numérique, être aussi courtes que possible et ne pas se croiser. Des flèches peuvent également être utilisées comme extension de ces lignes pour indiquer une zone globale ou, comme nous l'avons vu précédemment, la direction d'un mouvement. Il est intéressant de noter que le texte n'est pas recommandé dans l'illustration, sauf en cas de nécessité absolue.

2 Création de figures à base de tracés

La section précédente a présenté différentes notions et taxonomies pertinentes au sujet de comment une illustration peut représenter l'interaction. Dans cette section, nous nous intéressons aux outils qui permettent de produire de telles illustrations. En effet, les illustrations sont très utilisées dans la communauté IHM pour apporter un support visuel à la présentation d'une nouvelle technique d'interaction. Ces illustrations peuvent être de tout type : des photographies, des captures d'écran, des rendus ou des figures à base de tracés. Alors que prendre une photographie ou une capture d'écran est une tâche qui peut être effectuée en quelques secondes en cliquant sur un bouton physique ou virtuel, le processus de création d'une figure à base de tracés est quant à lui beaucoup plus complexe. Dans cette thèse, nous nous concentrons sur les processus créatifs et les outils utilisés pour créer ce type de figures. Dans les sections suivantes, nous présentons donc un état de l'art des méthodes actuelles, regroupées en deux catégories : les méthodes basées sur des modèles et les méthodes basées sur la mise en scène.

2.1 Méthodes basées sur des modèles

Si la production de figures à base de tracés est une pratique courante, en particulier pour les chercheurs en IHM, il est surprenant de constater que peu de travaux ont été consacrés à faciliter la production de ces contenus. La méthode la plus couramment utilisée reste le décalquage manuel de photographies [35, 153] qui peut être décrit comme suit. Tout d'abord, les utilisateurs mettent en scène la situation qu'ils veulent illustrer à l'aide de dispositifs physiques et d'autres personnes se tenant debout en tant qu'acteurs, et prennent une photo de celle-ci. Ensuite, ils transfèrent cette photo sur un ordinateur ou une tablette, et l'ouvrent en tant que modèle de fond dans un logiciel de retouche d'images. Troisièmement, ils esquissent des tracés sur la photo en utilisant leur dispositif d'entrée préféré (*p.ex.* souris, pavé tactile ou tablette graphique) et un logiciel de retouche graphique (généralement Inkscape ou Adobe Illustrator). Une fois le croquis terminé, ils

peuvent ajouter des éléments visuels, comme la superposition d'une interface sur un écran ou l'ajout de flèches, de textes ou de points de contact. Enfin, ils exportent la figure à base de tracés, la plupart du temps sous forme de fichier graphique vectoriel afin de pouvoir la redimensionner à volonté.

Cette méthode manuelle de traçage de photographies souffre encore de plusieurs limitations. Tout d'abord, elle est longue et relativement fastidieuse. Elle exige également que les utilisateurs disposent de dispositifs ou prototypes physiques ainsi que d'amis ou de collègues qui peuvent servir de modèles dans la photographie. Dans le cas contraire, ce processus peut nécessiter que les utilisateurs construisent des accessoires physiques, par exemple pour illustrer l'interaction avec des appareils qu'ils ne possèdent pas réellement ou qui n'existent que sous forme de modèles 3D. Une autre limitation est que certaines modifications dites mineures (par exemple un changement de posture des mains) peuvent obliger les utilisateurs à recommencer tout le processus depuis le début. Bien évidemment, un changement majeur dans la figure, tel que l'angle de l'illustration exige alors que les utilisateurs répètent le processus avec une nouvelle photo. Enfin, l'ajout d'interfaces sur les écrans des dispositifs illustrés peut nécessiter une transformation 3D en perspective, ce qui peut également être un défi pour les débutants.

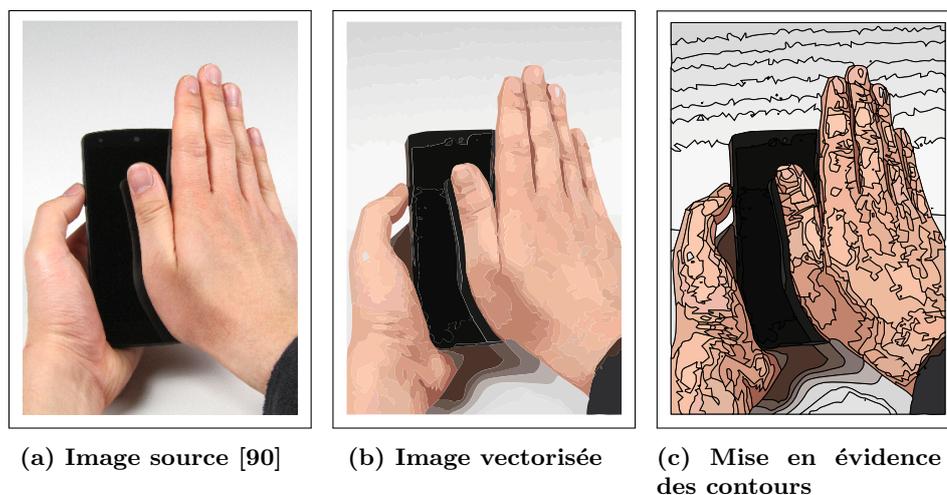


Figure 1.5 – Utilisation de la fonction de vectorisation automatique d'images dans Adobe Illustrator. L'image source [90] (a) est vectorisée en utilisant 30 couleurs (b). Cette vectorisation crée une quantité non négligeable de contours différents qu'il faut alors nettoyer (c).

Une technique permettant à l'utilisateur d'effectuer un ajustement local a été proposée pour faciliter le traçage manuel [154]. Cette méthode permet d'être assisté par l'ordinateur dans la tâche de traçage en détectant les contours locaux dans la zone sélectionnée par l'utilisateur. Cette approche semi-automatique, bien que facilitant la tâche de traçage de photographie, nécessite néanmoins toujours un processus manuel.

Certains logiciels d'édition de dessins vectoriels fournissent des outils automatiques

permettant de convertir une photo en une figure vectorielle ressemblant à un dessin (*p.ex.* Adobe Illustrator avec l'outil *Image trace*, *cf.* figure 1.5). Toutefois, ces méthodes reposent sur un nombre fini de nuances de couleurs à vectoriser. Ainsi, l'image vectorisée se retrouve segmentée en de nombreuses zones juxtaposées, représentant les différents remplissages de dégradés de couleurs et donnent ainsi une qualité visuelle différente de celle des figures à base de tracés (*cf.* figure 1.5b). En général, pour une main, la peau se compose de plusieurs couleurs pour recréer le gradient résultant de la lumière. De plus, une telle approche permet de vectoriser l'ensemble de l'image, y compris le fond ou les éléments qui seront finalement retirés de la figure à base de tracés, et requiert donc une étape de nettoyage supplémentaire des différents contours et objets à supprimer (*cf.* figure 1.5c).

Favreau *et al.* ont proposé Photo2ClipArt [43], une technique de vectorisation d'images tenant compte du gradient des objets. Cette méthode permet à partir d'une photographie et d'une version segmentée de cette photographie de générer une image vectorielle dont les couleurs des objets représentés sont dessinées en plusieurs couches superposées. Cette superposition de couches permet aux utilisateurs d'éditer avec précision le fichier SVG généré dans un logiciel d'édition externe. Malheureusement, pour obtenir de meilleurs résultats, leur approche nécessite que la photographie en entrée soit segmentée manuellement, ce qui peut vite être fastidieux pour une photographie complexe avec plusieurs utilisateurs et objets.

Les algorithmes de détection des contours, basés sur les travaux précurseurs de John Canny [27], semblent être une solution alternative pour tracer les contours d'une photographie modèle, mais ces algorithmes remplissent généralement les pixels plutôt que de produire un chemin vectoriel, et peuvent donner des résultats de faible qualité en fonction des conditions d'éclairage et de la qualité de l'image source. L'initiative mixte [113, 116] permet à l'humain de contrôler la détection de contours, mais souffre toujours des conditions d'éclairage et des objets flous.

La spécification des contours d'un objet est également un principe de base de la technique de rotoscopie utilisée en animation, et qui consiste à dessiner par-dessus des images consécutives (typiquement une vidéo). Le processus traditionnel de rotoscopie diffère toutefois du décalquage manuel de photographies, car il consiste toujours à créer plusieurs formes fermées pour chaque objet afin de faciliter les manipulations ultérieures [93]. Par conséquent, les outils spécialement conçus pour faciliter la rotoscopie, tels que Roto++ [93], sont moins adaptés à la production de figures à base de tracés.

L'outil de dessin de bande dessinée ClipStudio Paint ⁸ pourrait faciliter la production de figure à base de tracés lorsqu'un modèle de photographie ne peut être fourni. En effet, cet outil intègre une bibliothèque d'avatars 3D qui peuvent être utilisés comme support pour tracer manuellement des photos par-dessus. Cependant, cet outil est avant tout destiné à dessiner des personnages et n'est pas adapté à l'illustration de scénarios d'interaction.

8. ClipStudio Paint – <https://www.clipstudio.net/en/>

Il ne permet pas de se focaliser par exemple sur une partie du corps, telle que les doigts d'une main, ce qui pourrait servir à montrer une gestuelle sur un dispositif tactile.

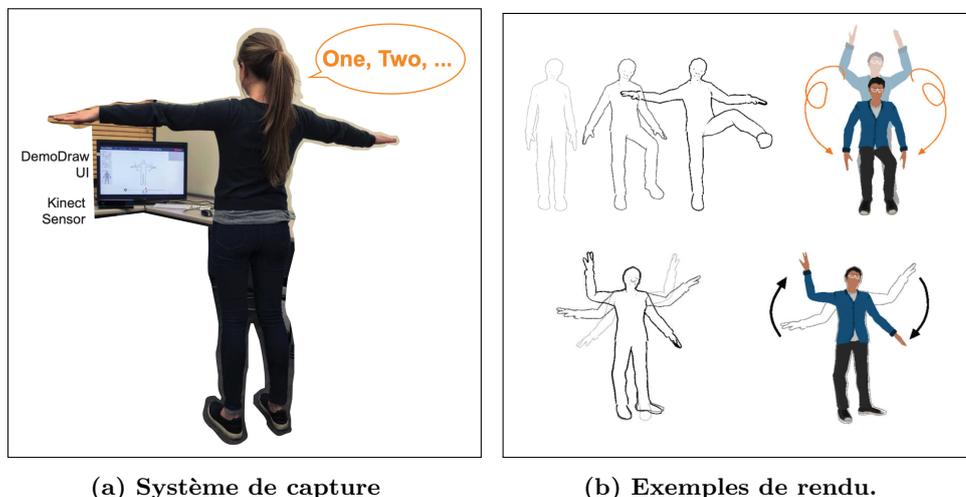


Figure 1.6 – Demodraw [35] permet à un utilisateur de créer une figure par démonstration. (a) L'utilisateur est placé devant la Microsoft Kinect et présente les différentes étapes de son mouvement. (b) Le système de rendu permet de générer des images illustrant le mouvement via un effet stroboscopique et/ou des flèches.

Enfin, DemoDraw [35] capture les mouvements du corps entier d'un utilisateur à l'aide d'une caméra Microsoft Kinect. L'utilisateur se place devant la caméra du système, et à l'aide de commandes vocales, demande au système d'enregistrer des postures de corps successives (*cf.* figure 1.6a). S'ensuit alors une phase d'édition permettant de personnaliser le rendu du mouvement capturé, tel que l'ajout d'annotations de type flèche en superposition ou l'utilisation d'un effet stroboscopique pour représenter le mouvement (*cf.* figure 1.6b). Bien que le système soit efficace pour illustrer une gestuelle, il ne se focalise que sur la capture du corps en entier, dû à l'utilisation de la Microsoft Kinect, et ne produit que des illustrations minimalistes et non vectorielles.

2.2 Méthodes basées sur la mise en scène

La mise en scène en 2D peut être utilisée pour produire des illustrations similaires aux figures à base de tracés, généralement en utilisant des cliparts trouvés en ligne ou suggérés par Google AutoDraw⁹, un outil basé sur l'apprentissage automatique et qui génère des cliparts à partir de croquis approximatifs dessinés par l'utilisateur, pour produire rapidement des figures à base de tracés sans s'appuyer sur un modèle photo. Néanmoins, cette approche nécessite d'abord de trouver les cliparts appropriés qui ne sont pas nécessairement disponibles en format vectoriel ou dans une résolution suffisante. Ensuite, la nature 2D même du clipart ne permet pas de modifier la perspective de la figure, ce qui peut être essentiel

9. AutoDraw – <https://www.autodraw.com/>

pour illustrer des scénarios interactifs impliquant plusieurs utilisateurs dans une pièce, ou des techniques d'interaction sur un smartphone basées sur l'inclinaison de l'appareil (par exemple, voir les figures 1 dans [87] et [129]).

SketchStudio est un outil de mise en scène 2.5D pour le prototypage de scénarios animés [83]. Cet outil, basé sur le contrôle de corps d'avatars via des nœuds de graphes qui décrivent le déplacement de cet avatar dans l'espace et le temps, permet de mettre en place des scènes qui peuvent être lues via une interface de visualisation dédiée (en réalité virtuelle). L'utilisation de cet outil pour la création de figures est toutefois limitée. Tout d'abord, il est axé sur la création de scénarios interactifs à *grande échelle* impliquant un utilisateur qui se déplace autour des appareils, ce qui le rend plus difficile à utiliser pour des interactions fines telles que l'interaction tactile sur un smartphone. En outre, tous les dispositifs et interfaces interactifs doivent encore être manuellement dessinés, ce qui nécessite des compétences en dessin de la part de l'utilisateur et limite le rendu de l'interface à un rendu de type croquis. Par ailleurs, l'outil ne produit pas de rendu vectoriel de la scène. Enfin, plusieurs participants à une expérience menée avec SketchStudio ont mentionné qu'il devient limité pour les idées qui doivent être illustrées sous forme 3D, c'est pourquoi Kim et al. suggèrent d'améliorer leur système pour prendre en charge l'utilisation de modèles 3D.

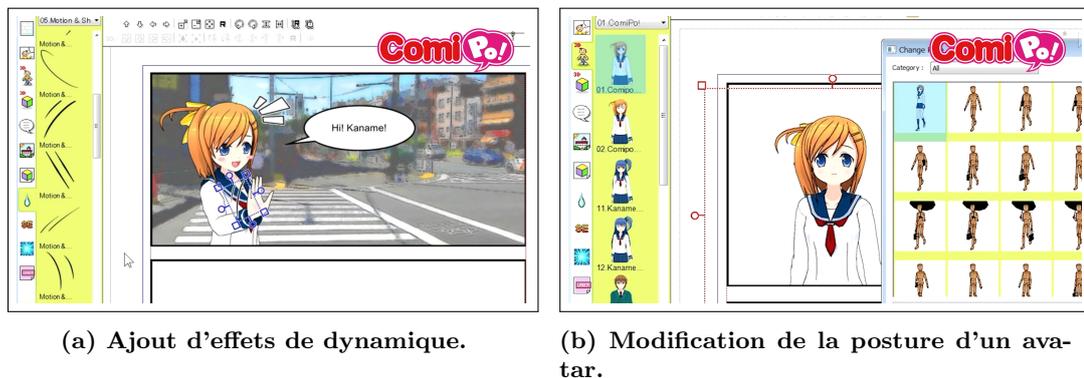


Figure 1.7 – Captures d'écran de la vidéo tutorielle de ComiPo! montrant les fonctionnalités offertes par ComiPo!. (a) L'outil permet d'ajouter des effets pour illustrer la dynamique, comme ici des traits de mouvement de la main droite de l'avatar. (b) ComiPo! permet aussi de modifier la posture d'un avatar via une bibliothèque de postures.

Enfin, ComiPo!¹⁰ est un outil de conception qui s'appuie sur des modèles 3D pour permettre de concevoir des bandes dessinées de type manga. La mise en scène en 3D pour produire des rendus en 2D résout ici de nombreux problèmes liés à la perspective. Cependant, outre le type de rendu, cet outil n'a pas pu être utilisé pour la production de figures à base de tracés pour l'IHM en raison du manque de prise en charge d'aspects tels que l'inclusion d'interfaces ou la production de format vectoriel. ComiPo! permet

10. ComiPo! – <https://www.comipo.com/>

globalement de mettre en scène un ou plusieurs personnages, ajouter un arrière-plan, des bulles de textes et des effets dynamiques comme dans les comics (*cf.* figure 1.7a). En outre, il n'utilise pas de système de caméra, ce qui signifie que le changement de perspective d'une scène nécessite que les utilisateurs fassent tourner indépendamment tous les objets de cette scène. Enfin, il n'intègre pas de système permettant une manipulation facile des armatures des modèles 3D articulés et requiert de choisir la sélection d'une posture via une bibliothèque de postures, ce qui peut montrer des limites lors de l'illustration d'un contexte interactif spécifique (*cf.* figure 1.7b).

Conclusion

Nous venons de voir dans ce chapitre que peu de travaux dans la littérature se sont focalisés sur l'illustration de l'interaction, et que les travaux existants concernent des domaines et représentations spécifiques, tels que la taxonomie de McAweeney *et al.* sur la représentation de la gestuelle [109]. Néanmoins, en reliant les travaux d'état de l'art avec des concepts d'illustration utilisés dans d'autres domaines tels que la bande dessinée, il est possible d'établir un ensemble d'éléments de conception, de codes préétablis permettant de caractériser les illustrations qui représentent des systèmes interactifs. Afin de palier au manque de travaux s'intéressant aux techniques d'illustration de tout type d'interaction, nous proposons dans le chapitre 2 une nouvelle taxonomie, établie par quatre chercheurs via un long processus de codage de figures extraites de quatre conférences en IHM. Cette taxonomie incorpore les éléments de conception proposés par les taxonomies existantes, mais propose aussi de nouveaux éléments permettant ainsi de couvrir un large spectre de représentations de l'interaction. Nous verrons ensuite dans ce chapitre qu'en analysant l'utilisation de ces éléments de conception au sein des diverses figures, il est possible d'identifier un ensemble de stratégies d'illustration permettant de représenter l'interaction.

Cet état de l'art a aussi montré les limites des outils et des techniques permettant de créer des figures à base de tracés, un type spécifique d'illustrations de plus en plus utilisé dans les documents produits par la communauté IHM. Notamment, les techniques actuelles offrent des possibilités de modification de la figurée créée limitées, ont un processus créatif généralement long et fastidieux, ou permettent de ne représenter que des illustrations d'un domaine spécifique comme la vidéo avec Roto++ [93] ou un type spécifique d'interaction comme la gestuelle avec DemoDraw [35]. Nous émettons dans le chapitre 3 l'hypothèse que l'agencement de scènes 3D est une alternative intéressante pour faciliter la production de figures statiques illustrant des scénarios interactifs. Nous présentons notamment Esquisse, un outil que nous avons développé, disposant de plusieurs techniques d'interaction spécifiques facilitant la manipulation d'objets 3D, et d'un algorithme de rendu innovant permettant de générer des figures vectorielles à base de tracés illustrant des scénarios interactifs.

2

Étude des stratégies d'illustration de scénarios interactifs

Sommaire

Introduction	31
1 Taxonomie des scénarios interactifs	33
1.1 Construction du jeu de données	33
1.2 Procédure de codage	34
1.3 Statistiques globales	37
2 Stratégies d'illustration de scénarios interactifs	42
2.1 Procédure	42
2.2 Stratégies de structure	44
2.3 Stratégies d'interaction	54
2.4 Stratégies de brevets	64
3 Outils pour le codage, la visualisation et l'exploration de la taxonomie	66
3.1 Outil de codage	66
3.2 Outil de visualisation	67
3.3 Outil d'exploration	68
Discussion	69
Discussion sur la méthodologie	69
Limitations de l'étude	71
Conclusion	71

Introduction

L'interaction avec les systèmes informatiques est une situation complexe dans laquelle un ou plusieurs utilisateurs interagissent avec un ou plusieurs systèmes. Nous appelons une telle situation un scénario interactif. Les illustrations de ces scénarios peuvent représenter des informations diverses et prendre des formes variées, qui peuvent être des photographies, des dessins, des diagrammes représentant des concepts et des idées ou des tables aidant à visualiser des données (*cf.* figure 2.1). La structure des illustrations peut également varier, en utilisant une ou plusieurs vignettes, étant complétée par des titres, des étiquettes, et même la nature des légendes peut être considérée comme faisant partie de l'illustration [45].

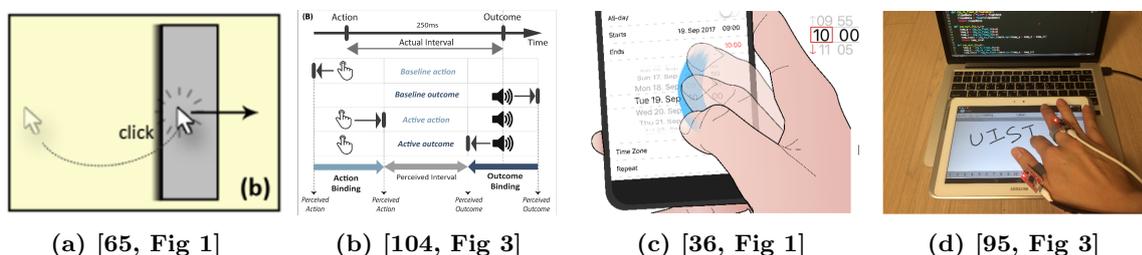


Figure 2.1 – Exemples de différents types d'illustrations de scénarios interactifs. (a) Capture d'écran d'une interface utilisateur. (b) Diagramme décrivant le comportement du système interactif. (c) Dessin d'un utilisateur interagissant avec un smartphone. (d) Photographie d'un utilisateur interagissant avec une tablette.

Dans ce chapitre, nous nous concentrons sur les techniques utilisées par la communauté de chercheurs en IHM lorsqu'ils produisent des figures statiques ayant pour but de représenter des scénarios interactifs. Nous appelons *figures statiques* les figures utilisées sur des supports statiques tels que les documents PDF contenant une illustration (*c.-à-d.* un dessin, une photographie) et des informations supplémentaires de type légende, titre, etc. (qui sont souvent considérées comme faisant partie des figures, car elles aident le lecteur à comprendre les illustrations [51]). Nous ne considérons dans ce travail que les figures représentant des contextes interactifs, où l'interaction peut être interprétée comme « *l'utilisation d'un outil* » [63], c'est-à-dire l'ensemble des figures « *définissant comment l'utilisateur agit avec un système et comment le système agit avec un utilisateur* ». Ces figures peuvent être particulièrement complexes à concevoir et nécessiter des techniques d'illustration élaborées pour représenter des aspects spécifiques de l'interaction tels que la 3D, la perspective, les gestes dynamiques, la temporalité, la gestion du corps de l'utilisateur, etc.

Nous avons vu dans le chapitre 1 que si certaines taxonomies concernant les illustrations ont été proposées, elles restent néanmoins limitées à des contextes spécifiques et il n'existe pas de travaux antérieurs proposant une taxonomie des représentations gra-

phiques couvrant un large spectre de scénarios interactifs. En effet, les travaux connexes portent sur la représentation gestuelle [109] ou des domaines spécifiques tels que le design ou l'éducation [45, 119]. Nous avons par ailleurs aussi élargi notre domaine de recherche, en combinant les concepts introduits par les taxonomies existantes et ceux introduits par des sources extérieures à l'IHM telles que les bandes dessinées, et les manuels ou vidéos d'instructions [5, 34, 111].

Ce chapitre a pour principal objectif d'identifier les stratégies d'illustration utilisées par les chercheurs en IHM pour produire ces figures. Pour étudier ces stratégies, nous avons besoin d'identifier les éléments de conception qui constituent les figures, c'est-à-dire des attributs conceptuels ou visuels d'une illustration utilisés pour coder l'interaction [109]. Nous proposons donc dans ce chapitre une taxonomie globale sur la manière dont l'interaction peut être illustrée, en s'appuyant sur les taxonomies existantes [45, 109, 119], mais sans se limiter à des contextes d'interaction spécifiques [109]. Nous avons construit un vaste jeu de données en utilisant des figures extraites de quatre conférences majeures en IHM et avons appliqué une approche empirique (*c.-à-d.* développer une théorie à partir de données ou d'observations empiriques lorsque la littérature existante est limitée [89, section 11.4.1.1]), pour déterminer les éléments de conception qui composent notre taxonomie. Notre hypothèse globale concernant ce travail est que l'identification des relations entre les éléments conceptuels – *ce qui* est représenté dans la figure – et visuels – *comment* est-ce représenté – nous aiderait à comprendre les stratégies employées pour représenter des types d'interaction spécifiques. Comme le nombre d'articles publiés dans le domaine de l'IHM ne cesse de croître et qu'il faut produire de plus en plus d'illustrations graphiques, ces stratégies identifiées pourront alors servir de lignes directrices pour les futures créations de figures illustratives.

Un deuxième objectif est de développer des outils en ligne dédiés à faciliter le long et fastidieux processus de codage et pour faciliter l'exploration et l'identification des stratégies mentionnées ci-dessus. Nous pensons que les différents outils que nous proposons, qui sont mis à disposition en open source, peuvent être utilisés par la communauté pour explorer davantage notre jeu de données, s'inspirer des figures existantes et identifier de nouvelles stratégies intéressantes.

La structure du chapitre est la suivante. Premièrement, nous présentons notre nouvelle taxonomie, en partant de la manière dont le jeu de données a été extrait des actes de quatre grandes conférences en IHM, à la procédure de codage des figures par quatre codeurs. Deuxièmement, nous nous intéressons à l'approche d'exploration que nous avons suivie pour identifier des stratégies utilisées pour représenter l'interaction dans les figures. Troisièmement, nous présentons les trois outils que nous avons créés pour visualiser la taxonomie, faciliter le processus de codage des figures, et explorer les stratégies dans le jeu de données. Enfin, nous discutons et concluons sur les limites de notre taxonomie, sa portée et ses possibles pistes d'améliorations.

1 Taxonomie des scénarios interactifs

1.1 Construction du jeu de données

Nous avons rassemblé les actes des conférences ACM CHI, UIST, CSCW et Ubicomp de 2018. Nous avons choisi ces conférences, car elles étaient, lorsque ce projet a démarré, les quatre premières conférences dans le classement Google Scholar en IHM¹. De plus, ces conférences couvrent un large spectre des contextes interactifs qui peuvent être représentés dans notre communauté, tels que les contextes de collaboration pour la conférence CSCW, les systèmes interactifs ambiants et ubiquitaires pour Ubicomp, les systèmes de détection ou de capture pour UIST et tout type de contexte pour CHI. Nous n'avons utilisé que les actes de l'année 2018 car le processus de codage est extrêmement fastidieux (voir les sections ci-dessous) et représente déjà un travail conséquent pour une seule année. Nous avons extrait automatiquement des actes 7243 figures avec leur légende correspondante, en utilisant l'outil *pdffigure2*². Enfin, nous avons manuellement trié les figures extraites qui au final, se retrouvent classées dans 8 catégories. Nous listons brièvement ci-dessous les types de figures rencontrés dans ces catégories, et nous nous référons à l'annexe A pour des exemples de figures.

- *Graphiques* (1893 figures) : graphiques (camemberts, barres, nuages de points), résultats NASA TLX, visualisations de signaux, cartes géographiques, etc.
- *Diagrammes* (617 figures) : diagrammes de flux, diagrammes UML, descriptions de système, circuits électriques, diagrammes de groupes et/ou d'entités, descriptions de structures de données, structures d'algorithme visuel, machines à états, calendriers, etc.
- *Systèmes et Environnements* (1491 figures) : images de résultats de reconnaissance (automatiques ou non), constructions et/ou prototypes (construits physiquement ou modélisés), rendus 3D, représentations de l'environnement du système.
- *Tableaux* (1774 figures) : résultats d'expériences, données, classifications.
- *Textes* (63 figures) : codes sources, textes utilisés pour de la reconnaissance de texte, commentaires des participants, citations, listes d'items, etc.
- *Interfaces* (734 figures) : captures d'écran brutes n'impliquant pas d'interaction (principalement axées sur l'aspect conception des interfaces graphiques et/ou leur apparence), messages d'erreur, etc.
- *Scénarios interactifs* (616 figures) : images impliquant des interactions entre un utilisateur et un système interactif, vues complètes ou partielles d'une installation interactive, interfaces utilisateurs impliquant une interaction, etc.

1. https://scholar.google.es/citations?view_op=top_venues&hl=en&vq=eng-humancomputerinteraction

2. <https://github.com/allenai/pdffigures2>

- *Artefacts* (55 figures) : ensemble des images générées par une erreur d'extraction de *pdffigures2* telles que des images vierges, des parties de figures incomplètes, des morceaux de texte, etc.

Parmi les 7243 figures, nous n'avons donc gardé que les 616 figures illustrant des scénarios interactifs, c'est-à-dire, où l'interaction peut être considérée comme un « *outil* » [63] et/ou impliquant un ou plusieurs utilisateurs interagissant avec un ou plusieurs systèmes interactifs. Comme présenté dans le chapitre d'état de l'art, nous avons également décidé de considérer des figures de brevets représentant des contextes interactifs. Nous avons donc complété notre jeu de données avec des figures de brevets trouvés dans les 30 premières pages de *GooglePatent* en utilisant le mot-clé "interaction" (recherche effectuée le 06/11/2018). Nous n'avons conservé que les brevets comportant au moins une figure représentant un scénario interactif, ce qui nous a donné un total de 47 brevets. Nous avons ensuite extrait manuellement 179 figures de ces brevets.

Notre jeu de données est finalement composé d'un total de $179+616=795$ figures illustrant des scénarios interactifs, extraites à la fois d'actes de conférences en IHM et de brevets.

1.2 Procédure de codage

1.2.1 Étape 1 : Établissement d'une taxonomie préliminaire — approche empirique.

Nous avons d'abord extrait aléatoirement un sous-ensemble de 100 figures de notre jeu de données. Comme la proportion de figures issues de brevets dans notre jeu de figures est d'environ 22%, nous nous sommes assurés que la même proportion était conservée dans le sous-ensemble extrait. Trois chercheurs qui ont travaillé sur ce projet ont agi en tant que codeurs *subjectifs* dans cette étape. Ces trois codeurs ont donc été chargés d'identifier les éléments de conception de chaque figure du sous-ensemble en créant des codes correspondants à partir de zéro, avec une seule contrainte : chaque code devait répondre à l'une des deux questions suivantes : *Qu'est-ce qui est représenté sur la figure ?* (les codes « *quoi* ») et *Comment est-ce représenté ?* (les codes « *comment* »). Une fois les figures du sous-ensemble codées, tous les codes créés par les codeurs ont été analysés. Pour ce faire, chaque code a été imprimé sur papier en utilisant une palette de couleurs pour différencier les codes *quoi* (en rouge) des codes *comment* (en bleu) (*cf.* figure 2.2a). Chaque morceau de papier comportait le nom du code et le nombre de fois où chaque codeur l'a utilisé, donnant ainsi une notion d'importance d'un code par rapport à son nombre d'occurrences. Les codeurs ont ensuite disposé manuellement et en collaboration les codes dans différentes catégories/hierarchies en fonction des relations entre les codes, de leurs similitudes et de leur importance (*cf.* figure 2.2b). Ce processus manuel a permis de créer rapidement une version préliminaire de la taxonomie.



Figure 2.2 – (a) Nous avons imprimé les codes issus de la première étape du codage sur des morceaux de papier et les avons groupés selon leur couleur. Les codes en rouge sont les codes « quoi », les bleus sont les codes « comment ». (b) Nous avons ensuite créé une version préliminaire de la taxonomie en créant manuellement des groupes de codes.

1.2.2 Étape 2 : Amélioration itérative de la taxonomie

La version préliminaire de la taxonomie a ensuite été améliorée via des itérations successives sur les codes établis, cette fois-ci avec quatre codeurs (dont les trois de l'étape 1). Chaque itération était composée des trois actions suivantes. Premièrement, les codeurs ont échangé sur la version courante de la taxonomie afin d'établir une liste de modifications à appliquer telles que l'ajout, la suppression, la fusion ou la réorganisation de codes ou catégories. Deuxièmement, un nouveau sous-ensemble de figures a été extrait (tout en conservant une proportion de 22% de figures de brevets) que les codeurs ont été invités à coder en utilisant les codes nouvellement modifiés, en utilisant une application de codage dédiée décrite dans la section 3. Troisièmement, un score de concordance entre les codeurs a été calculé. Comme recommandé par McDonald *et al.* [112, section 5.3.2], nous avons utilisé un score de concordance inter-juges comme mesure pour vérifier la concordance entre les codeurs. Plus précisément, nous avons utilisé les scores de concordance fournis par le Kappa de Fleiss et l'Alpha de Krippendorff [44, 86] (voir la sous-section 3.3 en fin de chapitre pour une discussion sur la façon dont ces méthodes ont été adaptées à notre problème). Ces mesures ont uniquement été utilisées comme indicateur pour évaluer notre processus de codage et observer la progression des scores de concordance. Ces trois étapes ont donc été répétées jusqu'à ce que le score de concordance entre les codeurs se stabilise.

Pour aider les codeurs dans la tâche de codage, nous leur avons également fourni un outil de visualisation, décrit dans la section 3, qui présente la version courante de la taxonomie sous la forme d'une arborescence interactive pouvant être parcourue pour visualiser les codes et leurs définitions grâce à des exemples visuels. Les codeurs étaient également autorisés à suggérer de nouveaux codes au cours de chaque phase de codage pour répondre à leurs besoins, mais ces codes ne pouvaient effectivement être utilisés pour les figures suivantes que dès l'itération suivante et uniquement si tous les codeurs étaient d'accord sur la modification. Il est à noter qu'après la première itération, certains codes de certaines catégories étaient attribués presque systématiquement. Nous avons donc considéré les catégories correspondantes comme des catégories *clés*, ce qui signifie qu'au moins un code à l'intérieur de chaque catégorie devait probablement être attribué lors du

codage d'une figure (mais ce n'était pas obligatoire). Nous avons défini les catégories clés suivantes : *point de vue*, *nombre de vignettes*, *type de figure*, *coloration*, *objectif*, *temporalité*, *type d'interaction*, *canal d'entrée* et *modalité de sortie*.

Les codeurs ont procédé à quatre itérations d'amélioration de la taxonomie au total, les résultats respectifs des scores de concordance pour chaque itération sont présentés dans le tableau 2.1. La première itération a été effectuée avec un sous-ensemble de 40 figures pour obtenir un jeu de données conséquent au départ. Des sous-ensembles de seulement 20 figures ont été utilisés pour les trois autres itérations. Les codeurs ont arrêté le processus après la quatrième itération, après laquelle 25% de l'ensemble du jeu de données a été codé et les scores de concordance ont été confirmés comme étant stables et *substantiels* [88] suggérant que les codeurs s'étaient mis d'accord sur les codes et leurs définitions. À la fin de la quatrième itération, les modifications proposées par les codeurs étaient systématiquement mineures, comme le simple fait de renommer un code ou d'ajouter un nouvel élément dans une catégorie exhaustive (par exemple les catégories *activité* ou *situation*), qui dépend directement du contexte de la figure sans fournir d'informations utiles. Enfin, nous avons également calculé le nombre moyen de codes sélectionnés par les figures pour chaque itération. Il est intéressant de noter que ce nombre augmente avec les itérations, peut-être en raison des nouveaux codes qui ont été ajoutés et du fait que les codeurs sont de plus en plus confiants au cours des itérations. Un aperçu complet de la taxonomie se trouve dans le tableau 2.2. Une version complète, avec les descriptions des codes, est présentée dans l'annexe B du manuscrit.

Itération	#Figures	#Codes	Kappa de Fleiss	Alpha de Krippendorff
1	40	24.1	0.61 ($\sigma = 0.08$)	0.61 ($\sigma = 0.08$)
2	20	23.6	0.66 ($\sigma = 0.11$)	0.66 ($\sigma = 0.11$)
3	20	25.4	0.61 ($\sigma = 0.10$)	0.61 ($\sigma = 0.10$)
4	20	27.2	0.63 ($\sigma = 0.10$)	0.63 ($\sigma = 0.10$)

Table 2.1 – Scores de concordance pour le Kappa de Fleiss et l'Alpha de Krippendorff entre les codeurs pour les quatre itérations d'amélioration de la taxonomie.

1.2.3 Étape 3 : Codage du reste des figures

Nous avons ensuite procédé au codage des 595 figures restantes, dont 460 extraites des articles des conférences en IHM et 135 des brevets. Comme le score de concordance était stable entre les codeurs, un seul codeur a finalement pris part à cette étape. Deux mesures ont été utilisées pour évaluer la précision de ce codeur. La première mesure était subjective et consistait simplement à donner un score de confiance de 1 à 5 après avoir codé chaque figure du jeu de données. Le score de confiance moyen était de 4.3 ($\sigma = 0.8$). La deuxième mesure était objective et consistait à calculer un score de stabilité. Nous avons dupliqué 5% des figures que le codeur devait coder – soit 29 figures – afin de vérifier que le

codage était cohérent entre les figures dupliquées. Le score de stabilité calculé pour chaque figure avec l'Alpha de Krippendorff et le Kappa de Fleiss était en moyenne d'environ 0.8 ($\sigma = 0.1$, $min = 0.5$, $max = 1.0$) pour les deux mesures. Notez qu'une seule figure a eu un score de 0.5 et huit un score de concordance parfaite de 1.0, sachant que le nombre moyen de codes sélectionnés pour les 29 figures est de 16.19. En se basant sur le fait qu'un score supérieur ou égal à 0.8 est considéré comme bon [88], nous avons donc considéré que le codeur était stable dans son codage.

1.2.4 Étape 4 : Recodage des figures utilisées pour la taxonomie préliminaire

Pour terminer, le codeur qui a codé le reste du jeu de données à l'étape précédente a aussi recodé les 200 figures utilisées dans les étapes 1 et 2, afin de mettre à jour le codage des figures avec la version définitive de la taxonomie. Cette étape est nécessaire afin d'homogénéiser les codes utilisés pour toutes les figures et de pouvoir ensuite analyser l'ensemble des 795 figures du jeu de données plutôt que de limiter l'analyse aux 595 figures codées à l'étape 3.

Notez enfin qu'au cours des étapes 3 et 4, respectivement 18 et 4 figures ont été exclues, car ces dernières n'ont finalement pas été considérées comme des représentations de scénarios interactifs par le codeur. Ce sous-ensemble correspond à 2.8% de l'ensemble des figures utilisées pour établir la taxonomie. Finalement, notre jeu de figures codées est composé de 773 figures, chacune étant codée avec un ensemble de codes *quoi* et *comment*.

1.3 Statistiques globales

Nous avons analysé les tendances globales des codes du jeu de données. Nous avons choisi d'exclure les figures issues des brevets de l'analyse, car ces figures semblent toujours utiliser les mêmes codes. Ceci n'est pas surprenant étant donné que les illustrations de brevets doivent suivre les règles mentionnées dans le chapitre 1. Cependant, nous examinerons en détail les stratégies utilisées pour les figures de brevets dans la section 2. Seules les 594 figures provenant des articles des actes des conférences en IHM sont donc analysées dans la section suivante.

1.3.1 Apparence des figures

387 figures (65%) ont utilisé des photographies tandis que 175 figures (28%) ont utilisé des dessins. Notez que les deux types ne sont pas exclusifs et l'utilisation de photographies et de dessins a été trouvée simultanément dans 28 figures (5%). Nous n'avons trouvé que 71 figures (11%) utilisant une interface utilisateur dessinée, et 142 (23%) utilisant une capture d'écran, ce qui représente un total de 213 figures (36%) incluant une interface utilisateur. Parmi ces 213 figures incluant une interface, cette dernière était la principale composante de la figure pour 117 d'entre elles (20%). 555 figures (93%) utilisaient des

couleurs, 26 (4%) étaient monochromes et 20 (3%) utilisaient seulement des niveaux de gris.

1.3.2 Objectif et structure

Nous avons trouvé 250 (42%), 232 (39%) et 171 (28%) figures représentant respectivement un espace de conception, un système interactif et une séquence d'interaction. La structure utilisée semble d'ailleurs être en corrélation avec le contexte représenté.

Espace de conception. Un espace de conception est une figure illustrant un ensemble de fonctionnalités, de possibilités, de commandes ou de cas d'utilisation disponibles dans un système. 229 figures (91% de 250) étaient composées de plusieurs vignettes indépendantes, c'est-à-dire qu'il n'y a pas de relation directe entre les vignettes. Dans ces 229 figures, 193 (84% de 229) ont utilisé une temporalité fixe à travers les vignettes (nous nous référons à la taxonomie détaillée fournie en annexe pour une définition des codes utilisés).

Système interactif. Une figure montrant un système interactif se concentre sur la configuration d'un système et ses capacités de détection plutôt que sur l'interaction entre l'utilisateur et le système. 136 figures (58% de 232) se sont concentrées sur un système, utilisant la plupart du temps une seule vignette. Parmi ces 136 figures, 129 (94% des 136) utilisaient une temporalité fixe dans la vignette illustrée.

Séquence d'interaction. Une séquence d'interaction est une séquence d'actions qui doivent être accomplies dans un ordre spécifique lors de l'interaction avec un système. 145 figures (84% de 171) étaient composées de plusieurs vignettes dépendantes, c'est-à-dire qu'il y a une relation entre ces vignettes. Pour ces 145 figures, le temporalité est évolutive dans 127 d'entre elles (87% des 145), c'est-à-dire que le temps progresse à travers les différentes vignettes.

1.3.3 Représentation de la dynamique

Nous avons trouvé 170 figures (28%) représentant une temporalité évolutive. Il est intéressant de noter que parmi ces 170 figures, 147 (86%) illustrent une séquence d'interaction. L'évolution du temps est illustrée à l'aide de trois techniques principales.

Vignettes multiples. 141 (82% des 170) ont utilisé plusieurs vignettes pour illustrer la progression du temps à travers la figure. Pour se référer à chaque vignette, 97 figures (68% de 141) ont utilisé des lettres, 23 figures (16% de 141) des titres et 9 (6% de 141) des chiffres.

Lignes et flèches. Nous avons également constaté une utilisation importante des lignes et des flèches pour représenter les mouvements (52 figures, 30% de 170), les trajectoires (41 figures, 24% de 170) et des transitions entre les vignettes (24 figures, 14% de 170).

Effets du mouvement. 41 figures (24% des 170) ont utilisé un effet stroboscopique pour représenter une progression temporelle des éléments dans la figure. Ce petit nombre peut

s'expliquer par le fait qu'il faut de meilleures compétences en dessin et en conception de figures pour produire un effet stroboscopique.

1.3.4 Représentation de l'utilisateur

Nous avons trouvé un total de 558 figures (96%) illustrant des utilisateurs, ce qui laisse 37 figures où aucun utilisateur n'est représenté (*p.ex.* représentation du système interactif uniquement).

Nombre d'utilisateurs. 491 figures (88% des 558) représentent un seul utilisateur et seulement 67 figures (12% des 558) plusieurs utilisateurs.

Parties du corps. Sans surprise, la partie du corps la plus fréquemment représentée est la main avec 302 figures (50%). Cela s'explique facilement car la majorité des dispositifs d'interaction (clavier, souris, écrans tactiles, etc.) s'opèrent avec les mains. De plus, les contextes d'interaction les plus représentés sont l'interaction tactile (192 figures, 32%) et l'interaction tangible (152 figures, 25%). Ensuite, nous avons trouvé des représentations du haut du corps de l'utilisateur, de tout son corps et de ses doigts avec des proportions correspondantes de 130 (21%), 85 (14%) et 73 figures (12%).

Anonymisation. Étonnamment, nous n'avons trouvé dans le jeu de données que 9 figures (1%) qui avaient été modifiées dans le but de préserver l'anonymat des utilisateurs pris en photo (en utilisant une technique de floutage dans notre cas).

Point de vue. Le point de vue le plus fréquemment utilisé est le point de vue à la 3^{ème} personne avec 323 figures (54%). Parmi les 323 figures, 254 (78% des 323) illustrent un utilisateur unique et 52 figures (16% des 323) plusieurs utilisateurs. Le deuxième point de vue le plus utilisé est le point de vue à la 1^{ère} personne avec 170 figures (29%), principalement utilisé pour illustrer le point de vue d'un seul utilisateur (152 figures, 89% de 170).

Scénario interactif ○	Objectif *	Espace de conception	
		Séquence d'interaction	
		Système interactif	
	Temporalité* ○	Évolutive	
		Fixe	
	Activité	Production	
		Divertissement	
		Médical	
		Conduite	
		Création 2D/3D	
Fabrication			
Communication			
Manipulation de données			
Utilisateurs ○ ○ ○	Nombre	Un utilisateur	
		Utilisateurs multiples	
	Partie du corps ●	Membre spécifique ●	Main ○
			Doigt ○
			Tête
			Pied
		Bas du corps ●	
Haut du corps ●			
Corps entier ●			
Système interactif	Action	Discrète	Pointage
			Frappe de touches
			Geste symbolique ●
		Continue	Translation
			Rotation
			Mise à l'échelle
			Défilement
			Déformation
			Écriture/dessin
			Type de l'interaction *
	Interaction dans l'air ○		
	Interaction à distance		
	Interaction sur le corps		
	Interaction sur ordinateur		
	Interaction tangible ○		
	Interaction via des contrôleurs		
	Interaction via des crayons		
	Interaction via le regard		
	Canal d'entrée *	Système de capture	Capture externe
			Capture interne
		Dispositifs	Dispositif mobile ○
			Grande surface
			Dispositif de bureau
			Dispositif portable
			Contrôleur
			Objet tangible
			Dispositif de RA
Dispositif de RV			
Utilisation du son	Entrée vocale		
Modalité de sortie *	Haptique		
	Visuelle	Grand écran	
		Écran d'ordinateur	
		Écran de dispositif mobile	
		Écran projeté	
		HMD	
	Lumière		
	Audible	Son	
		Sortie vocale	
	Situation ●	Bureau	
Transport public ou privé			
Extérieur			
Intérieur privé			
Intérieur public			

Caractéristiques visuelles	Type *	Dessin ●	Réalisme	Simpliste	
		UI	Dessin	Réaliste	
		Photographie			
		Visualisation de données			
		Clipart/icône			
		Texte ●○			
		Transparence			
	Coloration * ●○●	Teinte ○●	Couleurs ●○●		
			Niveaux de gris		
			Monochrome ●		
Style de ligne	Pointillée ●				
	Épaisseur				
Composition	Point de vue * ●	1 ^{ère} personne ●			
		3 ^{ème} personne ●			
		Par dessus l'épaule			
		Vue du dessus			
		Interface utilisateur seule			
	Disposition	Nombre de vignettes * ●●	Une vignette ●		
			Vignettes multiples ●		
		Sous-vignettage	Image dans l'image		
			Agrandissement		
			Incrustation d'interface		
		Interface superposée			
		Juxtaposition			
Techniques visuelles	Mise en évidence ○	Région ●	Zone colorée		
		Élément	Flèche		
			Élément coloré ○		
			Contours	Contour de l'objet	
			Contour géométrique		
	Arrière-plan	Grisé/Flouté			
		Supprimé			
	Anonymisation	Floutage			
		Remplissage			
	Grouper et lier	Grouper par couleurs			
		Grouper avec des indicateurs			
		Grouper avec des flèches			
		Annotations textuelles			
	Dynamique	Effet dynamique	Effet stroboscopique ●		
			Onomatopée		
			Flou de mouvement		
		Ligne et flèche ●	Trajectoire		
Direction					
Transition					
Point de contact ●	Projection				
Structure	Indicateur	Chiffre ●			
		Lettre			
		Sous-titre			
Mesure	Flèche				
	Indicateur textuel				

□ Codes *Comment* □ Codes *Quoi* * Catégories clés

- Élément de conception équivalent *explicité* ○ Élément de conception équivalent *mentionné*
- Fleming [45] ●○ McAweeney *et al.* [109] ●○ Pei *et al.* [119]

Table 2.2 – Notre hiérarchie d'éléments de conception se compose de six catégories principales. Les catégories clés utilisées pour faciliter le processus de codage sont affichées avec un symbole d'étoile. Les éléments de conception existants introduits avec les taxonomies précédentes sont affichés avec un cercle coloré. Les cercles pleins indiquent les éléments composants les taxonomies existantes tandis que les cercles vides indiquent les éléments mentionnés dans les articles correspondants. Nous avons pris en compte les termes équivalents tels que *Ligne pointillée* dans notre taxonomie contre *Ligne discontinue* dans la taxonomie de McAweeney *et al.* [109]

2 Stratégies d'illustration de scénarios interactifs

2.1 Procédure

Nous avons rassemblé un jeu de données de 773 figures, chacune étant codée avec un ensemble de codes *quoi* et *comment*. Notre objectif principal est de trouver des relations entre les utilisations des codes et les figures, c'est-à-dire de trouver des groupes de figures pertinents utilisant des groupes de codes similaires. Notre hypothèse est que dans chaque groupe de figures identifié, l'analyse des relations entre les codes *quoi* et les codes *comment* permettrait de comprendre les stratégies utilisées pour représenter l'interaction. Nous définissons donc une stratégie comme la relation entre un groupe de codes *quoi* et/ou *comment* dans un ensemble de figures. Une hypothèse est que, plus un groupe de codes donné est présent dans différentes figures, plus on s'attend à ce qu'une stratégie correspondante soit utilisée pour représenter le concept illustré. Nous n'avons pas trouvé de méthode appropriée ou standard pour analyser ce problème multidimensionnel (voir la discussion sur les méthodes statistiques possibles et leurs limitations à la fin du chapitre), c'est pourquoi nous avons choisi de chercher à identifier des stratégies en utilisant une approche basée sur l'exploration à l'aide d'un outil dédié (décrit dans la section 3). Au cours de l'exploration des données, nous avons identifié trois types de stratégies : les **S**tratégies de **S**tructure (**SS**), les **S**tratégies d'**I**nteraction (**SI**) et les **S**tratégies de **B**revets (**SB**).

Les **S**tratégies de **S**tructure sont des stratégies qui peuvent être appliquées à n'importe quelle figure, quel que soit le type d'interaction illustré, car elles sont principalement composées de groupes de codes *comment*. Elles fournissent des lignes directrices sur la manière de traiter les bases de la conception des figures. Pour trouver ces stratégies, nous avons appliqué une méthodologie ascendante. Nous avons successivement sélectionné les codes *comment* dans notre outil exploratoire pour observer l'impact sur les autres codes *comment* et les codes *quoi* sans autre traitement. Nous avons ensuite analysé les chiffres obtenus pour comprendre les tendances décrites dans les sections suivantes. Nous avons identifié cinq stratégies de structure : *choisir une structure de vignette* (**SS1**), *choisir un point de vue* (**SS2**), *sous-vignettage* (**SS3**), *définir des relations entre les éléments* (**SS4**) et *prétraitement de photographies* (**SS5**).

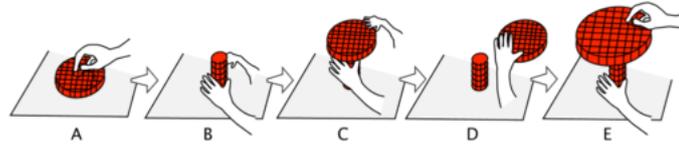
Les **S**tratégies d'**I**nteraction sont des stratégies qui s'appliquent principalement à des contextes interactifs spécifiques. Elles sont composées de groupes de codes *comment* et de codes *quoi* et fournissent donc des lignes directrices sur la représentation d'un élément spécifique de la figure. Pour identifier ces stratégies, nous avons appliqué une méthodologie de regroupement et de classification descendante. Nous avons successivement sélectionné les codes *quoi* un à un et, pour chacun d'eux, nous avons récursivement sélectionné les autres codes *quoi* les plus fréquemment associés jusqu'à ce que le groupe de figures formé soit pertinent. Ensuite, nous avons observé les codes *comment* les plus fréquents dans

cet ensemble de figures résultant pour identifier la stratégie. Nous avons identifié cinq stratégies d'interaction : *définir l'espace interactif (SI1)*, *mettre en évidence les éléments interactifs (SI2)*, *représenter du contenu dynamique dans une seule vignette (SI3)*, *mettre en évidence les actions des utilisateurs (SI4)* et *illustrer le retour non visuel (SI5)*.

Les **S**tratégies de **B**revets (**SB**) étaient plus faciles à identifier. Comme mentionné dans l'état de l'art (*cf.* chapitre 1), il existe des lignes directrices sur la manière d'illustrer les figures de brevets, notamment afin de les normaliser, influençant donc nécessairement les stratégies utilisées. Nous avons utilisé la même méthodologie ascendante que pour les stratégies de structure afin d'identifier ces stratégies. Nous mettons également en évidence des exemples de figures de brevets montrant que les stratégies de structure et d'interaction introduites sont aussi utilisées pour illustrer les contextes interactifs dans les brevets.

Toutes ces stratégies sont décrites ci-dessous en utilisant des figures provenant d'articles de recherches et de brevets comme exemples. Veuillez noter que certaines figures présentées dans les sections suivantes sont des portions de figures originales afin de se focaliser sur un aspect spécifique (par exemple, une seule vignette est montrée alors que la figure en contient plusieurs). Aucune autre modification n'a été effectuée dans ces figures. Enfin, si vous lisez ce manuscrit numériquement, nous vous conseillons de configurer votre lecteur dans le mode de présentation en deux pages afin de visualiser à la fois les exemples et le texte des stratégies décrites dans les prochaines sections.

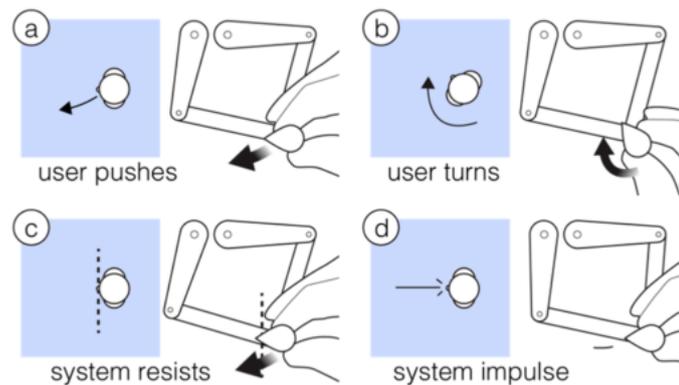
2.2 Stratégies de structure



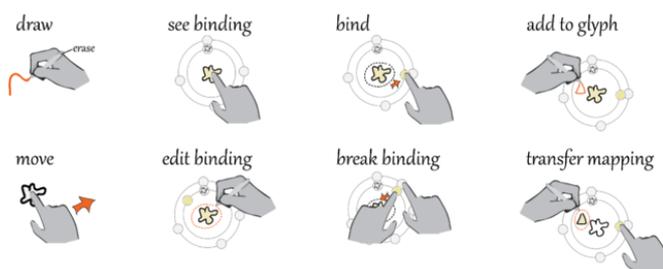
(a) Séquence d'interaction avec indicateurs de type lettre [140, Fig. 16]



(b) Espace de conception avec indicateurs de type lettre [90, Fig. 1]



(c) Espace de conception, organisation en grille de 2×2 , indicateurs de type lettre [131, Fig. 3]



(d) Espace de conception, organisation en grille de 2×4 , indicateurs de type sous-titre [152, Fig. 7]

Figure 2.3 – Exemples de figures utilisant différentes structures de vignettes.

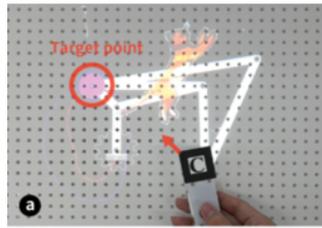
SS1 : Choisir une structure de vignettes

La structure des vignettes représente la disposition des vignettes dans la figure, leur nombre et la façon dont elles sont disposées. Selon le scénario interactif à représenter, différentes structures sont utilisées.

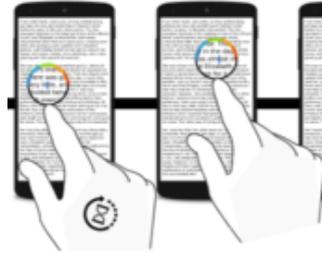
Nombre de vignettes. 414 (69%) figures de notre jeu de données utilisent plusieurs vignettes. Le nombre de vignettes dépend directement du contexte interactif. Alors que la seule représentation d'un système interactif peut ne nécessiter qu'une seule vignette, l'illustration de séquences d'interaction (*cf.* Figure 2.3a) ou d'espaces de conception (*cf.* Figures 2.3b, 2.3c et 2.3d) utilise plusieurs vignettes pour montrer les principales étapes de la séquence ou les différentes possibilités de l'interaction présentée. Parmi les figures utilisant plusieurs vignettes, 35% représentent des séquences d'interaction et 55% des espaces de conception.

Organiser les vignettes. L'organisation des vignettes dépend du contenu représenté, du nombre de vignettes et surtout de l'espace disponible pour la figure. Les vignettes de séquences d'interaction sont souvent organisées selon une disposition unidimensionnelle pour faciliter la succession des vignettes (généralement de gauche à droite, *cf.* figure 2.3a). Les espaces de conception n'ont eux pas besoin d'être séquencés et offrent donc plus de liberté pour utiliser différentes dispositions, généralement une disposition en grille, comme dans les figures 2.3c et 2.3d. Il est intéressant de noter que ces grilles peuvent être "imbriquées", comme dans la figure 2.3c où chaque vignette de la grille 2×2 (identifiée par des lettres, voir ci-dessous) est composée de deux sous-vignettes différentes juxtaposées (*cf.* stratégie SS3).

Identification des vignettes. Lorsqu'une figure a plusieurs vignettes, il peut être nécessaire d'indiquer l'ordre des vignettes au lecteur. Ces indicateurs sont généralement affichés dans un coin ou une bordure de vignette. Pour ordonner les vignettes, comme dans les séquences d'interaction, des lettres ou des chiffres sont généralement utilisés pour identifier chaque étape de la séquence (voir figure 2.3a). Les indicateurs sont souvent accompagnés d'un sous-titre lorsqu'un ordre des vignettes n'est pas nécessaire, par exemple, dans les espaces de conception. À titre d'exemple, la figure 2.3b utilise uniquement des lettres, la figure 2.3d des sous-titres et la figure 2.3c les deux. Comme les lettres et les chiffres sont utilisés à la fois pour ordonner ou non les vignettes, c'est au lecteur de comprendre le contexte pour deviner leur signification. Dans ce cas, des flèches de transitions entre les vignettes successives peuvent être utilisées comme indicateur supplémentaire pour renforcer une séquence et guider l'ordre de lecture de la figure (*cf.* figure 2.3a). Dans notre ensemble de figures, 44% utilisent des lettres, 13% des sous-titres, 5% des flèches de transition et 2% des chiffres.



(a) Vue du dessus [71, Fig. 5]



(b) Vue du dessus [49, Fig. 2]



(c) Vue à la 1^{ère} personne [100, Fig. 3]



(d) Vue à la 1^{ère} personne [143, Fig. 2]



(e) Vue à la 3^{ème} personne [29, Fig. 5]



(f) Vue à la 3^{ème} personne [151, Fig. 1]



(g) Vue par dessus l'épaule [108, Fig. 5]



(h) Vue par dessus l'épaule [136, Fig. 6]

Figure 2.4 – Exemples de figures utilisant différents points de vue.

SS2 : Choisir un point de vue

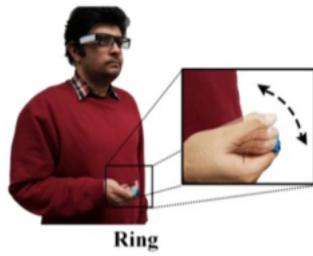
Chaque vignette d'une figure utilise un point de vue spécifique, c'est-à-dire un point d'observation, une perspective du contexte interactif représenté. Nous avons identifié quatre points de vue principaux : 3^{ème} personne, 1^{ère} personne, vue de dessus et vue « par-dessus l'épaule ».

Vue à la 3^{ème} personne. La perspective la plus fréquemment utilisée, avec 323 figures (54%), représente un utilisateur interagissant avec un système, ou le système lui-même, d'un point de vue à la 3^{ème} personne. Par exemple, la figure 2.4e montre un utilisateur portant un HMD et la figure 2.4f illustre la posture de l'utilisateur dans un contexte d'interaction sur son bureau.

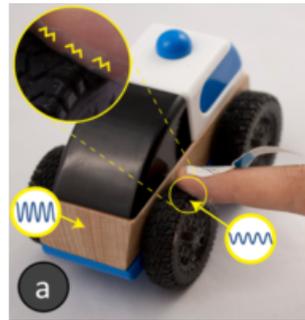
Vue à la 1^{ère} personne. Un autre point de vue fréquemment utilisé dépeint la scène du point de vue de l'utilisateur, en utilisant une vue à la 1^{ère} personne (170 figures, 28%). Selon le contexte, la façon dont ce point de vue est utilisé peut varier. Généralement, dans un contexte de RV (24%) ou RA (14%), la vue à la 1^{ère} personne est souvent utilisée pour montrer directement ce que l'utilisateur voit à travers son casque avec l'interface utilisateur correspondante (*cf.* figures 2.4e et 2.4c). Dans d'autres contextes, l'utilisation d'une perspective de type 1^{ère} personne est surtout utilisée pour se concentrer sur les mains ou l'appareil de l'utilisateur, par exemple dans un contexte d'interaction tactile (notez que 31% des figures représentent des appareils mobiles). À titre d'exemple, la figure 2.4d montre les gestes tactiles spécifiques à utiliser sur un smartphone et la posture de la main adoptée.

Vue du dessus. La représentation de la scène d'un point de vue du dessus (101 figures, 17%) peut être utilisée pour obtenir un grand angle d'une scène afin d'observer et faciliter la lisibilité des interfaces utilisateur en évitant l'utilisation d'une perspective. Par exemple, la vue de dessus est surtout utilisée pour les interactions tactiles (54 %) et tangibles (20 %) car elle permet d'avoir une vue de l'interface utilisateur, des membres de l'utilisateur tels que les mains ou les doigts et des appareils utilisés. À titre d'exemple, la figure 2.4a représente des objets tangibles et une surface interactive et la figure 2.4b illustre des gestes spécifiques sur un smartphone.

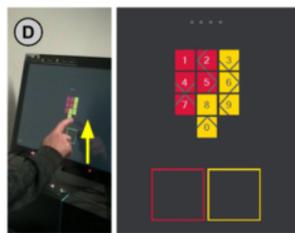
Vue par-dessus l'épaule. Utilisé dans 55 figures (9%), ce point de vue vise principalement à illustrer la distance entre l'utilisateur et le système avec lequel il interagit. Bien qu'il ne soit pas spécialement utilisé pour représenter de l'interaction à distance, 21 % des figures illustrent les tâches de pointage à distance. En positionnant le point de vue au-dessus de l'épaule de l'utilisateur, il est possible d'observer l'interface utilisateur et la direction pointée par l'utilisateur, comme illustré dans les figures 2.4g et 2.4h représentant du pointage à distance avec la main sur un mur ou avec une télécommande sur un téléviseur.



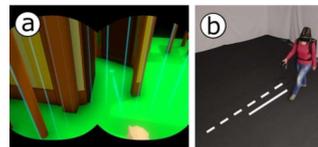
(a) Agrandissement de vignettes [7, Fig. 1]



(b) Agrandissement de vignettes [147, Fig. 5]



(c) Juxtaposition de vignettes [80, Fig. 1]



(d) Juxtaposition de vignettes [105, Fig. 12]



(e) Image dans l'image [22, Fig. 3]



(f) Image dans l'image [94, Fig. 4]



(g) Fusion de vignettes [156, Fig. 11]



(h) Fusion de vignettes [100, Fig. 6]

Figure 2.5 – Exemples de figures utilisant le sous-vignettage.

SS3 : Sous-vignettage

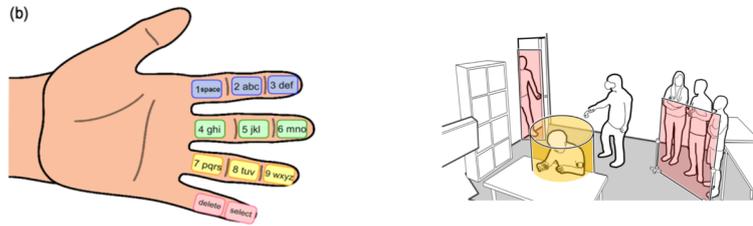
Pour une structure de vignettes donnée pour une figure et un point de vue pour chaque vignette, différentes stratégies peuvent être utilisées pour combiner des éléments visuels en utilisant le sous-vignettage.

Agrandissement de vignettes. L'agrandissement de vignettes consiste à intégrer des sous-vignettes dans la vignette principale pour représenter une partie zoomée de la figure. Ces sous-vignettes sont utilisées pour fournir à la fois des détails et du contexte à ce qui est représenté. Elle ajoute des informations secondaires qui ne peuvent pas être représentées au niveau de la figure principale tout en gardant une même perspective. La zone zoomée est généralement entourée et liée aux vignettes. Nous avons compté 24 figures (4% de l'ensemble du jeu de données) utilisant l'agrandissement des vignettes. Dans la figure 2.5a, la sous-vignette fournit des détails sur un geste spécifique de la main avec un effet stroboscopique et une flèche de mouvement qui seraient difficiles à remarquer au niveau de la figure. Dans la figure 2.5b, la sous-vignette donne une représentation détaillée du retour haptique.

Juxtaposition de vignettes. Des sous-vignettes peuvent également être juxtaposées et séparées par une ligne ou un espace blanc, verticalement ou horizontalement. Cette stratégie est souvent utilisée pour représenter différentes perspectives sur une même figure. 80 figures (13% du jeu de données) utilisent la juxtaposition de vignettes, dont 25% correspondent à un contexte d'interaction de RV. Les figures 2.5c et 2.5d sont deux exemples d'utilisation de sous-vignettes juxtaposées, où l'une représente la perspective de l'utilisateur et l'autre l'interface utilisateur.

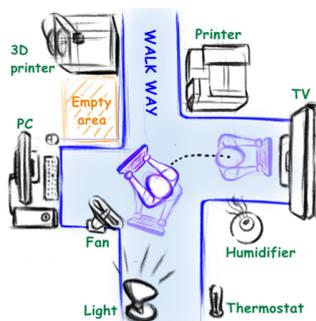
Incrustation (image dans l'image). Cette stratégie consiste à intégrer une sous-vignette dans la vignette principale. Contrairement à l'agrandissement des vignettes, la sous-vignette représente généralement la scène d'un point de vue différent, et non pas à un autre niveau d'échelle. Cette technique a été utilisée dans 38 figures (6% du jeu de données). Les figures 2.5e et 2.5f montrent deux cas d'utilisation d'image dans l'image, comme l'illustration de différents points de vue dans la même figure (provenant de différents utilisateurs par exemple) ou l'affichage de détails d'un dispositif créé pour un système.

Fusion de vignettes. Enfin, des vignettes peuvent être fusionnées (souvent deux) en les superposant les unes sur les autres par transparence. Cette stratégie est surtout utilisée pour superposer du contenu virtuel/numérique sur la représentation du monde réel. La fusion de vignettes a été utilisée dans 47 figures (7 %) de notre ensemble de figures, dont 67% illustrent l'interaction en RA. Par exemple, la figure 2.5g superpose avec transparence les limites de l'espace interactif pour l'utilisateur en RV. La figure 2.5h superpose l'interface utilisateur dans un contexte de RA.



(a) Groupement de zones d'interaction avec des couleurs [149, Fig. 1]

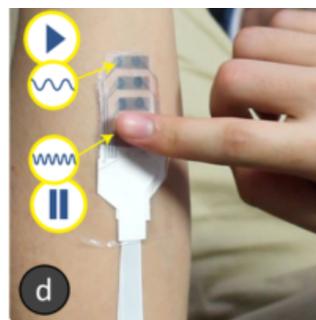
(b) Groupement de zones d'interaction avec des couleurs [156, Fig. 1]



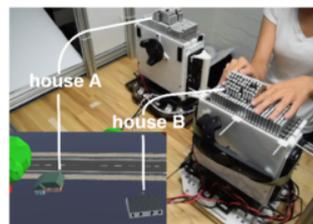
(c) Groupement d'objets avec des couleurs [67, Fig. 8]



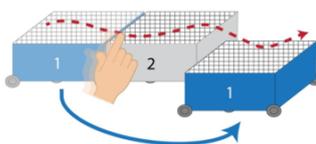
(d) Groupement de la légende avec des couleurs [68, Fig. 6]



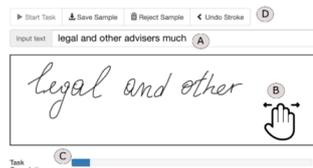
(e) Groupement avec des flèches [147, Fig. 5]



(f) Groupement avec des lignes [135, Fig. 9]



(g) Groupement avec des chiffres [135, Fig. 7]



(h) Groupement avec des lettres [6, Fig. 14]

Figure 2.6 – Exemples de figures illustrant des relations entre les éléments.

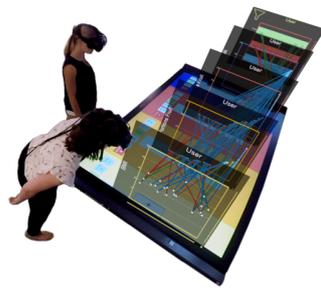
SS4 : Définir des relations entre les éléments

Ces stratégies spécifiques regroupent des techniques utilisées pour grouper ou lier des éléments à l'intérieur ou à l'extérieur des figures, par exemple en associant une étiquette ou une définition à un élément, ou en regroupant plusieurs éléments dans une ou plusieurs vignettes. Trois stratégies sont utilisées à cette fin : les couleurs, les formes géométriques et les indicateurs.

Groupement par couleur. Le groupement par couleur (utilisé dans 73 figures, 12%) consiste à grouper les éléments d'une figure en utilisant la même couleur pour le remplissage, les traits ou le texte. Par exemple, les zones d'interaction de la figure 2.6a sont affichées en utilisant des couleurs différentes, les zones correspondant au même doigt étant affichées avec la même couleur. La figure 2.6b regroupe les frontières de la zone d'interaction globale en rouge et de la zone personnelle en orange. La figure 2.6c utilise différentes couleurs, le vert pour les objets interactifs et le bleu pour la zone d'interaction. Enfin, la figure 2.6d utilise le groupement de couleurs pour relier les éléments de la figure à une légende.

Groupement à l'aide de formes géométriques. Des formes géométriques simples peuvent également être utilisées pour regrouper des éléments. En général, les flèches peuvent être utilisées pour grouper un objet du système interactif avec un élément descriptif (généralement une étiquette), tandis que les lignes peuvent être utilisées pour relier plusieurs objets d'une figure. Par exemple, la figure 2.6e utilise des flèches jaunes pour ajouter des informations supplémentaires afin de décrire les vibrations générées par les éléments haptiques. La figure 2.6f, en revanche, utilise des lignes simples avec une étiquette pour relier les objets réels de la photographie aux objets virtuels correspondants sur l'écran. 79 figures (12%) utilisent des flèches et des lignes pour grouper des éléments dont 71 figures utilisent des annotations textuelles supplémentaires.

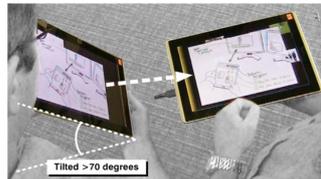
Groupement à l'aide d'indicateurs. Enfin, des indicateurs tels que des lettres ou des chiffres sont également utilisés. Ces indicateurs sont souvent placés sur l'objet lui-même ou à côté de celui-ci. La figure 2.6g utilise le chiffre 1 deux fois pour montrer que les parallélépipèdes bleus transparents et bleus opaques sont le même objet qui a été effectivement déplacé. La figure 2.6h utilise les lettres A, B, C et D pour relier des parties spécifiques de la figure à des informations supplémentaires en dehors de la figure, telles que des définitions dans la légende (non incluses). Nous avons trouvé 31 figures (5%) utilisant des indicateurs pour grouper des éléments dans notre jeu de données.



(a) Arrière-plan supprimé [24, Fig. 6]



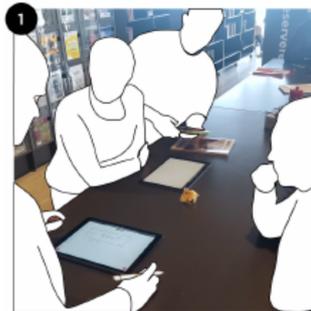
(b) Arrière-plan supprimé [102, Fig. 18]



(c) Arrière-plan grisé [103, Fig. 4]



(d) Anonymisation via le flou-tage [23, Fig. 5]



(e) Anonymisation via le remplissage [39, Fig. 1]

Figure 2.7 – Exemples de figures appliquant un prétraitement sur des photographies.

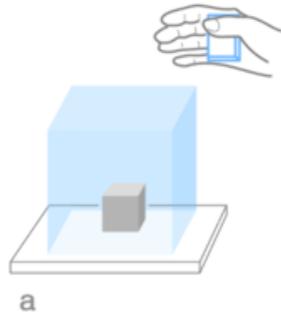
SS5 : Prétraitement de photographies

Deux stratégies, spécifiques aux photographies, consistent à effectuer un prétraitement sur l'image brute afin de mettre en valeur des éléments spécifiques ou d'anonymiser les utilisateurs.

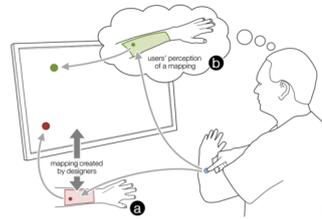
Suppression de l'arrière-plan. Nous avons trouvé 12 figures (2%) qui ont supprimé l'arrière-plan de la photographie et deux figures qui l'ont grisé dans notre jeu de données parmi les 387 figures utilisant des photographies. Bien que peu courante, cette stratégie consiste à supprimer ou à atténuer les couleurs de l'arrière-plan d'une photographie afin de faire ressortir les principaux éléments tels que les utilisateurs et le système interactif, tout en supprimant les détails inutiles qui pourraient distraire l'attention du lecteur ou engendrer un encombrement visuel de la figure. Les figures 2.7a et 2.7b sont de bons exemples de détournement d'utilisateurs et de dispositifs dans une photographie. La figure 2.7c [103] montre elle un arrière-plan grisé mettant en évidence les tablettes.

Anonymisation par le floutage ou le remplissage. Deux stratégies assez similaires peuvent être utilisées pour rendre les utilisateurs anonymes. La plus simple consiste à flouter les parties d'une figure où apparaissent les visages des utilisateurs (*cf.* figure 2.7d). Nous n'avons trouvé que 9 figures utilisant une telle technique d'anonymisation dans notre jeu de données. Un autre type d'anonymisation, non présent dans notre jeu de données, pourrait également être utilisé, comme par exemple remplir toutes les zones des figures montrant les utilisateurs à anonymiser avec une couleur. La figure 2.7e [39] est un bon exemple de cette technique. Buxton [26] et Greenberg *et al.* [51] font référence dans ce contexte à la technique de l'esquisse hybride : une partie de la photographie originale est conservée, tandis que d'autres parties sont esquissées par-dessus (dans ce cas-ci afin de les anonymiser). Outre l'anonymat, le croquis hybride peut être plus largement utilisé pour tout scénario d'interaction où il est important de fournir de riches informations sur le contexte de l'interaction représentée (par le biais de la photographie), et de superposer ces photographies avec des éléments dessinés illustrant une technique d'interaction, un dispositif ou une interface utilisateur.

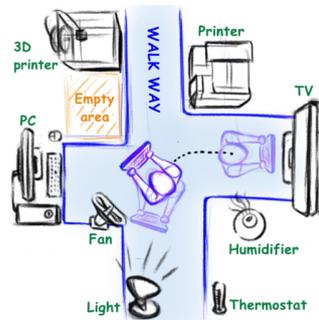
2.3 Stratégies d'interaction



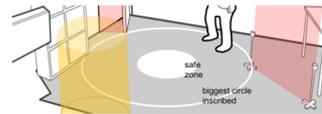
(a) Coloration de l'espace interactif 3D [141, Fig. 9]



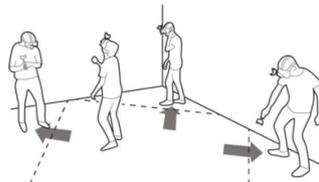
(b) Coloration de l'espace interactif 2D [14, Fig. 1]



(c) Coloration de l'espace interactif 2D [67, Fig. 8]



(d) Cercles délimitant l'espace interactif [156, Fig. 10]



(e) Lignes pointillées délimitant l'espace interactif [106, Fig. 4]



(f) Bordures transparentes de l'espace interactif en RV [156, Fig. 15]

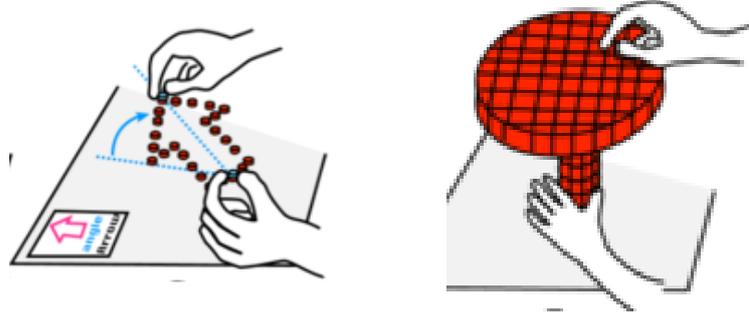
Figure 2.8 – Exemples de figures définissant un espace interactif.

SI1 : Définir l'espace interactif

La mise en évidence des dimensions physiques d'un espace d'interaction (par exemple, la zone bidimensionnelle ou un volume tridimensionnel) est utile pour montrer un espace restreint ou un environnement spécifique dans lequel une personne peut interagir.

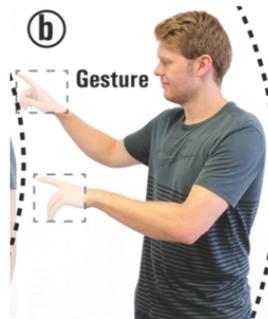
Colorer l'espace interactif. Appliquer une couleur spécifique à une partie de la figure est une stratégie qui est souvent utilisée pour mettre en valeur un espace interactif en 2D. Nous avons trouvé 60 figures (10%) utilisant cette stratégie dans notre jeu de données. Les figures 2.8b et 2.8c fournissent de bons exemples d'espaces interactifs colorés. La première montre une surface interactive contrainte sur la peau, limitée aux bras de l'utilisateur. La seconde montre la zone de déplacement possible de l'utilisateur dans un contexte d'interaction à distance. La mise en évidence d'espaces 3D en couleur peut également être réalisée à l'aide d'une forme géométrique 3D entourant ces espaces physiques. La figure 2.8a démontre cette technique où un cube bleu semi-transparent est utilisé pour représenter le volume 3D dans lequel une personne peut interagir avec le système.

Mettre en évidence les bordures de l'espace interactif. Au lieu de mettre directement en évidence l'espace interactif, certaines figures mettent plutôt l'accent sur les bordures de l'espace représenté. Une approche classique consiste à tracer des lignes sur la figure en tant que délimiteurs. Les figures 2.8d et 2.8e sont de bons exemples de l'utilisation de ces lignes. Alors que ces deux figures délimitent des espaces en RV, la première utilise des cercles blancs et la seconde des lignes pointillées. La figure 2.8f utilise à la place des panneaux colorés transparents intégrés dans la figure pour illustrer les bordures de chaque espace interactif individuel dans l'environnement global de RV.



(a) Mise en avant à l'aide de couleurs [139, Fig. 13]

(b) Mise en avant à l'aide de couleurs [140, Fig. 16]



(c) Mise en avant à l'aide de transparence [54, Fig. 4]



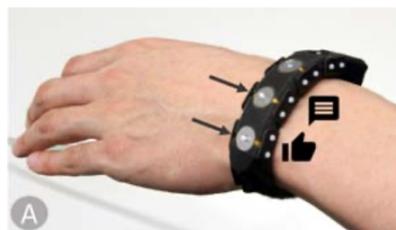
(d) Mise en avant à l'aide d'un rectangle [102, Fig. 18]



(e) Mise en avant à l'aide d'un cercle [23, Fig. 5]



(f) Mise en avant des contours de l'objet [91, Fig. 7]



(g) Mise en avant avec des flèches [118, Fig. 5]

Figure 2.9 – Exemples de figures mettant en évidence les éléments interactifs.

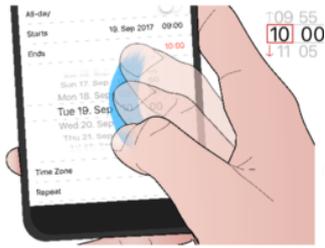
SI2 : Mettre en évidence les éléments interactifs

Ces stratégies mettent en évidence des éléments interactifs à l'aide de couleurs et de transparence, de contours, ou de flèches.

Utilisation de la couleur et de la transparence. La couleur est souvent utilisée pour mettre en évidence des éléments spécifiques, une stratégie observée dans 117 (19%) figures de notre jeu de données. Par exemple, la figure 2.9a met en évidence en rouge les objets tangibles, tandis que les lignes et les flèches utilisées pour illustrer l'angle et la direction de la manipulation actuelle sont affichées en bleu. Dans certains cas, un seul élément est coloré, tandis que le reste de la figure est grisé (*cf.* Figure 2.9b). La transparence peut également être utilisée pour mettre en évidence un élément interactif. Par exemple, la figure 2.9c utilise une couche de transparence uniquement pour les mains de l'utilisateur, afin de mettre en évidence le fait que l'utilisateur interagit avec ses mains.

Utilisation de contours. Une autre technique pour mettre en avant des éléments spécifiques dans une figure est d'utiliser des contours. Cette stratégie consiste à entourer les éléments de lignes afin de les mettre en valeur, souvent en utilisant différents styles (lignes en couleur, pleines ou pointillées). On distingue deux types de contours : les contours de forme simple (40 figures, 6%) et les contours de l'objet (15 figures, 2%). Entourer avec des contours de forme simple consiste à utiliser une forme géométrique pour entourer l'élément souligné. Par exemple, la figure 2.9d utilise un rectangle en lignes pointillées pour mettre en valeur le smartphone et la figure 2.9e utilise un cercle de couleur orange pour mettre en valeur la tablette. Entourer avec des contours de l'objet consiste à suivre le contour extérieur de l'objet mis en évidence. Par exemple, la figure 2.9f met en évidence la gâchette du contrôleur en utilisant un contour de couleur verte.

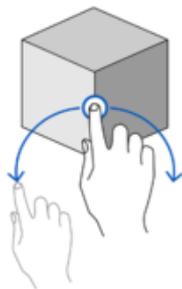
Utilisation de flèches. Si l'utilisation de flèches pointant vers un objet est une technique simple pour mettre en valeur un élément, il est surprenant qu'elle ne soit pas utilisée dans notre jeu de données. La figure 2.9g (extrait de l'article [118]) est un bon exemple de cette technique où les flèches mettent en évidence les aimants fixés au dispositif portable.



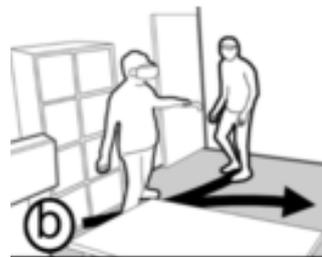
(a) Effet stroboscopique basé sur un dessin [36, Fig. 1]



(b) Effet stroboscopique basé sur une photographie [82, Fig. 3]



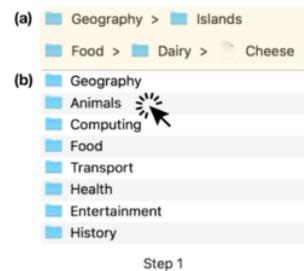
(c) Flèches de direction [109, Fig. 9]



(d) Flèches de trajectoire [156, Fig. 12]



(e) Chemin parcouru [68, Fig. 10]



(f) Onomatopée visuelle : illustrer le clic [98, Fig. 1]



(g) Onomatopée visuelle : illustrer la vitesse [100, Fig. 1]



(h) Onomatopée textuelle : illustrer la vibration [121, Fig. 1]

Figure 2.10 – Exemples de figures représentant un contenu dynamique dans une seule vignette.

SI3 : Représenter du contenu dynamique dans une seule vignette

Si les comportements dynamiques d'une figure peuvent être représentés à l'aide de plusieurs vignettes pour montrer les différentes étapes d'une séquence ou d'un mouvement, il existe également quelques techniques pour illustrer du contenu dynamique en une seule vignette. Dans cette section, nous décrivons brièvement ces techniques, et nous nous référons à McAweeney *et al.* pour une taxonomie spécifique sur la représentation des gestes dans les figures [109].

Effet stroboscopique. Les effets stroboscopiques consistent à dessiner les états et/ou positions précédents d'un élément en mouvement (*p.ex.* contrôleur, dispositif, partie du corps) en utilisant différentes couleurs, transparences ou styles de lignes. La figure 2.10a montre un effet stroboscopique basé sur un dessin illustrant le défilement du pouce sur un écran tactile. La figure 2.10b illustre un effet stroboscopique basé sur la photographie du corps d'un utilisateur en mouvement.

Lignes et flèches. Les lignes et les flèches sont des éléments simples qui peuvent être aussi utilisés pour montrer des actions dynamiques. En général, les lignes sont utilisées pour décrire le chemin parcouru par un objet (*p.ex.* chemin d'un dispositif, d'une partie du corps ou d'un curseur de souris) comme dans la figure 2.10e où le chemin parcouru par l'utilisateur est représenté par une ligne rose transparente sur la photographie. Les flèches sont elles plus souvent utilisées pour les directions et les trajectoires. La figure 2.10c montre les deux rotations possibles pour manipuler le cube à l'aide de deux flèches bleues. La figure 2.10d montre la trajectoire de l'utilisateur sur le sol de la pièce.

Onomatopées. Également décrits comme des métaphores visuelles de ce qui ne peut être vu (p. 128, [111]), les « onomatopées » sont un ensemble de petits détails ajoutés à une figure, directement inspirés des techniques utilisées dans les bandes dessinées, pour donner l'impression d'actions dynamiques. On en distingue deux types : les onomatopées textuelles et les onomatopées visuelles. Par exemple, de petits traits peuvent être utilisés derrière un objet pour montrer une certaine vitesse (Figure 2.10g) ou disposés en cercle autour d'un curseur de souris pour illustrer un clic (Figure 2.10f). La figure 2.10h, quant à elle, utilise des onomatopées textuelles pour illustrer les vibrations de la montre connectée.

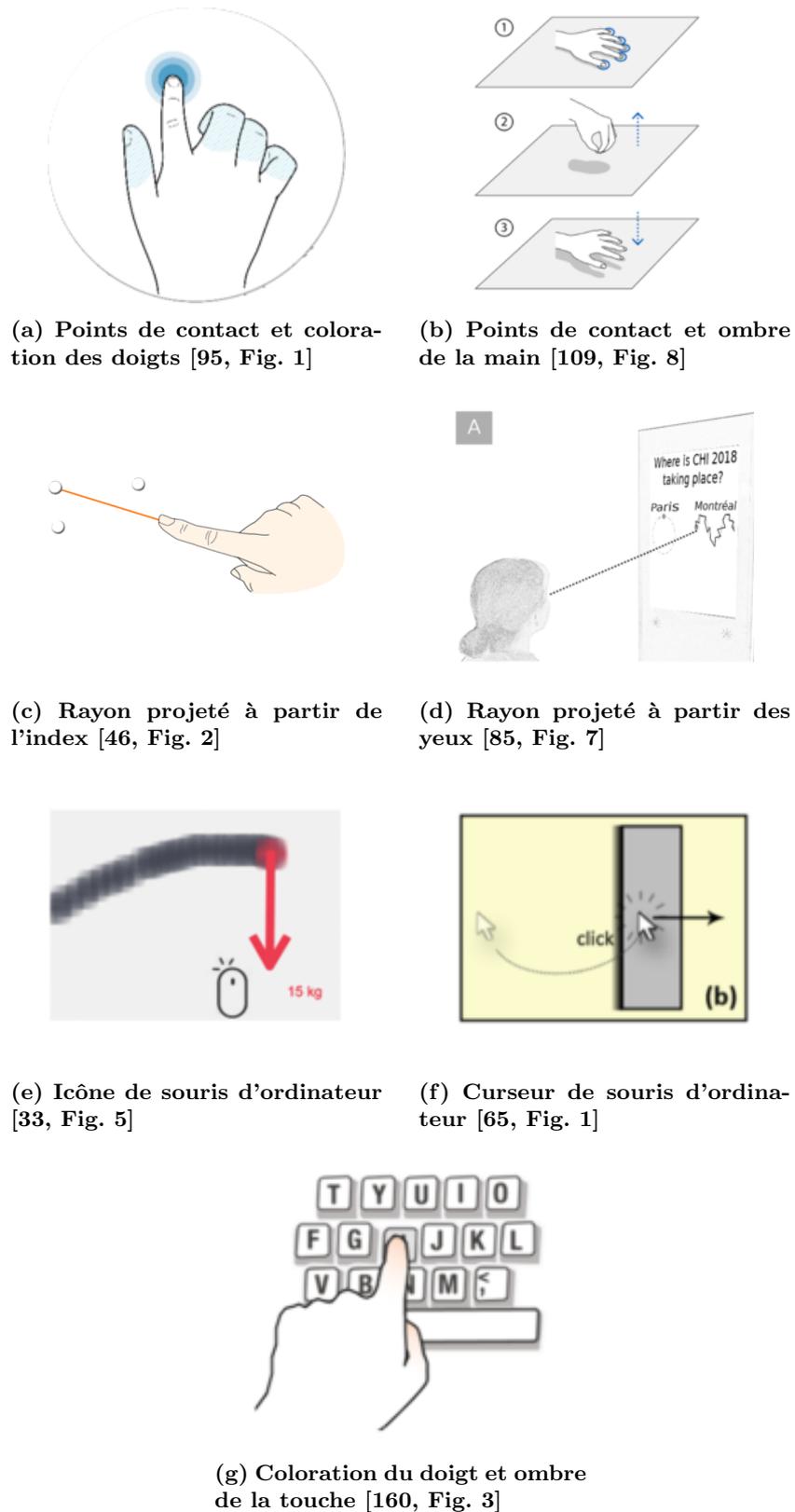


Figure 2.11 – Exemples de figures mettant en évidence les actions des utilisateurs.

SI4 : Mettre en évidence les actions des utilisateurs

Cette section présente des stratégies visant à mettre l'accent sur l'action des utilisateurs avec un système interactif.

Dessiner les points de contact. Une stratégie pour représenter le contact physique entre deux éléments consiste à mettre en évidence la zone de contact en dessinant une forme géométrique (généralement un disque coloré) comme couche intermédiaire entre les éléments. Cette technique est couramment utilisée pour représenter un contact tactile entre un doigt et une surface sensible au toucher (25 des 42 figures utilisant des points de contact illustrent l'interaction tactile). À titre d'exemple, les figures 2.11a et 2.11b utilisent des cercles bleus pour mettre en évidence les points de contact.

Montrer les rayons projetés. Des lignes ou des flèches peuvent être tracées pour montrer la projection de rayons (visibles ou non) de l'utilisateur (à partir de contrôleurs, de mains, de doigts ou des yeux) vers le système. Ces lignes de projection sont généralement affichées en utilisant une couleur ou un style spécifique. Par exemple, la figure 2.11c montre un rayon de couleur orange projeté à partir du doigt de l'utilisateur. De même, la figure 2.11d matérialise le regard de l'utilisateur à l'aide d'une ligne pointillée.

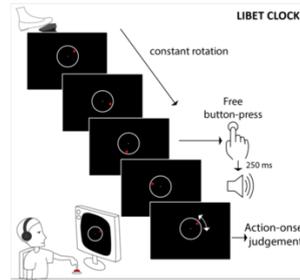
Jouer avec les ombres. L'utilisation de différents types d'ombres portées peut aider à illustrer la distance physique avec le système. Par exemple, la figure 2.11b n'affiche pas l'ombre de la main lorsqu'elle touche la surface, mais l'affiche lorsqu'elle la survole. Inversement, la figure 2.11g supprime l'ombre de la touche pressée tout en changeant sa couleur du blanc au gris clair.

Montrer le curseur. Ajouter explicitement un curseur système peut aider à mieux comprendre l'interaction, par exemple lorsque l'utilisation d'une souris est nécessaire. La figure 2.11e affiche une icône de souris intégrée dans la figure et la figure 2.11f montre le curseur de la souris. De plus, les deux figures utilisent une onomatopée visuelle de type *clic*, c'est-à-dire des petits traits disposés en cercle autour de l'icône, pour représenter une pression physique sur le bouton de la souris.

Colorer les éléments interactifs. Une couleur différente pour les éléments interactifs peut être utilisée pour mettre en évidence les éléments du système ou les parties des utilisateurs qui peuvent être utilisés pour interagir (par exemple les télécommandes, les contrôleurs, les doigts). Par exemple, la figure 2.11a utilise un doigt blanc (par opposition au bleu clair) pour montrer le doigt sur la surface tactile. La figure 2.11g utilise une couleur orange clair pour montrer le doigt qui appuie sur la touche **H** du clavier.



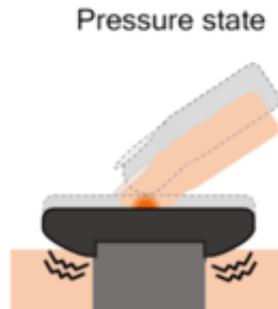
(a) Retour audio suggéré par les écouteurs [137, Fig. 1]



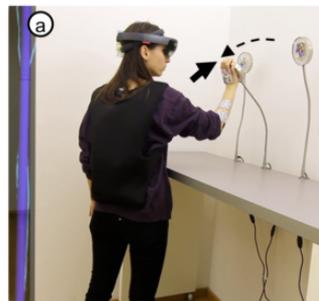
(b) Retour audio suggéré par l'icône sonore [104, Fig. 5]



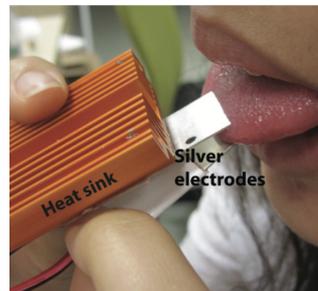
(c) Retour haptique suggéré par l'onomatopée textuelle [121, Fig. 1]



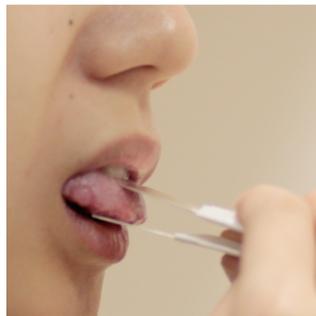
(d) Retour haptique suggéré par l'onomatopée visuelle [132, Fig. 1]



(e) Retour de force suggéré par les flèches [100, Fig. 12]



(f) Retour gustatif suggéré par le focus sur bouche [128, Fig. 4]



(g) Retour gustatif suggéré par le focus sur bouche [127, Fig. 2]

Figure 2.12 – Exemples de figures illustrant des retours non visuels.

SI5 : Illustrer les retours non visuels

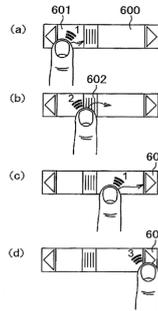
Les retours non visuels, tels que les retours sonores ou haptiques, peuvent être difficiles à représenter dans une illustration statique. Dans cette section, nous expliquons quelques stratégies utilisées pour surmonter cette difficulté en ajoutant des informations supplémentaires dans les figures.

Retour audio. 21 figures (3% de notre jeu de données) illustrent indirectement un retour sonore en montrant le dispositif utilisé (*p.ex.* des haut-parleurs ou des écouteurs) qui permet au lecteur de déduire qu'il y a une réponse audio fournie par le système interactif. Par exemple, la figure 2.12a montre un utilisateur jouant à un jeu de course avec des écouteurs sur la tête, ce qui suggère l'utilisation d'un retour sonore. Une autre stratégie consiste à ajouter une icône représentant le son dans la figure. C'est le cas de la figure 2.12b qui illustre une chronologie d'événements, avec une flèche de transition allant d'une pression de bouton à une icône sonore signifiant qu'une fois le bouton appuyé, un son est émis par le système.

Retour haptique. Le retour haptique tel que les vibrations ou le retour de force est illustré à l'aide de stratégies d'illustration dynamiques. Nous avons d'abord observé l'utilisation des onomatopées, mentionnées dans la stratégie d'interaction SI3. La figure 2.12c utilise le texte « *ITCH* » deux fois pour illustrer les vibrations de la montre connectée, tandis que la figure 2.12d utilise un effet de vagues. Des flèches sont également utilisées, en particulier pour le retour de force, où différents styles (épais, pointillés) sont utilisés pour représenter les différentes forces appliquées par l'utilisateur et le système (*cf.* figure 2.12e)

Retour gustatif. Bien que notre jeu de données ne contienne pas de figures représentant l'interaction impliquant le sens du goût, nous avons trouvé des documents sur l'interaction multisensorielle tels que [128, 127] afin d'en extraire une stratégie. Celle-ci consiste à montrer directement l'équipement utilisé, c'est-à-dire la bouche de l'utilisateur en gros plan à la 3^{ème} personne avec le dispositif interactif placé dans celle-ci montrant des électrodes en contact avec la langue, et suggérer ainsi un retour de goût (*cf.* figures 2.12f et 2.12g).

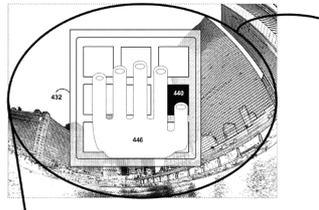
2.4 Stratégies de brevets



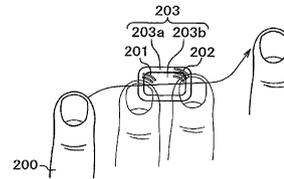
(a) Vignettes multiples [123, Fig. 10]



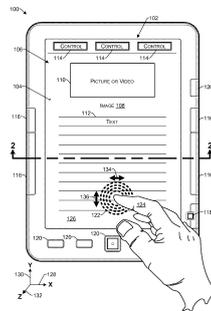
(b) Prétraitement de photographies [144, Fig. 1]



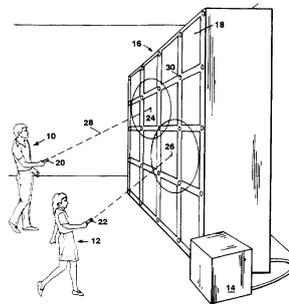
(c) Fusion de vignettes [120, Fig. 4]



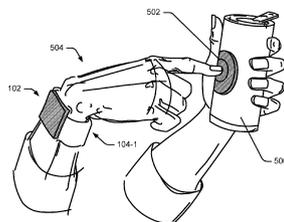
(d) Effet stroboscopique [123, Fig. 3]



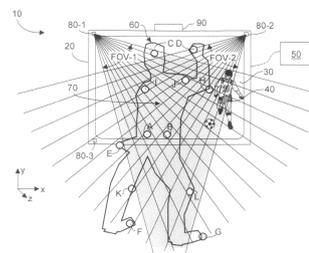
(e) Points de contacts [25, Fig. 1]



(f) Rayons projetés [73, Fig. 1]



(g) Éléments colorés [122, Fig. 5]



(h) Zone interactive colorée [126, Fig. 1A]

Figure 2.13 – Exemples de figures de brevets qui utilisent des stratégies d'illustration de l'interaction.

Stratégies pour les brevets

Nous l'avons vu dans l'état de l'art, les figures de brevets suivent des recommandations en termes d'aspect et de structure. Néanmoins, ces figures utilisent aussi les stratégies de structure et d'interaction présentées dans les précédentes sections.

Aspect visuel. Les figures extraites des brevets sont toutes réalisées en noir et blanc ou en niveaux de gris et le plus souvent au moyen de dessins au trait. Certaines figures de brevets intègrent des photographies dont les couleurs ont été converties en niveaux de gris (*cf.* figure 2.13b ou figure 2.13e). Le style de dessin va du schématique (Figure 2.13h) au réaliste (Figure 2.13f). Les interfaces utilisateur sont généralement dessinées dans le plan et sans perspective, comme le montre la figure 2.13e.

Identifiants. Les figures de brevets utilisent toutes des indicateurs spécifiques pour relier des éléments à des définitions extérieures à la figure. Selon les recommandations pour les figures de brevets, ces identificateurs sont toujours composés d'un nombre et d'une flèche/ligne pointant vers un élément de la figure.

Utilisation des stratégies de structure et d'interaction. Les figures de brevets utilisent également des stratégies similaires à celles que nous avons identifiées dans les figures extraites des actes de conférences en IHM. Nous présentons brièvement un ensemble de stratégies utilisées dans les figures de brevets que nous avons collectés. Différents points de vue sont aussi utilisés : du dessus (*cf.* figure 2.13a), 1^{ère} personne (*cf.* figures 2.13d et 2.13e) ou 3^{ème} personne (*cf.* figure 2.13f). Poupyrev *et al.* ont illustré une séquence d'interaction en utilisant deux stratégies dans leur figure [123] : plusieurs vignettes (*cf.* figure 2.13a) et un effet stroboscopique (*cf.* figure 2.13d). La figure 2.13c utilise la fusion de vignettes pour représenter le point du vue du système et de l'arrière-plan dans un contexte de RA. Les zones interactives peuvent également être mises en évidence à l'aide d'une couleur grise plus foncée (*cf.* figure 2.13g) ou en dessinant un champ de rayons provenant de la caméra pour définir la zone d'interaction, comme le montre la figure 2.13h. Enfin, les actions entre les utilisateurs et les systèmes sont également représentées à l'aide de points de contact pour l'interaction tactile (*cf.* figure 2.13e) ou de rayons pour l'interaction à distance (*cf.* figure 2.13f).

3 Outils pour le codage, la visualisation et l'exploration de la taxonomie

Comme mentionné précédemment dans le document dans les sections 1 et 2, nous avons développé trois outils qui ont facilité la tâche pour créer notre taxonomie et identifier des stratégies visuelles. Ces outils étaient essentiels pour permettre un codage et une révision/analyse efficaces du vaste jeu de données de 795 figures. En particulier, nous avons conçu et mis en œuvre deux outils dédiés au codage (*cf.* figures 2.14 et 2.15), et un outil supplémentaire pour explorer la taxonomie finale (*cf.* figure 2.16). Nous fournissons ces outils de manière open-source, pour aider les autres chercheurs de notre communauté à créer des taxonomies dans le futur. Ces trois outils sont disponibles à l'adresse suivante <https://ns.inria.fr/loki/taxonomy/>.



Figure 2.14 – L’outil de codage a été utilisé pour attribuer des codes aux figures. Il affiche la figure avec sa légende dans un widget central, deux widgets de notation à 5 étoiles pour entrer la confiance du codage et le score de la figure, ainsi qu’un widget type arborescence pour afficher la hiérarchie des codes et une zone de commentaires.

3.1 Outil de codage

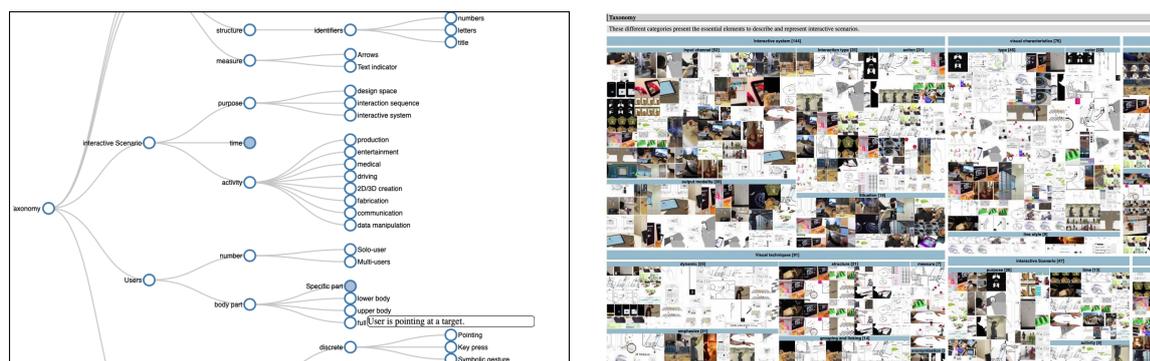
Tout d’abord, nous avons développé une application de codage avec PyQt (*cf.* figure 2.14). Elle affiche une seule figure à la fois ainsi que sa légende afin d’aider les codeurs à comprendre la figure affichée. Dans la partie droite de l’application, une arborescence affiche la hiérarchie des codes issus de la taxonomie. En survolant un code avec la souris, une info-bulle contenant la description du code s’affiche. Pour chaque code dans l’arborescence, une case à cocher est associée et permet de sélectionner ou de désélectionner le code pour la figure courante. Nous avons également ajouté la possibilité d’ajouter, de supprimer et de renommer des codes dans la hiérarchie, ce qui s’est avéré très utile pour les premières itérations de codage lorsque des modifications étaient nécessaires. Des raccourcis clavier

peuvent être utilisés pour interagir rapidement avec l'application et optimiser la tâche tels que `cmd`+`P` ou `cmd`+`N` pour afficher la figure précédente ou suivante, `←` pour donner le focus au widget d'arborescence des codes, etc.

Les catégories clés de la taxonomie (*c.-à-d.* des catégories où certains codes étaient systématiquement sélectionnés) sont mises en évidence par un fond rouge dans le widget d'arborescence lorsqu'aucun de leurs codes « enfants » n'a été sélectionné, puis le fond passe au vert lorsqu'au moins un code enfant est sélectionné. Un indicateur de couleur global situé dans le coin supérieur droit passe du rouge au vert lorsque toutes les catégories clés sont vertes. Le système de couleurs des catégories clés aide les codeurs à identifier rapidement quand une figure a été codée, c'est-à-dire quand au moins un code de chaque catégorie clé a été sélectionné.

Enfin, une zone de commentaires textuels et deux widgets de notation à 5 étoiles peuvent être utilisés par les codeurs pour donner un retour d'information sur leur confiance dans le codage et la notation subjective globale de la figure.

3.2 Outil de visualisation



(a) Visualisation de la taxonomie sous la forme d'une hiérarchie navigable.

(b) Visualisation de la taxonomie sous la forme d'une cartographie des figures.

Figure 2.15 – L'outil de visualisation a été utilisé pour naviguer dans la hiérarchie des codes, voir les différentes catégories, les codes, leur définition et des exemples de figures associées.

Nous avons aussi créé un outil de visualisation disponible en ligne³ en utilisant la bibliothèque graphique D3 [20]. Cet outil propose deux types de visualisation pour la taxonomie. Une vue arborescente simple et extensible, permettant de cliquer sur chaque nœud de la hiérarchie des codes (*cf.* figure 2.15a) et une vue cartographique, permettant d'observer des exemples de figures pour chaque code de notre taxonomie (*cf.* figure 2.15b). Les deux outils de visualisation permettent de lire la description de chaque code et ont été automatiquement mis à jour à chaque itération de codage. Cet outil en ligne a aidé

3. <https://ns.inria.fr/loki/IllustrationTaxonomy>

les codeurs à maîtriser les différents codes de la taxonomie, et a également été utilisé comme support visuel pour les discussions sur les différentes étapes d'amélioration de la taxonomie.

3.3 Outil d'exploration

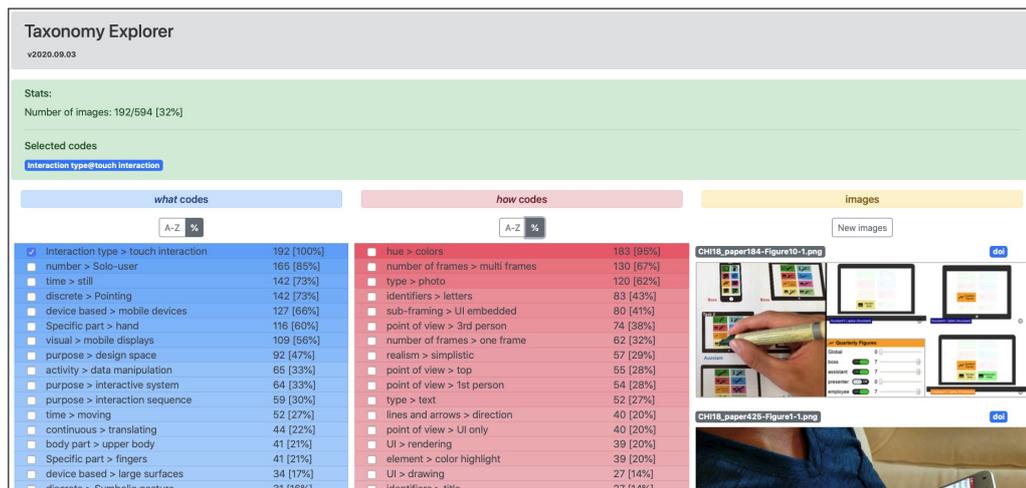


Figure 2.16 – L’outil d’exploration, *Taxonomy Explorer*, est une application web permettant de naviguer dans les données codées. En sélectionnant les codes *quoi* et *comment* (respectivement dans les listes bleues et rouges), un filtre est appliqué sur l’ensemble des données pour obtenir le nombre de figures taguées avec ces codes. Les deux listes peuvent être triées par ordre alphabétique ou par fréquence et sont mises à jour en temps réel lors de la sélection de codes. La colonne de droite présente des exemples de figures correspondant à la sélection courante.

Enfin, nous avons développé un outil logiciel d’exploration, *Taxonomy Explorer*, aussi disponible en ligne⁴, qui permet de parcourir les codes utilisés dans la taxonomie, de les trier et de les filtrer selon leur fréquence d’utilisation, et d’afficher les figures associées aux codes sélectionnés (cf. figure 2.16). L’idée principale de cet outil est de pouvoir sélectionner des codes spécifiques et d’observer l’impact de la sélection sur les figures et la fréquence d’utilisation des autres codes. Cet outil est composé de quatre parties principales : une bannière horizontale verte, et trois colonnes verticales (une bleue, une rouge et une affichant des figures). Les colonnes bleues et rouges affichent respectivement la liste des codes *quoi* et *comment* de notre taxonomie.

Chaque ligne de ces colonnes est une structure HTML contenant une case à cocher pour (dé)sélectionner le code, le nom du code (concaténé avec le nom de sa catégorie parente), et des chiffres indiquant le nombre d’occurrences et le pourcentage d’apparition du code dans l’ensemble des figures correspondantes aux codes actuellement sélectionnés. Lors de la (dé)sélection d’un code, l’ensemble des figures comportant tous les codes sélectionnés

4. <https://ns.inria.fr/loki/IllustrationTaxonomy>

est automatiquement calculé (c'est-à-dire l'intersection), et l'interface se met à jour pour refléter la (dé)sélection du code (le code apparaît ou disparaît de la bannière verte). Les lignes dans les colonnes bleues et rouges ont également une couleur de fond différente, un fond plus foncé correspondant à un pourcentage plus élevé d'apparition du code dans l'ensemble des figures correspondantes. Par défaut, si aucun code n'est sélectionné, toutes les figures sont incluses dans la sélection. Afin de permettre différentes stratégies d'exploration des codes, les deux listes peuvent être triées dynamiquement par ordre alphabétique ou par fréquence d'apparition. Le bandeau horizontal vert affiche la liste des codes actuellement sélectionnés. En cliquant sur l'un d'entre eux, le code est supprimé de la sélection actuelle et sa case à cocher correspondante est décochée dans la colonne bleue ou rouge. Enfin, la dernière colonne affiche une vue dynamique défilante avec dix exemples de figures qui ont été taguées avec les codes actuellement sélectionnés. Un bouton permet de faire un nouveau tirage des dix figures exemples.

Comme mentionné dans les sections précédentes, cet outil a été utilisé pour observer les tendances globales des codes au sein du jeu de données, ainsi que pour identifier des stratégies d'illustration.

Discussion

Discussion sur la méthodologie

Nous avons utilisé une approche empirique [89] qui consiste à construire une théorie à partir de zéro et une analyse de données qui est, dans notre cas, une analyse manuelle des figures de notre jeu de données. Une partie du processus de codage utilisé pour construire la taxonomie a été réalisée par plusieurs codeurs, jusqu'à ce que nous observions un score de concordance stable entre eux. Après cela, un seul codeur a effectué le processus de codage pour les figures restantes, puis a analysé manuellement les figures codées pour identifier les stratégies. Plusieurs outils logiciels ont été conçus et mis en œuvre afin de simplifier ce processus et d'assister les codeurs au cours de ces étapes manuelles. Cela étant dit, plusieurs alternatives ont été envisagées et testées au cours de ce travail, notamment en ce qui concerne la mesure de concordance entre les codeurs et l'extraction des stratégies.

Mesure de la concordance entre les codeurs

À notre connaissance, il n'existe pas de méthodologie standard pour calculer un score de concordance dans un processus de codage. En effet, les métriques couramment utilisées [44, 86] pour mesurer la concordance entre évaluateurs ont été conçues pour des tâches de catégorisation où un objet peut être classé dans une seule et unique catégorie, alors que dans notre cas, à chaque figure peuvent être affectés plusieurs codes. Pour surmonter ce problème, nous avons utilisé une adaptation commune du Kappa de Fleiss et de l'Alpha

de Krippendorff [44, 86] à un contexte de codage en calculant un score de concordance pour chaque figure au lieu de l'ensemble des figures. À cet égard, nous avons considéré à l'intérieur d'une figure chaque code comme une entité à attribuer à l'une et une seule des catégories exclusives : *Sélectionné* ou *Non-sélectionné*. Le score de concordance global a ensuite été calculé comme la moyenne du score de concordance de chaque figure. Toutefois, cette méthode reste une adaptation de ce pour quoi elle a été conçue et, à ce titre, doit être soigneusement prise en compte lorsqu'elle est utilisée ou comparée dans d'autres contextes.

Exploration manuelle des figures codées

Nous avons décidé de mettre en place l'outil d'exploration pour soutenir l'exploration manuelle des figures codées afin d'extraire des stratégies. Avant cela, nous avons envisagé d'explorer le jeu de données codées en utilisant une Analyse en Composantes Principales (ACP) ou des méthodes de regroupement telles que K-means, mais nous avons réalisé que ces techniques n'étaient pas adaptées. Tout d'abord, les méthodes de *clustering* regroupent généralement tous les objets (ici les figures) d'un jeu de données en un nombre prédéterminé de clusters, chaque objet ne pouvant appartenir qu'à un seul et unique cluster. Bien qu'un regroupement flou [40] tel que le *fuzzy c-means* [16] puisse surmonter ce problème, les méthodes algorithmiques seraient encore inadaptées sur notre jeu de données, tant en termes de nombre de codes que de diversité des figures. En effet, ces algorithmes sont très sensibles aux valeurs « extrêmes » (ici, des figures taguées avec des codes peu attribués à d'autres figures) et notre jeu de données comporte un certain nombre de valeurs extrêmes en raison des types d'interaction sous-représentés (*p.ex.* la RV était beaucoup plus représentée que l'entrée vocale). Nettoyer les données pour éliminer les figures extrêmes ou pondérer l'impact des codes aurait pu être une solution, mais cela déplace le problème et nécessite de trouver des critères de nettoyage au risque de perdre trop d'information. Plus important encore, nous sommes également intéressés par l'analyse des valeurs extrêmes. En effet, bien qu'un des objectifs de ce travail soit de trouver des stratégies couramment utilisées pour une majorité de figures, il n'est pas moins intéressant de trouver des stratégies tout aussi pertinentes, mais peu utilisées, car le scénario interactif représenté dans la figure est rare. À titre d'exemple, les codes *pieds*, *entrée vocale*, *anonymisation par le floutage*, *arrière-plan grisé* ou *arrière-plan flouté* et plus de dix autres codes sont utilisés moins de dix fois dans l'ensemble du jeu de données, et pourtant, la plupart d'entre eux font partie des stratégies que nous avons identifiées et présentées dans la section 2. En fin de compte, nous pensons que notre méthodologie d'exploration était la bonne voie à suivre. Elle repose sur un codage systématique et rigoureux des figures avec une exploration manuelle du jeu de données codé, assistée par un logiciel conçu pour cette tâche. L'exploration du jeu de données par un seul codeur a par ailleurs déjà permis d'extraire plusieurs stratégies utilisées pour illustrer l'interaction.

Limitations de l'étude

Nous nous sommes concentrés sur l'identification de stratégies « de haut niveau » pour créer des représentations visuelles pour des scénarios d'interaction, en ignorant les stratégies « de bas niveau » telles que la manière dont les variables visuelles (symboles, styles de lignes, etc.) pourraient être utilisées. Nous avons décidé de nous concentrer sur les premières, car des recherches importantes ont déjà été menées sur les secondes [28] à tel point qu'il est possible de trouver des ouvrages pédagogiques afin de mieux comprendre comment utiliser correctement ces variables visuelles [146]. Pourtant, plusieurs stratégies (*p.ex.* SS3 ou SS4) bénéficieraient d'une compréhension plus approfondie de la sémiologie des graphiques [15] ou des lois de la Gestalt [84] pour optimiser leur mise en œuvre. En effet, si les stratégies extraites peuvent servir de lignes directrices pour illustrer un scénario d'interaction spécifique, la sémiologie et les lois de la Gestalt indiqueraient si certaines variables visuelles sont adaptées [28], ou si certains types de juxtaposition de vignettes ou d'objets suggèrent par exemple correctement que les éléments appartiennent à un même groupe.

Une autre limite de notre travail est que nous nous sommes concentrés sur des figures illustrant un scénario interactif sur l'utilisation des outils/systèmes/prototypes, en ignorant d'autres figures qui peuvent être fréquemment utilisées dans les documents en IHM, comme les tableaux de données. Une fois de plus, nous avons décidé de nous concentrer sur un domaine largement sous-exploré plutôt que de nous concentrer sur des types de figures qui ont déjà fait l'objet d'études approfondies [57, 146, 134].

Nous avons également limité notre étude au cas des figures *statiques*, qui restent le principal support visuel pour décrire les scénarios interactifs dans les documents de recherche. Nous avons donc délibérément décidé d'ignorer les figures dynamiques [53, 76, 78] et interactives [107] même si certaines peuvent être trouvées dans des parutions récentes en IHM [47, 53]. Ces figures ne sont généralement pas prises en charge nativement par la plupart des formats utilisés pour communiquer sur les scénarios interactifs (généralement des documents de recherche de type PDF), et même lorsqu'elles sont utilisées, elles nécessitent souvent une figure statique équivalente [53, 107].

Conclusion

Dans ce travail, nous avons établi une taxonomie de figures illustrant des scénarios interactifs et identifié des stratégies qui sont utilisées (et peuvent être réutilisées) lors de la production de ces figures. La taxonomie a été élaborée selon un processus en deux étapes qui consistait à produire des codes à partir de zéro par un groupe de codeurs, et à itérer pour affiner ces codes et leurs définitions correspondantes. Cette approche a été intrinsèquement guidée par notre jeu de données qui était composé de figures illustrant des scénarios interactifs extraites des actes de quatre grandes conférences en IHM. Bien

que potentiellement incomplet, ce jeu de données a permis d'aller au-delà de ce que l'on savait déjà sur les figures de scénarios interactifs, et de produire une taxonomie couvrant un large spectre d'interactions illustrées dans des figures statiques. Comme l'illustre le tableau 2.2, notre taxonomie intègre des catégories et des codes déjà proposés par la littérature, mais s'étend au-delà de celle-ci avec des éléments qui n'étaient pas encore ou seulement partiellement pris en compte par les travaux précédents.

Le codage complet de ce jeu de données donne ensuite des indications intéressantes sur l'illustration des scénarios interactifs, comme le montrent les statistiques sommaires qui se trouvent dans la section 1.3. Ces illustrations utilisent des mises en page, des structures et des points de vue différents, ce qui montre qu'il n'existe pas une seule approche pour illustrer l'interaction et que les lignes directrices et les stratégies utilisées sont différentes en fonction de ce qu'un auteur peut vouloir illustrer. L'exploration de ce jeu de données – soutenue par notre outil d'exploration – nous a permis d'identifier des stratégies de structure et d'interaction. Nous pensons que ces stratégies généralisées peuvent servir de lignes directrices et d'inspiration aux chercheurs et aux étudiants en IHM pour produire la meilleure représentation visuelle possible de leur scénario interactif.

Notre taxonomie et notre classification pourront probablement être étendues à l'avenir, et des stratégies supplémentaires pourraient être extraites en reproduisant notre méthodologie, soit avec un jeu de données différent, soit avec des objectifs différents. Nous avons choisi de fournir nos outils et notre jeu de données complet en ligne afin de permettre l'exploration créative de notre taxonomie (ou de différentes taxonomies à l'avenir). Premièrement, notre outil de visualisation permet l'exploration de notre taxonomie, de naviguer dans la hiérarchie de nos codes, de lire les descriptions des codes et d'observer des exemples de figure pour chaque code. Deuxièmement, notre outil d'exploration peut être utilisé pour explorer notre jeu de données de figures codées et observer des tendances ou découvrir de nouvelles stratégies. Il est en effet important de noter que nous nous sommes concentrés sur l'illustration de l'interaction sur des représentations visuelles statiques, mais notre taxonomie pourrait être étendue à d'autres domaines, tels que les manuels d'instructions où des stratégies spécifiques sont utilisées pour aider l'utilisateur à comprendre les différentes opérations [5].

3

Esquisse : un outil de production d'illustrations reposant sur de l'agencement de scènes 3D

Sommaire

Introduction	74
1 Enquête sur les processus de création de figures à base de tracés	76
2 Esquisse : processus créatif et interaction	78
2.1 Choisir son modèle de scène et ses objets	80
2.2 Ajout d'interfaces utilisateur	80
2.3 Faciliter la manipulation des objets 3D	81
2.4 Représentation du mouvement	86
3 Esquisse : algorithme de rendu vectoriel	89
3.1 État de l'art sur les techniques de rendu non photoréaliste vectoriel	89
3.2 Algorithme de rendu non-photorealiste vectoriel	90
4 Étude : illustration de techniques d'interaction	95
4.1 Méthode d'expérimentation	95
4.2 Résultats	97
Discussion	100
Limitations dues à l'utilisation de Blender	100
Modifier les pratiques actuelles	100
Conclusion	101

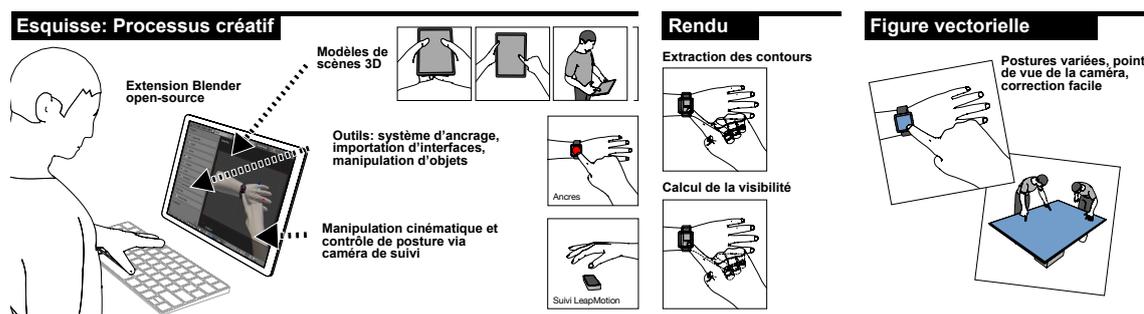


Figure 3.1 – Processus créatif d'*Esquisse* : (gauche) faciliter la mise en place de scènes 3D grâce aux modèles de scène prédéfinis, au système de points d'ancrage, à l'importation directe d'interfaces utilisateur, et à la manipulation de postures via l'utilisation de cinématique inverse et caméra de suivi, (centre) algorithme de rendu non photoréaliste d'*Esquisse*, et (droite) rendu vectoriel final de la scène 3D.

Introduction

Nous avons présenté dans le chapitre 2 notre taxonomie de figures illustrant un scénario interactif. Les stratégies d'illustration identifiées dans ce chapitre peuvent être utilisées pour créer des illustrations de tout type : des photographies, des captures d'écrans ou encore des figures à base de tracés. Alors que prendre une photographie ou une effectuer capture d'écran est une tâche qui peut être effectuée en quelques secondes en cliquant sur un bouton physique ou virtuel, le processus de création d'une figure à base de tracés est quant à lui beaucoup plus complexe. C'est pourquoi nous nous intéressons dans ce chapitre aux procédés pour créer ce type de figure.

Les figures à base de tracés sont des figures illustratives statiques créées pour se concentrer sur les aspects essentiels d'une situation, en supprimant les détails non nécessaires et en limitant la représentation graphique aux contours les plus importants des objets et personnes représentés (par exemple, la Figure 3.2 montre six figures à base de tracés). Les figures à base de tracés, même augmentées ou améliorées avec des annotations graphiques en superposition (telles que du texte, des flèches ou des formes géométriques), tendent à minimiser l'encombrement visuel et sont efficaces pour décrire l'essence d'un scénario interactif [26].

C'est grâce à leur clarté visuelle et la possibilité de se focaliser sur un contexte particulier que les figures à base de tracés sont fréquemment utilisées au sein de la communauté IHM, notamment pour illustrer de nouvelles techniques d'interaction ou de nouveaux systèmes interactifs. Par conséquent, on les retrouve dans de nombreux supports tels que les articles, les présentations ou les posters. À titre d'exemple, les actes de la conférence internationale ACM UIST¹ de ces trois dernières années [1, 2, 3] comportent plus de 25% d'articles publiés utilisant au moins une figure à base de tracés pour illustrer de nouveaux gestes, donner un aperçu d'une installation spécifique d'un système ou encore illustrer un

1. ACM UIST – <https://uist.acm.org/>



Figure 3.2 – Exemples de figures à base de tracés : (1) geste à un doigt [55], (2) chemin tracé par la main [115], (3) geste à une main effectué sur smartphone [64], (4) interaction à deux mains sur tablette avec stylet [99], (5) personne interagissant avec une table tactile [87], et (6) effet stroboscopique illustrant la gestuelle d'un corps [35].

espace de conception.

Nous avons vu dans le chapitre 1 qu'une technique utilisée pour créer ces figures est le décalquage manuel d'une photographie qui produit les contours principaux des dispositifs et personnes présents dans l'illustration souhaitée [35]. Notre enquête auprès de chercheurs qui utilisent cette technique de décalquage montre qu'il s'agit d'une tâche manuelle pénible, qui demande un temps de travail important. De plus, lorsque des changements mineurs sont apportés aux personnes, aux postures et dispositifs représentés dans l'illustration, il faut alors éventuellement recommencer tout le processus de dessin, ce qui montre les limites de cette technique. Ce problème est d'ailleurs flagrant dans les cas où plusieurs postures de main doivent être produites ou lorsque les utilisateurs veulent changer le point de vue de leur illustration. Comme évoqué dans le chapitre état de l'art, la recherche sur la production d'illustrations s'est principalement concentrée sur l'aide à la production de nouveaux types d'illustrations telles que des illustrations interactives ou animées [77, 79] ou alors des illustrations dans un domaine spécifique (*p.ex.* la représentation des gestes dans l'espace [35]). Toutefois, malgré l'usage fréquent des figures à base de tracés illustrant des scénarios interactifs et les difficultés rencontrées pour les produire, il n'existe toujours pas d'outil spécifique créé pour faciliter leur production.

Dans ce chapitre, nous défendons l'hypothèse que l'agencement de scènes 3D est une alternative viable pour faciliter la production de figures statiques illustrant des scénarios interactifs. Plus précisément, nous présentons *Esquisse*, un outil open-source implémentant un système d'agencement 3D pour produire des figures à base de tracés (voir la figure 3.1). En particulier, l'idée principale d'*Esquisse* est de s'appuyer sur la flexibilité d'un logiciel de modélisation 3D pour s'intéresser à la mise en place de scènes qui sont alors rendues en tant que figures vectorielles. Ces images vectorielles pouvant être éditées dans un logiciel d'édition vectorielle par la suite. Nous proposons des modèles 3D ainsi que des outils et des techniques pour rendre le processus créatif accessible à chaque type d'utilisateur, qu'importe son expérience avec les logiciels de modélisation 3D.

Le processus implémenté par *Esquisse* fonctionne comme suit. Les utilisateurs ouvrent une scène 3D à partir d'un des modèles de scène fournis. Si besoin, ils peuvent modifier directement la position et la posture des objets en utilisant les techniques d'interaction

fournies par *Esquisse* qui simplifient la manipulation de ces objets 3D. Par exemple, les utilisateurs peuvent modifier la posture d'une main en utilisant le système Leapmotion², ou en ajoutant un *point d'ancrage* qui accrochera le doigt d'une main 3D à ce point d'ancrage et modifiera de manière automatique la posture de la main 3D si celui-ci est déplacé grâce à la cinématique inverse intégrée. Bien sûr, ils ont toujours la possibilité d'ajouter d'autres avatars ou objets 3D dans la scène si nécessaire. Les utilisateurs positionnent ensuite la caméra dans l'espace 3D selon leurs besoins. Ils peuvent aussi intégrer des captures d'interfaces sur différents supports. De manière optionnelle, ils peuvent ajouter un effet stroboscopique ou des flèches pour illustrer le mouvement des objets et personnages. Un simple clic sur le bouton *Rendre* appellera alors l'algorithme de rendu d'*Esquisse* qui génère un fichier vectoriel de la scène intégrant toutes les captures d'interface ajoutées. Des exemples de rendus réalisés avec *Esquisse* sont visibles à travers les sections de ce chapitre.

La structure du chapitre est la suivante. Après avoir présenté une enquête que nous avons réalisée auprès de chercheurs en IHM sur les procédés qu'ils utilisent pour la création de figures, nous décrivons en détail les techniques d'interaction mises en place par *Esquisse* ainsi que leur implémentation pour ensuite présenter l'évaluation de l'outil et des figures produites par les participants. Enfin nous discuterons des avantages et des limitations de l'outil proposé.

1 Enquête sur les processus de création de figures à base de tracés

Un ensemble de techniques et outils actuels permettant de produire des figures à base de tracés ont été présentés dans l'état de l'art (chapitre 1 section 2). Néanmoins, afin de mieux comprendre les outils et le processus créatif que les chercheurs utilisent actuellement pour produire des figures à base de tracés et leurs limitations, nous avons contacté 12 chercheurs en IHM que nous avons identifiés comme utilisateurs de figures à base de tracés dans leurs articles. Nous leur avons demandé de répondre à une brève enquête leur demandant de (1) décrire leur processus actuel de création de ces figures, (2) dresser la liste des outils matériels et logiciels qu'ils utilisent, et (3) noter sur une échelle de Likert de 1 à 7 la tâche de production de figures à base de tracés en termes de difficulté, durée et pénibilité de la tâche.

Un seul chercheur n'a pas répondu à l'enquête parce que « Malheureusement, [il n'est] pas capable de produire ce type de figure par lui-même et a toujours dû demander de l'aide à quelqu'un » (R12). Les 11 autres chercheurs ont eux, répondu à l'enquête. Sept ont déclaré qu'ils utilisent *toujours* ou *presque toujours* des figures à base de tracés pour

2. LeapMotion – <https://www.leapmotion.com/>

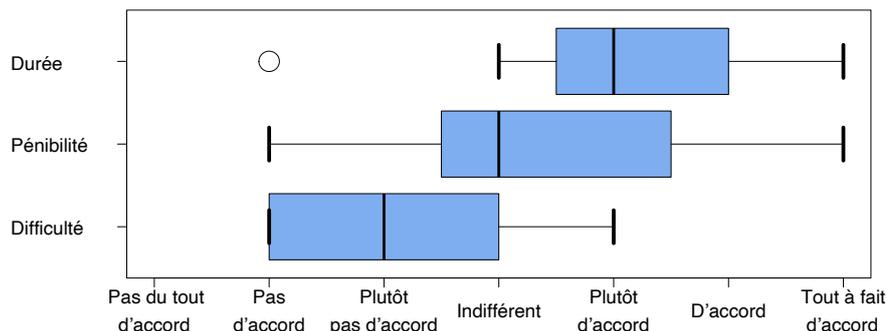


Figure 3.3 – Évaluation subjective de la difficulté, de la durée et de la pénibilité pour la production de figures à base de tracés.

illustrer leurs documents de recherche tandis que deux ont déclaré le faire *souvent* et deux autres le faire *parfois*.

Comme l'illustre la figure 3.3, si les participants ne considèrent pas la tâche de produire des figures à base de tracés comme particulièrement difficile, ils la trouvent longue et relativement fastidieuse. Il est important de noter que ces résultats, surtout en termes de pénibilité et de difficulté, sont peut-être biaisés par le fait que nous avons spécifiquement contacté des chercheurs qui étaient déjà familiers avec les figures à base de tracés et identifiés comme des utilisateurs fréquents. La durée et l'ennui signalés ne sont pas surprenants, compte tenu du processus créatif que les participants ont déclaré utiliser. À titre d'exemple, R5 a répondu « c'est l'ensemble des sous-tâches » qui rend la tâche difficile, et R6 que « le processus est vraiment long pour obtenir la figure que vous voulez ». Aucun participant n'a par ailleurs déclaré avoir dessiné la figure à base de tracés à partir de rien, c'est-à-dire sans utiliser un support visuel tel que des photos qu'il a décalquées. Les processus créatifs décrits étaient relativement similaires et s'appuient globalement sur la technique de décalquage de photographies déjà présentée, à savoir : (1) mise en scène réelle, (2) transfert de la photographie dans un ordinateur, (3) décalquage numérique en utilisant un logiciel d'édition et (4), ajout d'informations supplémentaires en superposition.

Bien que le déroulement du processus créatif déclaré soit généralement similaire d'un participant à l'autre, des différences entre les participants étaient également notables. En règle générale, alors que neuf participants ont déclaré produire *toujours* ou *très souvent* des figures à base de tracés dans un format vectoriel, un seul a déclaré les produire *toujours* dans un format pixelisé ainsi qu'un autre a déclaré produire *l'un ou l'autre*.

Un participant interrogé a insisté sur le fait qu'un problème critique du processus créatif est que la figure produite « n'est pas tout à fait correcte » (R1). Il est parfois possible de modifier manuellement les contours, afin de ne pas avoir à répéter tout le processus (R6). Cependant, cela n'est pas toujours possible, par exemple lorsque la figure représente une position différente de la main ou un cadrage différent de l'appareil photo. Dans ce cas,

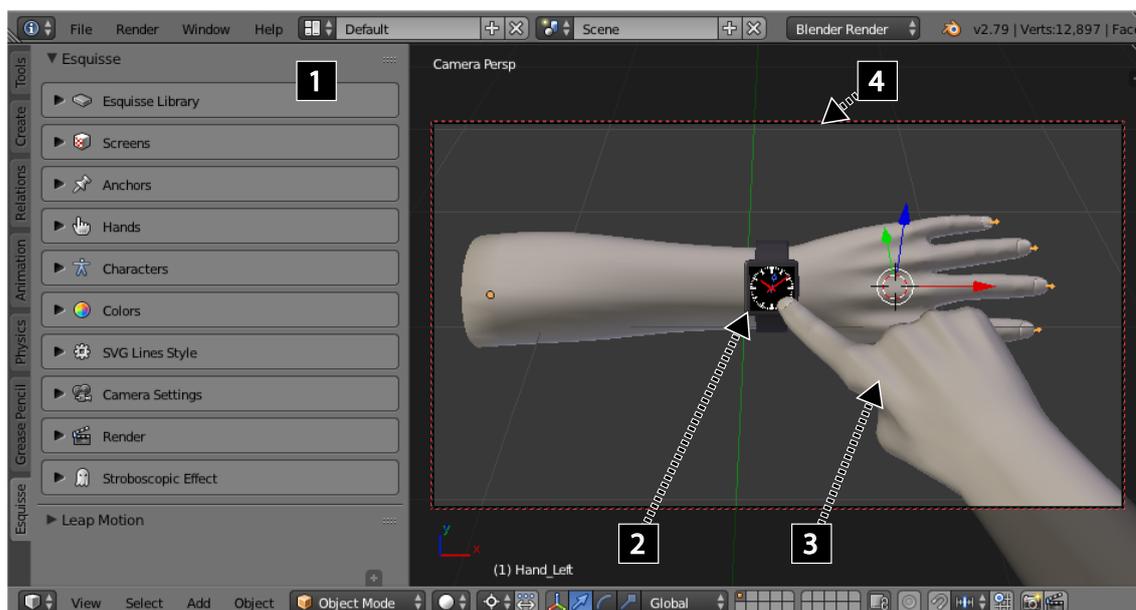


Figure 3.4 – Blender avec l’extension *Esquisse*. Tous les contrôles spécifiques à *Esquisse* se trouvent dans un panneau latéral dédié situé sur le côté gauche de l’interface (1). Le côté droit montre la vue 3D de la scène avec plusieurs objets de la bibliothèque *Esquisse* tels qu’une SmartWatch avec son interface graphique (2) et des modèles de main (3), ainsi que le cadre de la vue actuelle de la caméra (4).

l'utilisateur doit « prendre une nouvelle photo et recommencer tout le processus » (R1).

Certains participants ont également fourni des informations complémentaires, expliquant d'autres solutions qu'ils ont expérimentées pour produire des figures à base de tracés. Typiquement les techniques utilisées par les participants étaient principalement des techniques basées à partir d'un modèle, comme présenté dans le chapitre état de l'art : la recherche de cliparts en ligne, Google AutoDraw³ ou encore la vectorisation automatique d'images dans un logiciel d'édition. Pour cette dernière technique, R7 a par ailleurs ajouté qu'il a « essayé des méthodes de traçage automatique, mais [a] fini par faire beaucoup de travail de correction ».

Les résultats de cette enquête ont motivé notre travail en vue de mettre au point un outil facilitant le processus long, fastidieux et, selon les compétences existantes en matière de dessin, difficile de création d'illustrations de figures à base de tracés.

2 Esquisse : processus créatif et interaction

Esquisse utilise la mise en scène 3D pour faciliter la création de figures à base de tracés. Dans cette section, nous expliquons comment notre outil et nos techniques d'interaction proposées facilitent le processus de création de ce type de figures.

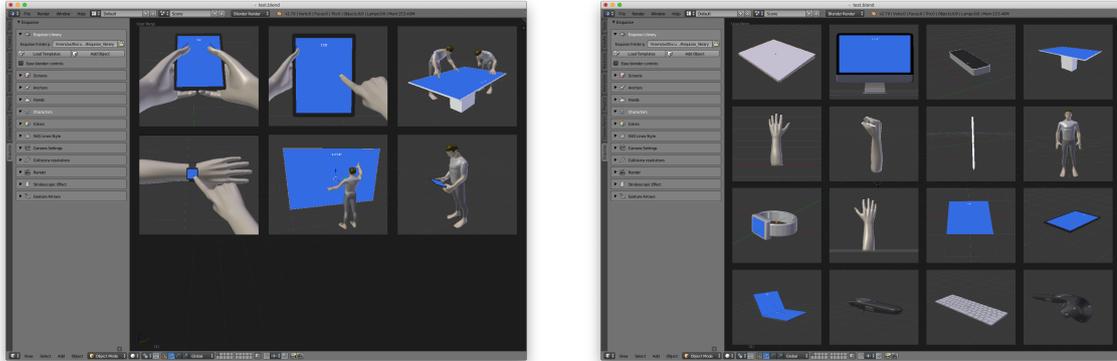
3. Google AutoDraw – <https://www.autodraw.com>

Pour utiliser *Esquisse*, l'utilisateur démarre Blender avec l'extension *Esquisse*⁴ précédemment installée. Nous avons choisi Blender car c'est un logiciel gratuit et multiplateforme largement utilisé pour modéliser, animer et rendre des scènes 3D. Cependant, en tant qu'environnement de modélisation 3D complet, Blender fournit des commandes et outils avec de nombreux degrés de liberté qui ne sont pas nécessaires pour la production d'illustrations, c'est pourquoi nous avons remplacé la plupart des fonctions par défaut par des techniques d'interaction plus simples spécifiquement mises en œuvre pour *Esquisse*. Ces techniques se trouvent dans un panneau latéral dédié, situé sur le côté gauche de l'interface Blender (*cf.* figure 3.4, numéro 1).

Pour créer une illustration graphique vectorielle avec *Esquisse*, les utilisateurs mettent en place une scène 3D en utilisant l'un des modèles de scène prédéfinie comme point de départ ou en ajoutant manuellement des objets 3D à la scène courante. Pour les objets comportant un écran (par exemple un smartphone), ils peuvent éventuellement charger un fichier externe contenant une capture d'écran de l'interface, qui est automatiquement rendue dans le cadre de l'écran de l'appareil sélectionné (*cf.* figure 3.6). Ils positionnent et alignent ensuite les objets en utilisant les techniques d'interaction spécifiques d'*Esquisse* conçues pour faciliter la manipulation des objets articulés (et en particulier pour simplifier la modification des postures des mains), mais ils peuvent toujours utiliser les commandes par défaut de Blender s'ils le préfèrent. Ensuite, ils orientent la fenêtre de visualisation de la caméra vers le point de vue souhaité. S'ils le désirent, ils peuvent ajouter un effet stroboscopique ou des flèches pour illustrer le mouvement d'un doigt, par exemple. Enfin, ils rendent la scène pour créer la figure à base de tracés sous la forme d'un fichier SVG vectoriel. En option, ils peuvent modifier ou ajouter des éléments graphiques en utilisant n'importe quel logiciel d'édition de graphiques vectoriels comme Inkscape ou Adobe Illustrator. Notez que ces étapes ne doivent pas nécessairement être effectuées dans cet ordre précis. Dans les sections suivantes, nous détaillons chaque étape du processus créatif d'*Esquisse*.

4. *Esquisse* – <https://ns.inria.fr/loki/EsquisseBlender>

2.1 Choisir son modèle de scène et ses objets



(a) Modèles de scène

(b) Objets

Figure 3.5 – Menus d'*Esquisse* permettant de sélectionner un modèle de scène prédéfinie (a) ou d'ajouter des objets à la scène courante (b).

Nous avons identifié dans notre taxonomie que de nombreuses figures à base de tracés ne varient que dans les détails. L'enquête que nous avons menée auprès de chercheurs en IHM a aussi montré que les figures à base de tracés ne sont la plupart du temps pas créées à partir de zéro. C'est pourquoi *Esquisse* fournit des modèles de scène 3D avec des ensembles préétablis de personnes et d'objets utilisés dans ces scénarios typiques.

Chaque modèle de scène définit une scène 3D dans Blender dont les objets sont chargés dans l'espace 3D actuellement ouvert, par exemple deux mains tenant un téléphone, une personne interagissant avec une smartwatch, ou deux personnes autour d'une table interactive. Les modèles de scène peuvent être facilement sélectionnés dans une galerie montrant une miniature pour chaque scène disponible (cf. figure 3.5a). De nouveaux modèles de scène peuvent être ajoutés en enregistrant une scène et en l'ajoutant à la bibliothèque *Esquisse* (c.-à-d. en copiant le fichier dans un dossier spécifique).

De la même manière, les utilisateurs peuvent compléter une scène 3D en y ajoutant de nouveaux objets (par exemple, des téléphones, des avatars) à partir de la bibliothèque d'objets d'*Esquisse* (cf. figure 3.5b). Des objets 3D personnalisés peuvent évidemment aussi être ajoutés à la scène à l'aide de la fonction de base d'importation de Blender.

2.2 Ajout d'interfaces utilisateur

Esquisse permet aux utilisateurs d'insérer des interfaces utilisateur (ce qui est le cas dans les figures de la stratégie SS3 de notre taxonomie) sur des objets 3D comportant un *écran*. Les objets courants avec écran (smartphones, tablettes, téléviseurs, montres intelligentes et tablettes) sont déjà fournis dans *Esquisse* avec un *écran* défini.

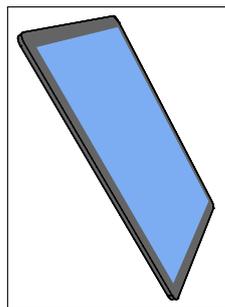
D'un point de vue technique, un *écran* est un plan en 3D, redimensionnable, sur lequel

une interface utilisateur (définie dans un fichier externe) peut être importée et affichée comme une texture 3D dans la scène. Les coordonnées du plan sont relatives à l'objet parent (s'il en a un), et il est généralement placé juste devant le maillage de l'objet parent afin qu'il reste visible (et éviter les artefacts de rendu dans le *depth buffer*). Afin de faciliter l'interaction des objets à écran, la sélection des écrans est par défaut désactivée dans la scène afin de sélectionner directement l'objet parent. Enfin, l'orientation de l'écran, et donc de l'interface, est déterminée grâce à l'ordre des quatre vertex du plan 3D qui est identique pour chaque plan. Il est possible d'ajouter un *écran* à tout objet 3D directement dans *Esquisse*, il suffit pour cela d'ajouter un objet *écran* dans la scène, le placer et le redimensionner comme souhaité, puis de sélectionner l'objet parent et valider la parenté (via le menu contextuel ou via le raccourci clavier `ctrl` + `P`).

Les *écrans* sont automatiquement détectés dans la scène par *Esquisse* et listés dans la partie correspondante dans le panneau latéral d'*Esquisse*. Ce panneau latéral permet par ailleurs d'ouvrir un explorateur de fichiers et d'associer une interface utilisateur à l'écran correspondant.



(a) Interface vectorielle 2D définie dans un fichier SVG externe et créée à partir d'un autre logiciel.



(b) Rendu vectoriel par *Esquisse* d'un modèle 3D de smartphone sans interface embarquée dans le rendu.



(c) Rendu vectoriel par *Esquisse* d'un modèle 3D de smartphone avec une interface embarquée dans le rendu, prenant en compte la perspective de l'écran du smartphone.

Figure 3.6 – Illustration du procédé d'inclusion automatique d'interfaces utilisateurs vectorielles sur les objets écrans dans *Esquisse*.

2.3 Faciliter la manipulation des objets 3D

Des objets simples comme les smartphones ou les tablettes peuvent être positionnés dans la scène 3D grâce au manipulateur 3 axes déjà disponible dans Blender pour la translation, la rotation et la mise à l'échelle des objets. Toutefois, lorsqu'il s'agit d'objets complexes comme la main ou le corps d'une personne, il ne suffit pas de manipuler la position globale, la rotation et l'échelle de l'objet de manière successive ou combinée, car ces modèles sont composés d'un ensemble de sous-objets (par exemple, les os pour un modèle de main avec armature) dont la position et l'orientation, souvent dépendantes entre

les sous-objets, doivent être modifiées pour chacun d'eux (*cf.* figure 3.7). Par exemple, pour un modèle de main 3D, il est nécessaire de manipuler l'armature d'un doigt de l'intérieur vers l'extérieur, car modifier l'orientation des métacarpes entraînera une rotation pour les phalanges. *Esquisse* fournit des modèles 3D avec armature qui prennent en compte les relations enfant-parent entre les os. Cela dit, sachant la difficulté de contrôler des rotations successives, modifier la posture d'un corps ou d'une main tout en conservant une apparence visuelle cohérente peut être un processus difficile et long pour les utilisateurs novices [4, 150]. Dans ce qui suit, nous présentons des techniques visant à simplifier cette tâche.

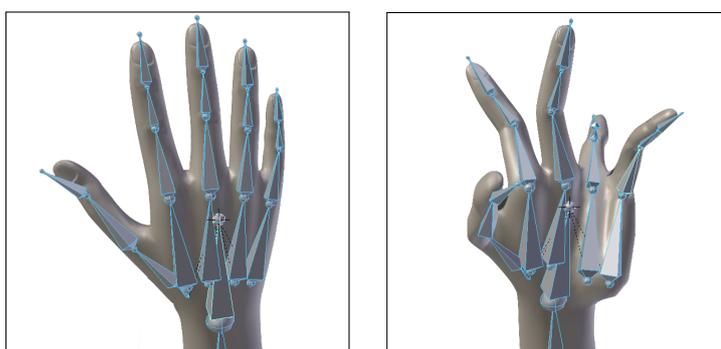


Figure 3.7 – Contrôler la rotation de chaque os d'un modèle 3D de main peut s'avérer être une tâche très compliquée. La posture de la main droite a été obtenue par les rotations successives des différents os de la main (surlignés en bleu dans les images) via le manipulateur de rotation intégré à Blender. Cette manipulation peut très vite conduire à une posture non naturelle de la main.

2.3.1 Système de points d'ancrage

Pour faciliter la manipulation d'objets à armature, nous avons introduit dans *Esquisse* un système d'ancres (ou points d'ancrage). Nous définissons une ancre comme un point virtuel sur un objet 3D qui peut être utilisé pour contraindre la posture d'un autre objet 3D (*p.ex.* un modèle de main). Par exemple, les ancres permettent aux utilisateurs de spécifier un point de contact sur un écran, d'y attacher un doigt d'un modèle de main, puis de modifier l'emplacement de ce point de contact tout en préservant les contraintes cinématiques de la main.

Créer un point d'ancrage et le relier à un objet. Pour ajouter un point d'ancrage, l'utilisateur rentre dans le mode *point d'ancrage* puis clique sur l'objet souhaité, ce qui positionne une nouvelle ancre à l'endroit où l'utilisateur a cliqué sur l'objet. La position 3D de l'ancre dans la scène est déterminée grâce à un lancer de rayon à partir des coordonnées souris, l'orientation de l'ancre est quant à elle identique à celle du polygone du maillage touché par le rayon lancé (*cf.* figure 3.8a).

Une fois l'ancre créée, l'utilisateur peut *lier* un point prédéfini d'un objet 3D à celle-ci

dans le panneau latéral d'*Esquisse*. Les points prédéfinis ont été manuellement ajoutés aux divers modèles 3D à armature fournis par *Esquisse* et correspondent actuellement au bout des doigts de nos modèles de main et d'avatar, mais pourraient être étendus à tout autre objet. La définition d'un *lien* contraint alors la position d'un bout de doigt donné à une ancre donnée.

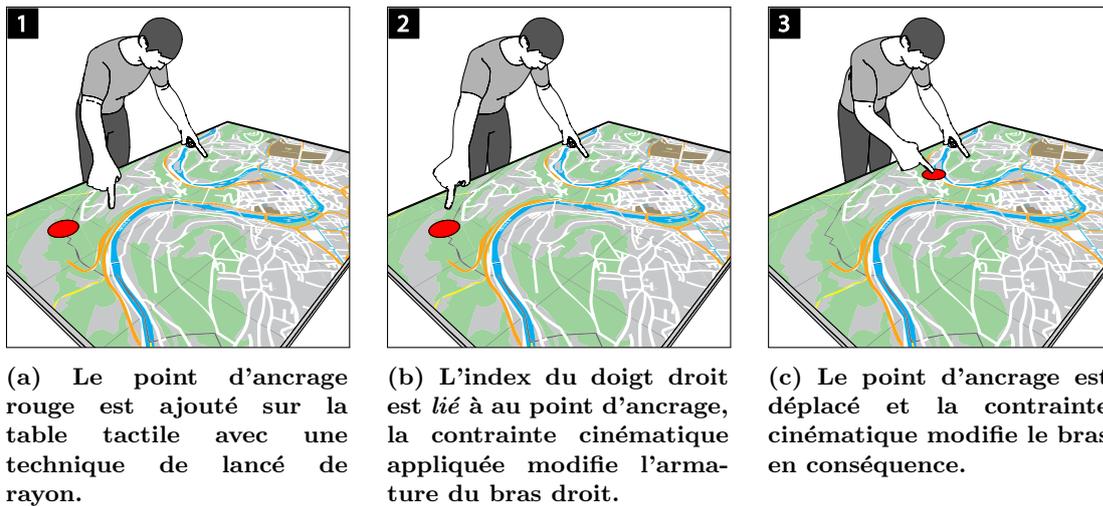


Figure 3.8 – Illustration de l'utilisation du mécanisme de points d'ancrage, de la pose d'une ancre à la modification de l'armature via des contraintes cinématiques.

Si le bout du doigt choisi n'est pas encore en contact avec l'ancre, le modèle de l'objet 3D qui contient le bout du doigt est déplacé de manière à ce que le bout du doigt soit en contact avec l'ancre avec une orientation par défaut du tangage et du roulis du doigt basée sur les résultats de Goguey et al. [48], sinon le modèle 3D n'est pas déplacé. Dans tous les cas, une nouvelle contrainte cinématique inverse est appliquée à partir de la position du dernier os du doigt choisi jusqu'à la position de la racine de son armature (*cf.* figure 3.8b). De manière générale, pour une chaîne d'os donnée qui définit une armature, le solveur de contraintes cinématiques inverses de Blender calcule la position et la rotation pour chaque os intermédiaire de la chaîne pour satisfaire les contraintes de position des premiers et derniers os. De ce fait, l'utilisateur ne se préoccupe que de la position de l'ancre pour définir des postures complexes d'armature.

Manipuler les ancres ou l'objet à armature. Les ancres peuvent être déplacées dans l'espace en utilisant le manipulateur 3 axes par défaut. Si le bout d'un doigt est lié à une ancre et que l'utilisateur la déplace alors le modèle de main réagira en fonction du mode de déplacement spécifié par l'utilisateur dans le panneau latéral d'*Esquisse* (figure 3.8c). Si le mode de déplacement est réglé sur *toute la main*, alors tous les os de la main peuvent adopter une nouvelle posture lorsque l'ancre est déplacée (*cf.* figure 3.9b). Si le mode de déplacement est réglé sur *doigt seulement*, le déplacement de l'ancre ne mettra

à jour que la position des quatre premiers os du doigt (phalanges et métacarpe) liés sans changer la position du bras, en se basant sur la même contrainte de cinématique inverse (cf. figure 3.9c). Bien sûr, plusieurs ancres peuvent être déplacées en même temps.

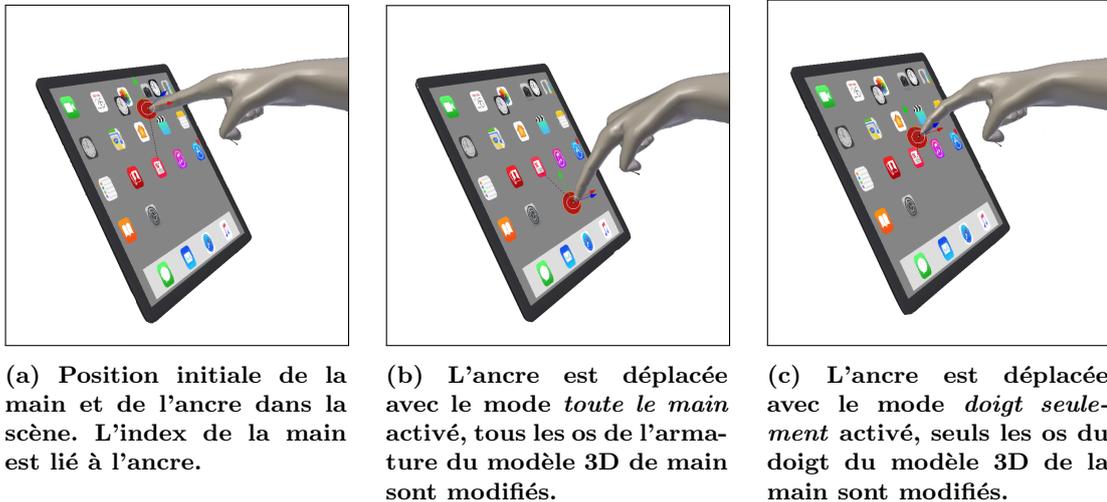


Figure 3.9 – Illustration des deux modes de déplacement de la main dans la scène 3D.

Lorsque les bouts des doigts sont ajustés aux endroits désirés, la pose de la paume de la main peut aussi être ajustée à l'aide des outils de manipulation fournis par Blender (tels que le manipulateur de rotation à 3 cercles, ou la trackball virtuelle). Le solveur de contraintes met alors à jour les contraintes définies afin que les bouts de doigts restent à leur emplacement. Enfin, l'utilisateur peut également déplacer le modèle de main tout en appuyant sur la touche `alt` pour déplacer les ancres et le modèle de main.

Ces techniques permettent d'illustrer rapidement des interactions complexes entre le toucher et les gestes, telles que des gestes de pincement et d'extension, comme illustré figure 3.8.

2.3.2 Flexion et extension des doigts

Alors que le système d'ancrage permet de définir facilement une posture de main ou d'avatar en attachant le bout des doigts aux ancres, la manipulation des doigts non ancrés reste toujours difficile [4, 150], c'est pourquoi *Esquisse* fournit deux autres contrôles spécifiques pour modifier les postures de la main.

Inspirés par Achibet *et al.* qui contrôle la pose d'une main virtuelle à l'aide de curseurs sur une tablette [4], nous avons ajouté des potentiomètres pour manipuler la flexion de chaque doigt de notre modèle de main. Toutes les mains et avatars dans la scène sont automatiquement détectés et listés dans le panneau latéral d'*Esquisse* (cf. sous-menus *Hands* et *Characters*, figure 3.4). Un potentiomètre est alors disponible pour chaque doigt et peut prendre une valeur entre 0 (doigt fermé) et 1 (doigt étendu). La modification de

la valeur d'un curseur met à jour la flexion du doigt correspondant dans la scène 3D en temps réel en ajustant l'orientation de chaque os du doigt.

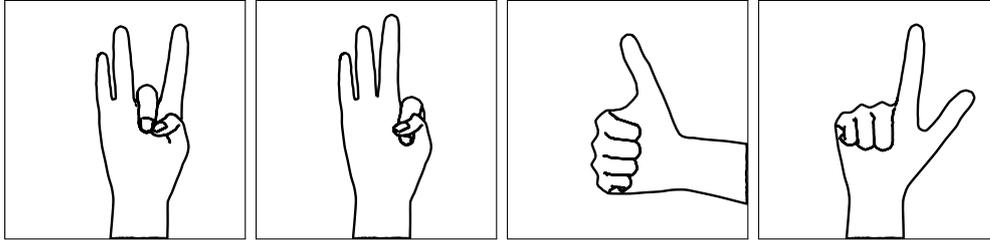


Figure 3.10 – Reproduction d’une partie de la figure 6 dans l’article de Lin et al. [97] réalisée avec *Esquisse* en manipulant uniquement les valeurs des curseurs pour les doigts de la main droite.

Enfin, nous avons ajouté la possibilité de modifier la posture de la main à l’aide d’une caméra de suivi de main LeapMotion. Cette technique utilise les coordonnées fournies par l’API de suivi de LeapMotion⁵ pour modifier la posture et l’orientation du modèle 3D de la main sur la base d’une démonstration. Pour cela, l’utilisateur sélectionne les mains à contrôler (une gauche et une droite au maximum) et entre en mode LeapMotion en cliquant sur le bouton approprié dans le panneau *Esquisse*, entrant ainsi en mode de suivi de la main. Ensuite, il positionne sa main dans le volume de travail du LeapMotion et fait une démonstration de la posture de la main souhaitée, ce qui met à jour en temps réel la posture du modèle 3D. Nous avons choisi d’extraire uniquement la rotation de chaque os de la main du LeapMotion pour ne contrôler que la pose de l’armature dans Blender. Enfin, quand les postures sont correctes, l’utilisateur les valide en appuyant sur la touche `[esc]` pour quitter le mode LeapMotion. En utilisant une approche similaire, une caméra Microsoft Kinect⁶ pourrait être utilisée comme méthode alternative pour modifier les postures du corps entier, comme Chi *et al.* le font dans DemoDraw [35].

Ces deux techniques permettent aux utilisateurs de modifier rapidement les postures des objets main, par exemple pour décrire un vocabulaire gestuel dans l’air (voir la figure 3.10 pour notre reproduction de certaines des postures de la main illustrées Figure 6 dans l’article de Lin et al. [97]).

2.3.3 Téléportation au point de vue de la caméra

Effectuer un rendu d’une scène 3D nécessite un objet caméra permettant de définir le point de vue du rendu, qu’il faut distinguer de la fenêtre de visualisation qui donne un aperçu de la scène 3D courante en temps réel. Par défaut, Blender considère une caméra comme n’importe quel autre objet, et peut donc être déplacée et orientée dans la scène 3D. Le point de vue de la fenêtre de visualisation, quant à lui, est facilement modifiable à l’aide

5. LeapMotion SDK – <https://developer.leapmotion.com/sdk/v2/>

6. Microsoft Kinect – <https://en.wikipedia.org/wiki/Kinect>

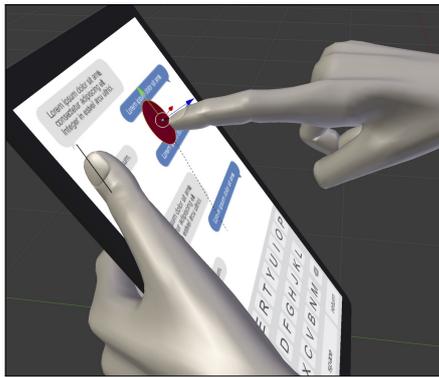
de mouvements latéraux effectués sur la fenêtre via une souris ou un pavé tactile. Sachant que les rotations 3D peuvent être difficiles à réaliser, nous avons mis en place une fonction spécifique qui peut, à la demande, téléporter la caméra de rendu à la position actuelle de la fenêtre de visualisation de la scène. De cette façon, l'utilisateur peut simplement déplacer la fenêtre de visualisation vers le point de vue souhaité et placer la caméra dans cette position en fonction du scénario interactif représenté (voir la stratégie structurelle SS2 dans notre taxonomie). De plus, au lieu de suivre une approche par essais et erreurs, en effectuant plusieurs rendus avant de trouver le bon point de vue, *Esquisse* propose de visualiser la scène 3D à partir de la caméra, c'est-à-dire de faire correspondre le point de vue de la fenêtre de visualisation avec celui de la caméra en tenant compte de l'aspect-ratio et de la longueur de la focale, qui peuvent être réglés directement à partir du panneau latéral d'*Esquisse* (cf. sous-menu *Camera Settings*, figure 3.4). Le passage de la vue *scène* à la vue *caméra* se faisant simplement via des boutons dans le panneau latéral d'*Esquisse*.

2.4 Représentation du mouvement

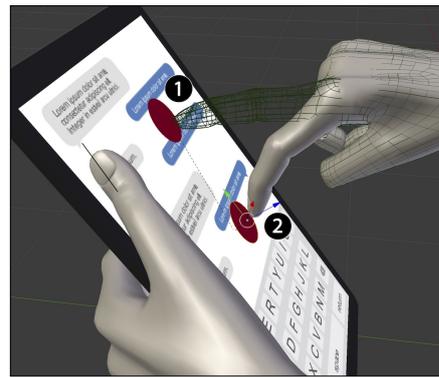
Esquisse permet aux utilisateurs de générer des figures qui illustrent une dynamique en utilisant un effet stroboscopique ou des flèches [110, 111] (voir la stratégie d'interaction SI3 dans notre taxonomie). Ces deux techniques, les plus utilisées, permettent d'illustrer le mouvement des objets dans le temps.

2.4.1 Effet stroboscopique

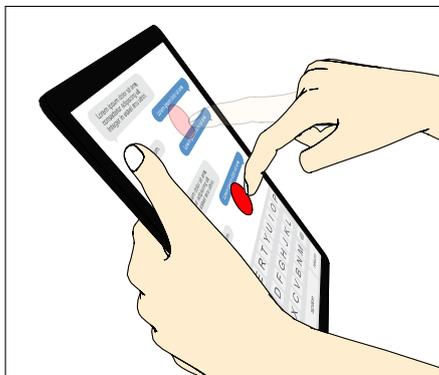
L'effet stroboscopique permet de rendre chaque étape d'un mouvement avec une transparence croissante. Une fois la scène mise en place (cf. figure 3.11, en haut à gauche), l'utilisateur peut entrer en mode *effet stroboscopique* en cliquant sur le bouton approprié dans l'interface. Ce mode permet de définir différentes *images clés*, chacune enregistrant les positions de tous les objets dans de la scène. Lors de la création d'une nouvelle image clé, *Esquisse* détecte automatiquement les objets qui se sont déplacés et crée une version « fantôme » des objets correspondants à la position initiale (c'est-à-dire crée une copie des objets de l'image clé). Par exemple, les figures 3.11a et 3.11b montrent la création automatique d'une version fantôme de l'ancre et de la main liée lorsque l'ancre est déplacée de la position (1) vers la position (2). Les objets fantômes peuvent également être manipulés directement dans la scène comme tout autre objet afin de procéder à des rectifications si nécessaire. Le nombre d'images clés est quant à lui à la préférence de l'utilisateur, mais *Esquisse* peut aussi générer un nombre défini de sous-images clés entre deux images clés en utilisant une interpolation linéaire entre les objets déplacés. Par exemple, la figure 3.11d montre l'ajout de deux étapes supplémentaires à l'effet stroboscopique entre la position (1) et la position (2) de l'ancre. Ces étapes intermédiaires sont calculées en interpolant les matrices 4x4 de départ et d'arrivée pour chaque objet ou os d'armature déplacé.



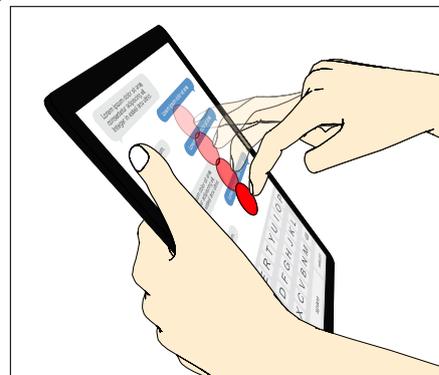
(a) L'index de la main droite est attaché à une ancre.



(b) Dès que la main est déplacée dans le mode stroboscopique (1), une version « fantôme » est créée à sa position initiale (2).



(c) Rendu vectoriel de la scène par *Esquisse*.



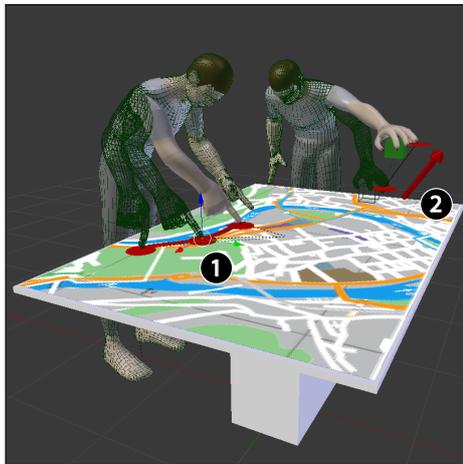
(d) Rendu vectoriel de la scène par *Esquisse* avec deux sous-images clés supplémentaires interpolées entre les deux images clés.

Figure 3.11 – Illustration de la création d'une figure avec un effet stroboscopique : manipulation dans la scène 3D et rendu des figures vectorielles.

2.4.2 Flèches de mouvement

En plus de l'effet stroboscopique, *Esquisse* permet aussi de générer des flèches dans l'espace 3D afin d'illustrer le mouvement des objets. Le processus d'utilisation des flèches est similaire à celui de l'effet stroboscopique. L'utilisateur déplace les objets souhaités et crée des versions « fantômes » à travers les différentes images clés.

L'ajout d'une flèche est alors simple, l'utilisateur sélectionne un objet et choisit de lui attribuer une flèche. *Esquisse* parcourt alors tous les objets de la scène, et cherche toutes les versions « fantômes » de cet objet. Sachant que les images clés sont ordonnées chronologiquement, *Esquisse* peut donc récupérer la position dans l'espace de l'objet à travers les différentes images clés et en déduire sa trajectoire. Un objet flèche est alors ajouté dans la scène. Cet objet est constitué d'un maillage suivant la courbe de Bézier définie par les différentes positions. Ce maillage peut être planaire, c'est-à-dire formé par



(a) Des flèches de mouvement sont *liées* aux ancrages et affichent leur trajectoire, que ce soit sur dans le plan de table interactive (1) ou dans les airs (2).



(b) Rendu vectoriel de la scène, la flèche dans le plan est rendue sous forme planaire, la flèche dans les airs sous la forme d'un cylindrique.

Figure 3.12 – Illustration de l'utilisation de flèches pour représenter le mouvement.

deux vertex espacés à intervalle régulier sur la courbe, pour définir une flèche dans le plan, ou cylindrique, formé par une succession de cercles, pour définir une flèche dans l'espace (*cf.* figure 3.12, où la flèche de mouvement du personnage de gauche est planaire et la flèche de mouvement du personnage de droite et cylindrique). Son épaisseur et sa rotation par rapport à l'axe directionnel peuvent également être modifiées par l'utilisateur. Les déplacements dans la scène des objets liés à une flèche mettent automatiquement à jour la trajectoire de cette dernière. Enfin, les flèches sont rendues par *Esquisse* comme de simples objets de l'espace 3D.

Toutes les techniques présentées dans cette section (l'ajout d'interfaces, le système d'ancrage, les potentiomètres, le LeapMotion, la téléportation de caméra et les effets de mouvement) sont conçues pour permettre aux utilisateurs inexpérimentés en matière de logiciels de modélisation 3D de mettre rapidement en place une scène avec des modèles 3D. Plus important encore, les techniques permettent une itération et une expérimentation rapides (par exemple, changer rapidement la posture d'une main ou la vue d'une caméra de la scène), ce qui n'est pas possible lors de la création manuelle de figures à base de tracés.

3 *Esquisse* : algorithme de rendu vectoriel

À partir de la scène 3D, l'objectif principal d'*Esquisse* est de générer des images vectorielles en se concentrant sur les contours des objets. Dans un fichier vectoriel, les objets dessinés sont décrits en tant que formes géométriques avec des propriétés telles que l'épaisseur du trait, sa couleur, son remplissage, son masque de visibilité, sa transparence, etc.

Une contrainte importante de notre approche de rendu est de créer uniquement des formes géométriques fermées, de sorte que les utilisateurs puissent modifier s'ils le souhaitent les objets dessinés dans un logiciel d'édition de graphiques vectoriels par la suite (*p.ex.* déplacer ou modifier la couleur ou l'opacité d'un objet). En résumé, nous voulons produire une image vectorielle aussi proche que possible de ce qu'un utilisateur créerait à la main afin de faciliter l'édition de l'image produite à l'aide d'autres logiciels.

3.1 État de l'art sur les techniques de rendu non photoréaliste vectoriel

L'approche naïve pour rendre des objets 3D en vectoriel consiste à projeter tous les polygones sur le plan image de la caméra en fonction de leur profondeur. Cela permet d'obtenir une image vectorielle similaire à celle de la version pixelisée, mais avec des artefacts liés à la difficulté de trier les polygones en fonction de leur profondeur, la *rasterization* et la gestion du *depth buffer* étant gérées différemment par OpenGL et le rendu SVG. Dans ce cas, la suppression de détails peut prendre du temps et la taille du fichier peut être importante en fonction du nombre de polygones. De plus, chaque triangle des maillages des objets étant dessiné de manière indépendante, il peut donc être fastidieux pour l'utilisateur de modifier la figure a posteriori. Une alternative consisterait à fusionner des triangles visibles adjacents, de sorte que chaque objet 3D ne donne qu'une seule forme vectorielle. Cependant, cette solution, lourde en calculs, ne résout pas le problème de découpage de triangles à moitié masqués par d'autres et nécessite des algorithmes de découpage de triangles qui peuvent mener à d'autres imprécisions dans le rendu.

Pour obtenir des contours fermés, Karsh et al. [75] ont développé l'algorithme *Snaxel*, une technique qui consiste à propager des points virtuels (*c.-à-d.* les *snaxels*) selon des règles prédéfinies le long des arêtes d'un maillage d'objet. Les *snaxels* sont reliés les uns aux autres pour former un *serpent* qui se déplace autour d'un objet. Une fonction d'énergie est associée à chaque *Snaxel* d'un *serpent* et correspond au produit scalaire entre la normale locale sur le maillage à la position du *Snaxel* et le vecteur de vue (vecteur défini entre une position donnée et la celle de la caméra). Pour extraire les contours de silhouette d'un objet, chaque serpent évolue jusqu'à atteindre une énergie de zéro (*i.e.* produit scalaire nul) et *avance* ou *recule* en fonction de leur visibilité. Les contours visibles de la silhouette sont alors déterminés sur la base des serpents ayant une énergie nulle et étant visibles (déterminés à l'aide de requêtes d'occultation OpenGL). La projection des contours 3D

sur le plan image fournit alors les contours 2D fermés de chaque objet à dessiner dans le SVG. Cependant, les *snaxels* évoluent le long des arêtes où la normale locale est interpolée à partir des sommets voisins, ce qui peut introduire des inexactitudes lors de l'extraction des contours. Par conséquent, la qualité des contours résultants dépend de la densité du maillage, en particulier lors du calcul de la visibilité d'un contour d'un objet cachant ou caché par autre, et donc, requiert de longs temps de calcul pour avoir des résultats satisfaisants.

Une autre approche consiste à analyser les arêtes du maillage des objets pour calculer une *carte de visualisation* [50]. La *carte de visualisation* est une structure spécifique qui attribue des propriétés à chaque arête d'un maillage telles que sa visibilité [8] (*caché* ou *visible*), sa nature (*silhouette* [59], *contour suggéré* [38]), sa longueur, sa profondeur, etc. Les auteurs ont par ailleurs développé leur propre langage de script permettant d'établir un processus de rendu personnalisé en quelques étapes (*selection*, *chaining*, *splitting* et *style*), basé sur la sélection d'arêtes ayant certaines propriétés choisies. Cette méthode permet d'obtenir les contours visibles des objets, mais une scène complexe où des objets se cachent les uns les autres produit des contours découpés en plusieurs traits non ordonnés. Il est alors difficile d'établir des règles pour ordonner les traits afin de former des polygones fermés, ce qui produit des artefacts de rendu lors du remplissage des polygones dans le fichier SVG et rend difficile pour l'utilisateur la modification du fichier produit par la suite.

Eisemann *et al.* ont résolu ce problème en utilisant l'API *2D arrangement* fourni par librairie CGAL⁷ [41]. L'API permet, pour un ensemble de traits quelconques dans le plan 2D, d'en extraire les régions 2D formées par ces traits. De cette façon, Eisemann *et al.* ont pu extraire les contours visibles de chaque objet, ajouter du style (couleur, gestion de la lumière, ombres) aux régions visibles de chaque objet avant de les écrire dans un format vectoriel.

Nous avons donc choisi d'adapter l'approche proposée par Eisemann *et al.* [41] pour le rendu d'*Esquisse* dans Blender. Cette approche semble être la seule à permettre de rendre des figures composées de formes géométriques simples pouvant facilement être éditées dans un logiciel de dessin vectoriel.

3.2 Algorithme de rendu non-photoréaliste vectoriel

Intéressons-nous maintenant en détail aux étapes de rendu vectoriel effectué par *Esquisse*. *Esquisse* calcule d'abord tous les contours issus de la scène 3D avant de les écrire dans un fichier SVG. Cette section explique comment ces contours sont calculés et comment le rendu dans le fichier SVG final est effectué.

7. <https://www.cgal.org>

3.2.1 Génération des contours

Nous avons adapté les deux approches présentées dans [41, 50] à nos besoins et les avons mises en œuvre dans Blender comme suit.

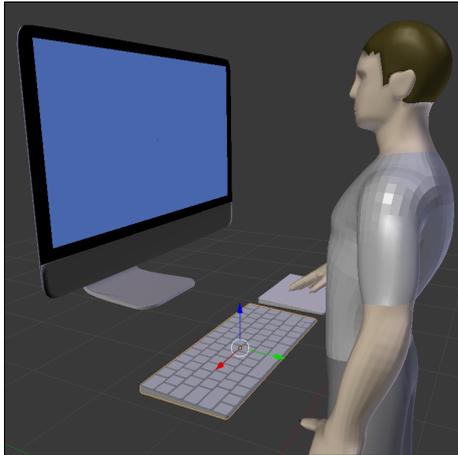


Figure 3.13 – Scène initiale.

Scène initiale

Nous prendrons comme exemple la scène suivante (cf. figure 3.13), constituée d'un avatar interagissant avec un ordinateur de bureau via un clavier et un pavé tactile.

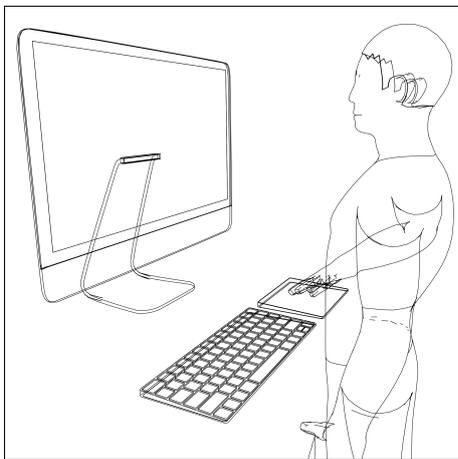


Figure 3.14 – Extraction des contours.

Étape 1 : Extraction des contours

Comme pour les deux approches [41, 50] qui construisent une *carte de visualisation*, la première étape du rendu consiste à itérer sur toutes les arêtes des différents objets de la scène et à en sélectionner des sous-ensembles correspondant à un type spécifique. Par exemple, les arêtes qui nous intéressent sont de type *silhouette* (arête entre une face visible et une face cachée), de type *pli* (l'angle formé entre des deux faces est supérieur à un seuil donné) ou encore de type *bordure de couleur* (la couleur des faces connexes est différente).

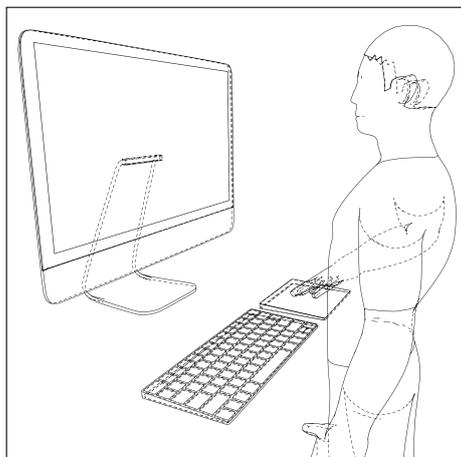


Figure 3.15 – Contours visibles en ligne continue, cachés en pointillés.

Étape 2 : Visibilité des contours

Les contours extraits des différents objets de la scène peuvent être cachés par d'autres objets. La deuxième étape consiste à calculer leur visibilité. La visibilité est calculée en utilisant des requêtes d'occlusion OpenGL (extension ARB⁸) et le décalage de polygones (*polygon offset*) comme suggéré dans [41] pour éviter les erreurs de calcul dans le *depth buffer*. Ces requêtes renvoient le nombre de pixels dessinés pour un objet donné, permettant de savoir s'il est visible ou non. La figure 3.15 montre les contours visibles en ligne continue et les contours cachés avec des pointillés. Il est à noter que la conservation des bords cachés est cruciale, car elle

peut être nécessaire pour illustrer certains scénarios interactifs (généralement des cas d'interaction à l'arrière d'un appareil) ou pour une question de style, où il peut être intéressant de dessiner des contours visibles et cachés avec des styles différents.

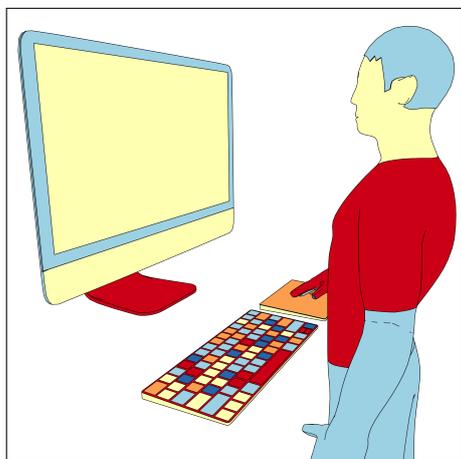


Figure 3.16 – Régions formées par les contours.

Step 3 : Régions de remplissage

Une fois que toutes les propriétés des arêtes des différents objets sont déterminées, nous calculons les régions formées par la projection en 2D sur le plan image des arêtes visibles. Comme [41], nous utilisons l'algorithme *2D arrangement* fourni par la librairie CGAL, qui calcule les régions formées par un ensemble de traits en 2D. La figure 3.16 montre l'ensemble des régions visibles formées par la projection des contours visibles des différents objets de la scène 3D avec une couleur différente.

8. ARB Extension – <https://www.khronos.org/registry/OpenGL/extensions/ARB>

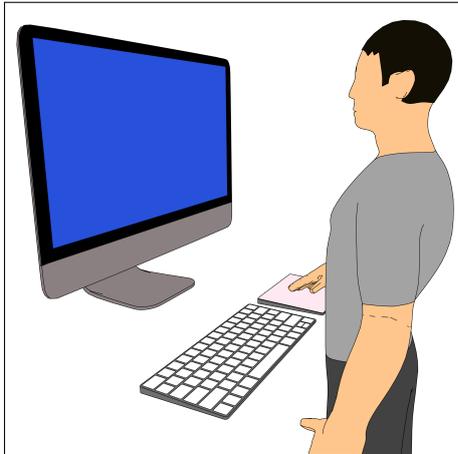


Figure 3.17 – Association des régions aux objets.

Étape 4 : Associer les régions aux objets 3D

Chaque région formée est ensuite associée à son objet 3D correspondant dans la scène. Pour cela, nous choisissons un point aléatoire situé à l'intérieur de la région (en utilisant la bibliothèque GPC⁹), puis nous lançons un rayon depuis ce point dans le plan image vers la scène 3D. En fonction de la face touchée par le rayon, nous savons alors quel est l'objet concerné par cette région, mais aussi sa couleur. À l'issue de cette étape, nous avons alors pour chaque objet, un ensemble de régions visibles et leur couleur associée (cf. figure 3.17).

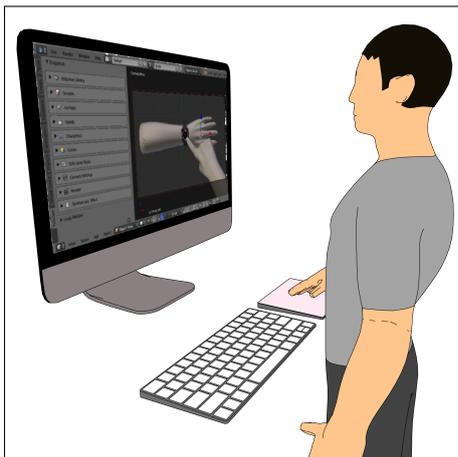


Figure 3.18 – Projection d'interfaces sur les objets 3D.

Étape 5 : Projection d'interfaces

Esquisse offre la possibilité d'inclure des interfaces utilisateur sur les objets écran. La transformation perspective de l'interface de l'espace 2D vers l'espace 3D est calculée via un solveur d'algèbre linéaire (inclu dans Python *numpy*) pour déterminer l'homographie plane entre les quatre coins de l'interface utilisateur et les quatre coins de l'objet écran dans la scène. Cependant, étant donné que le format SVG ne prend pas nativement en charge les transformations de matrices en perspective, nous appliquons l'homographie à tous les objets géométriques de l'interface utilisateur en calculant les nouvelles coordonnées.

Certains éléments comme les ellipses, les arcs ou le texte doivent alors être d'abord discrétisés en segments ou en courbes de Bézier pour pouvoir être projetés. Si le fichier source est une image (type PNG ou JPEG), alors la bibliothèque OpenCV permet facilement d'en calculer la transformation perspective.

9. <http://www.cs.man.ac.uk/~toby/gpc/>

3.2.2 Génération du fichier SVG

Une fois toutes les étapes de rendu terminées nous écrivons toutes les informations calculées dans un fichier SVG.

Rendu par couche Le format SVG est basé sur un système de couches pour définir l'ordre de priorité de rendu des objets dessinés. Le fichier SVG produit par *Esquisse* comprend trois couches principales, une pour les remplissages, une pour les contours visibles, et une dernière pour les contours cachés (qui doivent être dessinés en dernier pour être en haut de la pile de dessin et donc visibles). Chaque couche contient une sous-couche par objet pour faciliter la modification du fichier SVG a posteriori à l'aide d'autres logiciels.

Pour chaque trait et pour chacun des objets, le style peut être personnalisé directement dans le panneau latéral d'*Esquisse* tel que sa couleur, son épaisseur et son apparence (ligne pointillée ou continue).

Générer des contours remplis contenant des trous. Certaines régions extraites de l'étape 4 du rendu peuvent ne pas être pleines et avoir un ou plusieurs « trous » à l'intérieur. La spécification SVG dit que pour un polygone quelconque à trous, la règle de remplissage *non-zero*¹⁰ permet de remplir le polygone, sauf ses trous, si les points des contours intérieurs du polygone sont orientés dans le sens inverse des aiguilles d'une montre et les points du contour extérieur orientés dans le sens des aiguilles d'une montre. Sachant que l'algorithme *2D arrangement* de la bibliothèque CGAL renvoie les régions sous forme de hiérarchie, les régions situées à l'intérieur d'autres régions et n'ayant pas touché d'objet dans l'étape de lancé de rayons, peuvent être considérées comme des trous et donc être dessinées dans le sens inverse des aiguilles d'une montre.

Projection des IU sur les écrans. Les interfaces projetées en perspective sont incluses dans la couche des remplissages. Un masque de visibilité est appliqué à chaque interface générée en utilisant les contours visibles de l'objet écran afin d'en afficher que les portions visibles dans la scène et éviter les artefacts de chevauchement (*p.ex.* un doigt devant l'écran du smartphone).

3.2.3 Rendu d'une figure avec effet stroboscopique

Pour rendre une figure avec un effet stroboscopique, quelques ajustements doivent être effectués dans le pipeline du rendu. Tout d'abord, *Esquisse* effectue toutes les étapes de rendu pour chacune des images clés d'un effet stroboscopique, en utilisant dans la scène 3D uniquement la version de l'objet correspondante à l'image clé. Faire des rendus multiples implique d'avoir des contours multiples pour certains des objets (images clés et état final),

10. <https://www.w3.org/TR/SVG/painting.html>

et donc, nécessite de choisir avec soin l'ordre de dessin des objets dans le SVG. Tout d'abord, tous les objets qui n'ont pas bougé dans la scène sont dessinés. Ensuite, nous itérons sur chaque image clé dans l'ordre chronologique et dessinons les objets avec une transparence de plus en plus faible. Notez que ces modifications augmentent légèrement le temps de rendu, qui reste inférieur à 5 secondes par exemple pour les figures 3.8 et 3.11, lorsqu'il est effectué avec l'ordinateur équipé d'un processeur i7 4GHz décrit dans la section sur l'étude (cf. 4).

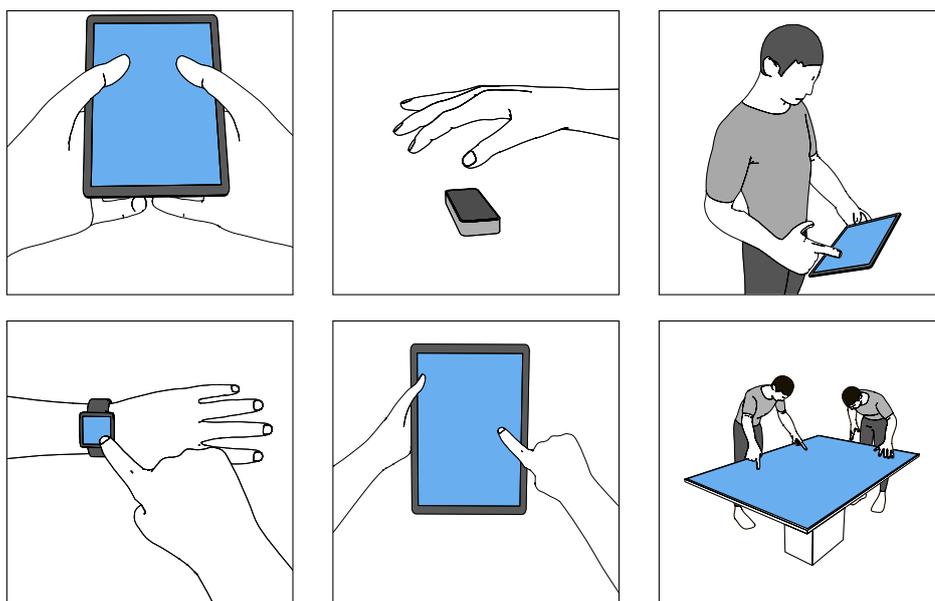


Figure 3.19 – Rendu des modèles de scène proposés par *Esquisse*

4 Étude : illustration de techniques d'interaction

Nous avons mené une expérience contrôlée demandant à des étudiants et des chercheurs en IHM de produire l'illustration visuelle de techniques d'interaction de l'état de l'art en utilisant *Esquisse* mais aussi les outils de leur choix.

4.1 Méthode d'expérimentation

4.1.1 Participants et dispositif

Nous avons recruté 8 participants ($\bar{x}=33$ ans, $\sigma=10$), tous chercheurs en IHM. Trois étaient des chercheurs académiques, trois doctorants et deux étudiants de deuxième cycle. Aucun de ces participants n'avait une expérience préalable avec le logiciel Blender même si six d'entre eux l'ont déjà testé une fois par curiosité. De tous les participants, cinq avaient déjà produit des figures à base de tracés auparavant (quatre en utilisant Inkscape¹¹, un

11. Inkscape – <http://Inkscape.org>

avec Adobe Illustrator¹²) et étaient également familiers avec la technique de décalquage de photographies (deux d'entre eux, des chercheurs senior académiques se considéraient même experts). L'expérience a été menée avec Blender v2.79 et l'extension *Esquisse* installée sur un iMac 27" 5K avec processeur Intel Core i7 à 4Ghz. Trois périphériques d'entrée étaient disponibles : une souris d'ordinateur Logitech G9 Laser, un Apple Magic Trackpad 2 et la caméra de capture LeapMotion. Safari, Chrome, Adobe Illustrator, Inkscape, Gimp, Apple Keynote et Microsoft Powerpoint ont été installés et pouvaient être utilisés à tout moment si les participants souhaitaient modifier le fichier vectoriel produit avec *Esquisse*.

4.1.2 Procédure et design de l'expérience

Les participants ont été invités à s'asseoir devant l'ordinateur et ont été informés que l'expérience consistait à produire des figures à base de tracés. Nous leur avons d'abord montré un ensemble de figures à base de tracés extraites de la taxonomie afin d'expliquer clairement ce que sont les figures à base de tracés. Les participants ont ensuite visionné un tutoriel vidéo de 10 minutes sur les commandes de base dans Blender (positionnement des objets 3D et de la caméra), et de l'interface d'*Esquisse* et de ses fonctionnalités¹³. Après cette introduction, nous avons présenté la tâche principale qui consistait à illustrer deux techniques d'interaction parmi un ensemble de quatre publiées dans la communauté IHM (*AuraSense* [162], *Put-that-there* [19], *TiltReduction* [30] et *Stitching* [61]). Nous avons choisi ces techniques d'interaction car elles reposent sur une variété d'appareils (*e.g.* smartphone, montre connectée, grand écran) et de modalités d'entrée (mouvement de l'appareil, tactile, interaction sur le corps et entrée vocale). Les techniques d'interaction ont été montrées via les vidéos originales plutôt que par des images afin d'éviter que les participants n'essaient de *reproduire* des illustrations au lieu de *créer* des illustrations originales.

Les participants devaient produire deux figures. Une devait être produite avec *Esquisse*, l'autre avec les outils de leur choix dans un délai de 30 minutes pour chacune des conditions. Les participants ont également été informés qu'ils pouvaient éditer le fichier SVG produit par *Esquisse* avec un logiciel d'édition graphique s'ils le souhaitaient. L'ordre des figures a été contrebalancé entre les participants en utilisant un carré latin. L'expérience a suivi un protocole de réflexion à voix haute afin de noter tout commentaire. L'objectif principal de l'expérience n'étant pas la production d'interfaces 2D, nous avons fourni un ensemble d'interfaces 2D associées aux 4 techniques d'interaction, à utiliser comme interfaces d'écran.

12. Adobe Illustrator – <https://www.adobe.com/fr/products/illustrator.html>

13. Tutoriel *Esquisse* – <https://ns.inria.fr/loki/EsquisseBlender>

4.2 Résultats

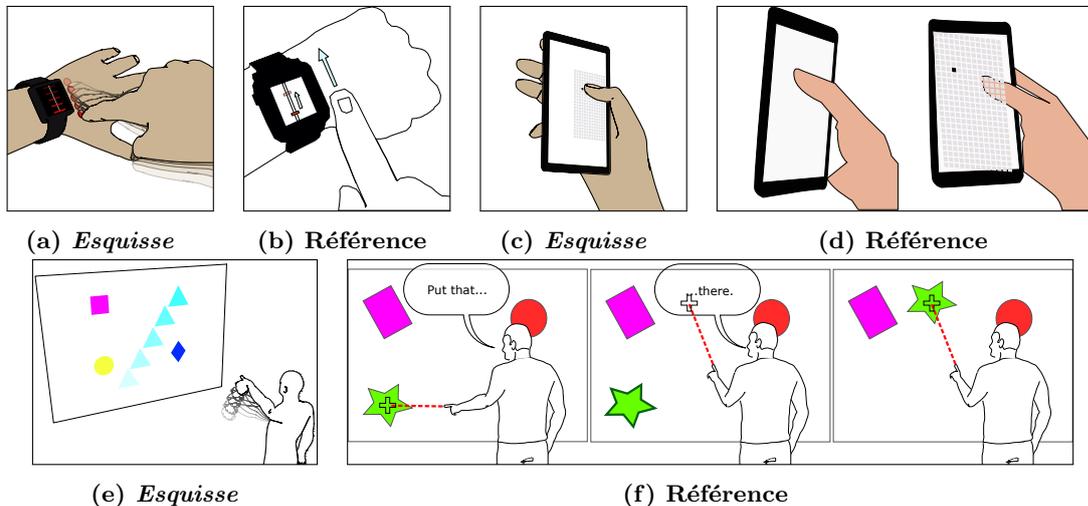


Figure 3.20 – Exemples de figures produites avec *Esquisse* et avec les outils de préférence des participants pour les techniques d'interaction *AuraSense* [162] (a) et (b), *TiltReduction* [30] (c) et (d), et *Put-that-there* [19] (e) et (f).

Le temps de production de la figure avec *Esquisse* a pris en moyenne 23min25s ($\sigma=6\text{min}5\text{s}$) et 28min42s pour la technique de référence du participant ($\sigma=1\text{min}56\text{s}$). En détail, les temps pour chaque techniques sont : *AuraSense* : *Esquisse* 17min17s, référence 26min14s ; *Put That There* : *Esquisse* 19min21s, référence 30min ; *Stitching* : *Esquisse* 27min1s, référence 28min36s ; *Tilting* : *Esquisse* et référence toutes les deux 30min0s.

La variabilité des temps entre les différentes technique d'interaction à illustrer s'explique facilement par les similitudes entre l'interaction à illustrer et les modèles de scène fournis par *Esquisse*. En général, *Put That there* et *AuraSense* étaient plus rapides à reproduire, car *Esquisse* fournit déjà un modèle d'un index interagissant sur une montre connectée et un autre modèle avec un personnage interagissant avec un tableau interactif. Les délais de production pour les figures concernant l'interaction *Tilting* sont étonnamment longs alors que nous disposons d'un modèle de main tenant un smartphone. Les deux participants ont essayé d'utiliser des rotations plutôt que d'utiliser le système d'ancrage et ont donc perdu beaucoup de temps. Nous nous intéressons en détail à l'utilisation d'*Esquisse* par les participants dans la section suivante.

4.2.1 Interaction et stratégies d'utilisation d'*Esquisse*

Utilisation des modèles de scène. Tous les participants sauf deux ont commencé leur illustration en utilisant l'un des modèles prédéfinis. Il est intéressant de noter que les deux participants qui ne l'ont pas fait (P3, P8) réalisaient des illustrations pour l'interaction *Stitching* qui consiste en un geste du stylo s'étendant sur deux tablettes interactives, une

situation pour laquelle *Esquisse* ne fournit pas de modèle de scène. P3 et P8 ont donc ajouté manuellement deux appareils mobiles et une main droite avec un stylet et les ont placés dans la scène, ce qui leur a pris à tous les deux moins de 5 minutes pour disposer d'une scène complète, sans compter les effets ajoutés. Dans les autres cas, *Esquisse* fournit des modèles prédéfinis suffisamment proches des interactions pour les illustrer.

Utilisation du système d'ancrage. Comme pour les modèles de scène, tous les participants ont utilisé des ancres sauf les deux participants qui ont dû illustrer *Stitching* (P3, P8). Dans ce cas particulier, les deux participants ont adopté un processus similaire : ils ont simplement déplacé la main droite de leur scène avec le stylet sur les deux dispositifs sur un seul axe sans utiliser le système d'ancrage. Il est intéressant de noter que l'un des participants souhaitait utiliser une ancre sur le stylet (P8), fonctionnalité qui n'est pas disponible dans *Esquisse* mais qui pourrait être rapidement mise en œuvre. Son idée était de placer une ancre sur l'écran de la tablette, puis de relier l'ancre à la pointe du stylet, et donc de déplacer le stylet et la main en même temps lors du déplacement de l'ancre. Dans l'ensemble, tous les participants ont compris et adopté très rapidement l'utilisation des ancres.

Effet stroboscopique. Il est à noter que la fonctionnalité de représentation du mouvement par des flèches n'était pas encore implémentée et donc indisponible lors de l'expérience. Tous les participants sauf un ont utilisé l'effet stroboscopique d'*Esquisse* afin d'illustrer le mouvement. Deux d'entre eux (P1, P2) qui devaient illustrer l'interaction *AuraSense* ont utilisé la fonction d'interpolation afin de générer des images supplémentaires entre les deux images clés définies (cf. figure 3.20a). Une fois de plus, ces participants ont effectué un processus similaire : ils ont positionné la main droite avec une ancre, ajouté une image clé, déplacé la main droite vers une position différente et enfin, défini les interpolations. Le participant qui n'a pas utilisé l'effet stroboscopique (P5) s'est considéré comme un expert en figures, et l'a fait intentionnellement pour construire à la place une figure de type *storyboard* pour l'interaction *TiltReduction* en utilisant des illustrations distinctes (cf. figure 3.20d).

Manipulations 3D. Tous les participants sauf deux ont utilisé le système d'ancrage et le manipulateur de translation à 3 axes pour effectuer tous les déplacements qu'ils souhaitaient, sans exprimer de difficultés. Seuls deux participants (P5, P6) ont eu besoin de tourner des objets pendant leur mise en scène pour illustrer l'interaction *TiltReduction*, qui implique une rotation du poignet pour faire basculer l'interface d'un smartphone. L'un d'entre eux (P6) était le seul participant qui avait des difficultés avec *Esquisse* et n'a pas réussi à créer ce qu'il voulait, commentant que « *les manipulations 3D sont trop difficiles pour [lui]* ». Cela dit, ce participant a ignoré des stratégies alternatives plus faciles qui

auraient pu être utilisées pour faire tourner les objets, comme l'ancrage de tous les doigts pour forcer la pose de la main, déplacer la caméra vers un point de vue différent, ou encore utiliser la caméra LeapMotion. Cette dernière a été utilisée par P5 pour changer plus facilement l'orientation de la main. Dans l'ensemble, seuls deux participants ont utilisé le LeapMotion pour changer la posture de la main, l'autre participant (P2) illustre l'interaction *AuraSense* et l'a utilisé pour définir une posture de pointage de l'index pour la main droite. Enfin, tous les participants ont utilisé la fonction de téléportation de la caméra à la position courante de la fenêtre de visualisation pour choisir le point de vue de leur illustration.

Post-édition du fichier SVG. Bien que tous les participants aient été invités à le faire, seuls trois participants (P4, P5, P7) ont post-édité la figure produite avec *Esquisse* dans un logiciel d'édition vectorielle. Alors que deux d'entre eux n'ont effectué que des ajustements mineurs (type changement de couleur ou épaisseur des traits), P5 a lui construit un *storyboard* constitué de plusieurs rendus distincts effectués par *Esquisse* et y a superposé une flèche représentant le mouvement sur une des figures.

4.2.2 Retour subjectif des participants

Apprentissage et utilisation. Dans l'ensemble, les participants ont apprécié utiliser *Esquisse*, l'outil était « facile à utiliser et amusant » (P4, P5, P8) et « rapide » (P5, P7, P8), même si *Esquisse* nécessite une phase d'apprentissage pour la manipulation des objets et l'interface de Blender (P1, P3, P8). Les participants ont rapidement compris toutes les fonctionnalités proposées en regardant simplement la vidéo tutorielle (P1, P3, P4, P5). Mieux encore, P1 a été « impressionné » et a déclaré qu'il était « rapide à apprendre et à maîtriser les différentes fonctionnalités », ce qui fait d'*Esquisse* un outil intéressant compte tenu de ses « faibles compétences artistiques ».

Post-édition. Globalement, les participants ont été satisfaits des résultats du rendu produit par *Esquisse*, mais certains préfèrent encore affiner la figure par la suite dans un logiciel d'édition graphique (P4, P5, P7). C'est un retour d'information surprenant, car P4 et P7 ont eu encore au moins 10 minutes à leur disposition pour leur production de figure avec *Esquisse* et n'ont pas du tout modifié leur figure dans un logiciel externe. Plus important encore, les participants ont vu dans *Esquisse* un outil de « prototypage rapide pour réaliser des illustrations à l'aide d'objets complexes à modifier par la suite » (P5, P7).

Suggestions d'améliorations. Les participants ont été enthousiasmés par *Esquisse* et ont commenté plusieurs caractéristiques supplémentaires qui pourraient enrichir ses fonctionnalités. P2 a fait remarquer que la capacité de contrôler n'importe quel objet de

la scène en utilisant la détection de mouvement (comme Yoon *et al.* l'ont fait avec un smartphone [158]) pourrait être utile pour rendre la mise en scène 3D encore plus facile. P4 aurait souhaité identifier les objets des images clés de la scène, par exemple en utilisant une couleur différente ou un numéro associé à l'index des images clés. P8 a fait remarquer que la possibilité de modifier un fichier d'interface utilisateur directement dans *Esquisse* optimiserait le processus créatif. Enfin, deux participants (P5, P8) ont fait remarquer que la possibilité de superposer des flèches « *entre ici et là* » directement dans *Esquisse*, en utilisant les ancres ou en affichant éventuellement des flèches entre les images clés, supprimerait la nécessité d'ouvrir ensuite le fichier SVG produit dans un logiciel d'édition graphique. L'ajout de flèches a depuis été implémenté dans *Esquisse*, malheureusement, après cette expérimentation. Les autres propositions d'améliorations, quant à elles, peuvent facilement être mises en œuvre dans une future version d'*Esquisse*.

Discussion

Limitations d'*Esquisse*

Un aspect d'*Esquisse* qui peut être considéré à la fois comme un avantage ou un inconvénient est qu'il a été construit comme une extension Blender. Nous avons pris cette décision parce que Blender fournit un environnement solide pour la modélisation 3D et la manipulation d'objets dont *Esquisse* bénéficie, cela simplifie ainsi sa distribution et sa maintenance. Pour cette raison, cependant, l'interaction avec *Esquisse* est limitée par ce que l'environnement Blender permet aux extensions de faire (par exemple, la conception de l'interface utilisateur des modules complémentaires est limitée). À cet égard, nous avons conçu *Esquisse* en essayant d'offrir la meilleure expérience utilisateur et la meilleure interaction que Blender nous a permis de concevoir. Par exemple, nous avons ajouté des commandes de curseur dédiées pour faciliter la définition des poses des mains et du corps et avons mis en œuvre une intégration directe de LeapMotion pour aider les utilisateurs à manipuler les poses des mains. De même, inspirés par DemoDraw [35], l'intégration de la caméra Microsoft Kinect pourrait être intégrée dans *Esquisse* pour faciliter la manipulation des poses du corps entier. Par ailleurs, une mise à jour majeure de Blender (version 2.8) a eu lieu entre la création d'*Esquisse* et la rédaction de ce manuscrit, et nécessite malheureusement une quasi-réécriture complète d'*Esquisse*.

Modifier les pratiques actuelles

Une différence significative entre le processus créatif proposé par *Esquisse* et celui des utilisateurs actuels est qu'*Esquisse* s'appuie sur des modèles 3D existants plutôt que sur des photographies. Bien que l'ajout de nouveaux modèles à *Esquisse* soit aussi simple que l'ajout d'un fichier dans un dossier, la conception de ces modèles spécifiques nécessite

encore des compétences de base en modélisation 3D et en mise en scène. Nous prévoyons qu'à l'avenir, les progrès de la vision par ordinateur et de la recherche en infographie permettront d'extraire des objets d'une photographie pour en créer automatiquement des modèles 3D correspondants [161, 32, 81, 155, 69, 92].

Conclusion

Les figures à base de tracés sont des supports efficaces et fréquemment utilisés pour illustrer les documents de recherche en IHM. En se basant sur les stratégies identifiées dans une version préliminaire de notre taxonomie¹⁴, nous avons conçu *Esquisse*, un outil qui met en œuvre un nouveau processus créatif basé sur la manipulation d'objets 3D pour mettre en place une scène en 3D et l'exporter sous forme de figure vectorielle. Nous avons développé des techniques d'interaction pour faciliter cette mise en scène (notamment l'utilisation de points d'ancrage pour faciliter la manipulation d'objets articulés complexes tels que les mains et l'intégration d'effets de rendu de mouvement) et un pipeline de rendu spécifique pour extraire les différents contours, les remplir et enfin générer un fichier SVG vectoriel.

Esquisse fournit ainsi un processus créatif alternatif pour produire des figures à basse de tracés, qui peut être utilisé par les utilisateurs qui pensent ne pas avoir les compétences requises pour effectuer un décalquage manuel de photographies. *Esquisse* peut également être utile aux utilisateurs qui, eux, ont l'habitude de recourir au décalquage, par exemple, comme outil de prototypage rapide, comme l'ont indiqué les participants de notre expérience. Une scène 3D mise en place avec *Esquisse* peut également être exportée « telle quelle » comme modèle pour décalquage manuel, ce qui pourrait être utile dans les situations où il faut illustrer un scénario interactif « complexe et lourd » (impliquant par exemple des tables, de grands écrans, plusieurs utilisateurs, etc.) sans avoir à préparer l'installation physique nécessaire pour prendre une photo de la scène.

Nous avons implémenté *Esquisse* comme une extension pour le logiciel de modélisation 3D open source Blender et évalué sa facilité d'utilisation dans une expérience qualitative avec 8 participants. Les résultats de cette expérience suggèrent que les utilisateurs, même s'ils ne connaissent pas Blender et n'ont pas l'habitude de produire des figures à base de tracés, ont réussi à produire rapidement des figures à base de tracés illustrant des techniques d'interaction tirées de la littérature avec peu ou pas d'expérience dans la production de figures à base de tracés ou l'utilisation d'un logiciel de modélisation 3D.

14. Le projet *Esquisse* a été effectué avant le projet de taxonomie durant la thèse. Une version préliminaire d'une taxonomie avait été établie dans la publication d'*Esquisse* et se concentrait essentiellement sur les figures à base de tracés. Cette version préliminaire a servi de base à notre taxonomie plus complète, présentée dans le chapitre 2.

4

Étude des techniques de rotation d'objets 3D

Sommaire

Introduction	103
1 État de l'art sur les techniques de rotation 3D via des sphères virtuelles	105
1.1 Interaction sur la sphère virtuelle	105
1.2 Interaction en dehors de la sphère virtuelle	106
1.3 Comparaisons des sphères virtuelles	107
2 Étude : influence de la taille de la sphère de contrôle	108
2.1 Méthode d'expérimentation et dispositif	108
2.2 Procédure d'expérimentation	109
2.3 Résultats	111
Discussion	116
Discussion des résultats	116
Stratégies de rotation	117
Conclusion	119

Introduction

La manipulation d'objets est une tâche fondamentale dans les environnements 3D qui peut-être décomposée en deux sous-tâches : le positionnement et la rotation d'objets 3D. Les rotations 3D sont également utilisées pour le changement du point de vue de caméras dans une scène. La tâche de rotation 3D a été perçue comme difficile par plusieurs des participants à l'évaluation du système *Esquisse* présenté au chapitre précédent. En effet, bien que nous ayons mis à disposition des techniques d'interaction permettant d'éviter l'emploi de rotations d'objets, deux participants ont tout de même eu recours à la rotation du modèle 3D de smartphone pour illustrer la figure de l'interaction *TiltReduction* [30]. Pour effectuer de telles rotations et modifier l'orientation des objets, *Esquisse* se base sur les contrôles natifs de Blender. Vu la difficulté rencontrée par ces participants, nous nous sommes posé la question de savoir comment cette tâche pourrait être facilitée. En particulier, nous nous intéressons dans ce chapitre aux techniques de rotation 3D dans un contexte d'environnement de bureau.

Plusieurs techniques de base permettent de modifier l'orientation d'un objet. La manière la plus simple pour définir l'orientation d'un objet 3D est probablement via la description de ses trois angles d'Euler (la précession, la nutation et la rotation propre). Ces valeurs peuvent généralement être définies directement dans les logiciels de modélisation 3D via une fenêtre de propriétés (par exemple, Blender permet de définir la rotation d'un objet selon les axes X, Y et Z en degrés ou radians). Cette technique peut s'avérer efficace quand la valeur à définir est connue à l'avance, ou quand la rotation n'est nécessaire que suivant un seul axe, mais elle montre ses limites dès qu'il s'agit de définir des degrés de rotation plus complexes, c'est-à-dire selon deux ou trois axes simultanément.

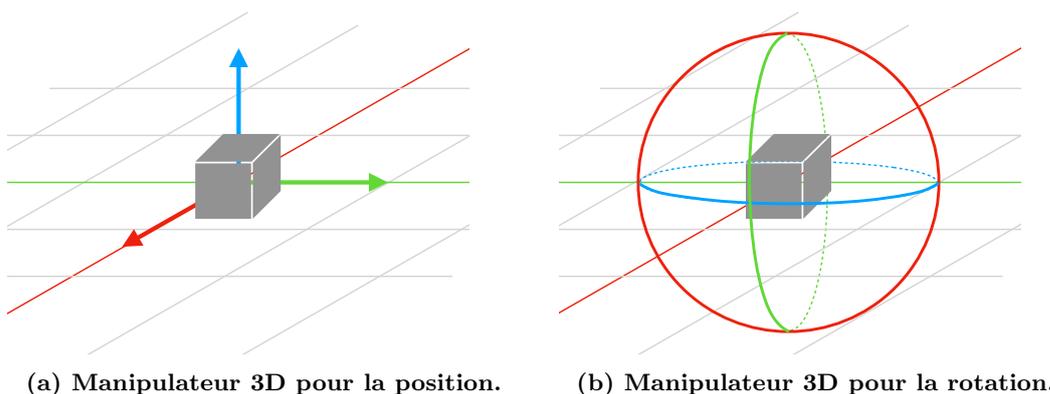


Figure 4.1 – Illustration des widgets spécifiques pour manipuler la position et la rotation d'un objet dans Blender. (a) Le manipulateur 3D à 3-axes permet de modifier la position de l'objet selon un seul axe en cliquant sur la flèche correspondante. (b) Le manipulateur 3D à 3-cercles permet de modifier la rotation de l'objet selon un seul axe en cliquant sur le cercle correspondant.

Une autre technique est d'utiliser un widget spécifique, un manipulateur 3-axes, visible

dans la scène 3D. Ce widget, développé spécifiquement pour les interactions de bureau (à base de pointeur) est globalement disponible dans tous les logiciels de modélisation 3D (*p.ex.* Blender, Unity, Maya, etc.) et se déplace à la position de l'objet sélectionné. Cet objet existe en deux versions. La première version affiche trois flèches (généralement en rouge, vert, et bleu pour les axes X, Y et Z) avec lesquels l'utilisateur peut interagir via un clic et un déplacement du pointeur souris pour contrôler la position de l'objet [17] (*cf.* figure 4.1a). Une deuxième version, basée sur la technique *Overlapping slider* de Chen *et al.* [31] et appelée *Discrete Sliders* par Rybicki *et al.* [130], consiste à afficher trois cercles formant une sphère centrée sur l'objet représentant les 3 axes de rotation possibles (*cf.* figure 4.1b). Cliquer et déplacer le curseur de souris sur l'un de ces trois cercles permet de contrôler directement la rotation de l'objet sélectionné selon l'axe de rotation choisi dans la scène. Bien qu'apportant un contrôle direct et précis de la rotation de l'objet dans la scène, cette technique se limite toujours au contrôle d'un seul axe de rotation à la fois.

Afin de pouvoir contrôler directement et simultanément plusieurs axes de rotations, des techniques spécifiques basées sur une sphère virtuelle (en anglais *trackball*) ont été proposées dans la littérature [11, 31, 133]. Ces techniques associent les deux degrés de liberté du curseur souris aux trois degrés de liberté de rotation, permettant ainsi de contrôler les trois axes de rotation de l'objet d'un seul mouvement. Le principe est de projeter la position du curseur de souris sur une sphère englobante de l'objet manipulé. Le déplacement du curseur de souris implique un delta entre les positions projetées sur la sphère, delta qui forme alors l'angle de la rotation à effectuer. À ce titre, les techniques de *trackball* paraissent particulièrement intéressantes pour être intégrées à l'application *Esquisse* dans le but de faciliter les rotations 3D.

Dans ce chapitre, nous nous intéressons au contrôle de la rotation d'un objet 3D et plus spécifiquement à la technique de *trackball* via les dispositifs de pointage classiques tels que la souris ou le pavé tactile. Un des objectifs est de comprendre dans un premier temps les facteurs impliqués lors de l'utilisation d'une *trackball* et l'impact qu'ils peuvent avoir sur le contrôle de rotation et le comportement de l'utilisateur dans une tâche de rotation 3D. En particulier, nous émettons l'hypothèse que la taille de sphère virtuelle utilisée par une *trackball* a un impact direct sur les performances de l'utilisateur. En effet la sphère détermine non seulement la zone d'interaction disponible pour orienter l'objet, mais aussi la vitesse et la précision de la rotation de l'objet 3D. Ainsi, une sphère de petite taille peut réduire les déplacements de pointeur nécessaires car les rotations sont amplifiées par rapport à des sphères de plus grand diamètre, mais l'utilisateur pourrait manquer de précision. À l'inverse, une sphère de grande taille peut augmenter le nombre de débrayages nécessaires pour réaliser une rotation de grande amplitude si l'utilisateur souhaite conserver le curseur dans une certaine zone de l'écran ou minimiser les déplacements physiques du dispositif de pointage. Les objectifs principaux de cette étude sont d'identifier et mesurer l'impact sur les performances et le comportement des utilisateurs de la taille de la

sphère, et d'ouvrir la réflexion sur comment les techniques de *trackball* pourraient être adaptées pour ne plus dépendre de leur taille.

Ce chapitre est structuré comme suit. Après avoir établi un bref état de l'art des connaissances actuelles sur les techniques de rotation 3D reposant sur des sphères virtuelles, nous présentons les résultats d'une étude qui, au travers d'une expérience contrôlée, s'intéresse à la mesure de l'influence de la taille de la sphère sur les performances et le comportement des utilisateurs dans une tâche de rotation 3D. Enfin, nous discuterons des suites possibles de cette étude.

1 État de l'art sur les techniques de rotation 3D via des sphères virtuelles

Trois approches différentes de techniques de rotation 3D via une sphère virtuelle sont majoritairement mentionnées dans la littérature, *Two-axis Valuator* [31] la *Trackball* [31] et l'*Arcball* [11, 133]. La différence entre ces techniques repose sur le fait qu'elles modifient le comportement de la rotation selon que l'utilisateur clique dans ou en dehors de la sphère virtuelle.

1.1 Interaction sur la sphère virtuelle

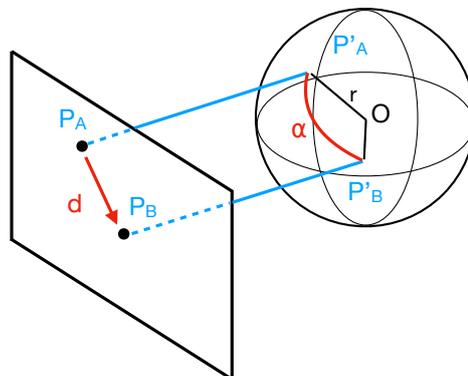


Figure 4.2 – (a) L'angle de rotation α est calculé entre les points P'_A et P'_B , projections orthographiques sur la sphère des points P_A et P_B .

La technique *Two-axis Valuator*, est une version initiale introduite par Chen, où un déplacement du curseur de souris sur la sphère virtuelle se traduit par une rotation de l'objet selon l'axe X et Y proportionnellement au déplacement du curseur selon les axes X et Y [31]. Les techniques *Trackball* [31] et *Arcball* [11, 133] elles se basent sur un même principe qui est d'associer aux coordonnées en deux dimensions sur le plan image du pointeur de la souris (point P), des coordonnées en trois dimensions dans l'espace 3D

(point P'), tel que P' soit la projection orthographique du point P sur la sphère virtuelle de rayon r et de centre O . La rotation de l'objet 3D induite par le déplacement de la souris d'un point P_A vers un point P_B dans le plan image est alors calculée comme la plus petite rotation possible de la sphère du point P'_A vers le point P'_B dans l'espace 3D (cf. figure 4.2). La *Trackball* proposée par Chen décrit la rotation à effectuer avec des angles d'Euler [31], et est donc soumise à une hystérésis, c'est-à-dire que deux mouvements différents de la souris vers un même point donnent deux rotations différentes. Les versions de l'*Arcball* proposées par Shoemake et Bell résolvent le problème d'hystérésis en décrivant les rotations à l'aide de quaternions [11, 133], définis par la rotation du vecteur unitaire $\overrightarrow{OP'_A}$ au vecteur $\overrightarrow{OP'_B}$. Nous nous référons à l'article proposé par Henriksen pour une comparaison approfondie de ces trois techniques de rotation [58]. Généralement, ces techniques utilisent une projection orthographique qui rend la moitié de la sphère virtuelle « visible » par projection, et permet ainsi une bijection parfaite entre les points 2D projetés et les points 3D sur la demi-sphère. Dans une projection perspective, seule une partie de la sphère virtuelle est disponible pour la projection, à cause de la focale de la caméra, diminuant ainsi la surface d'interaction. La zone d'interaction et la surface de la sphère visible ont un impact sur le comportement de la rotation. En effet, si on reprend le schéma illustré figure 4.2 et pour deux points espacés à l'écran d'une distance donnée, plus le curseur de souris est proche du bord de la sphère, plus la longueur de l'arc défini par les deux projections successives des points sur la sphère virtuelle sera importante. Si la longueur d'arc augmente, alors la variation de l'angle de rotation augmente aussi et peut donner un certain effet « d'accélération » lorsque le curseur est proche du bord de la sphère.

1.2 Interaction en dehors de la sphère virtuelle

Les techniques *Two-axis Valuator*, *Trackball* et *Arcball* permettent une rotation selon l'axe Z (l'axe passant par le centre de la sphère et le point de vue) lorsque le curseur de souris se rapproche du bord de la sphère dans le plan image. Cette manipulation est délicate parce qu'il est difficile de suivre parfaitement la circonférence de la sphère, mais surtout parce que l'utilisateur contrôle alors plusieurs axes de rotation simultanément. Pour faciliter la rotation selon l'axe Z, ces techniques utilisent l'espace extérieur à la sphère. Ainsi, tout point situé en dehors de la sphère est projeté vers le point le plus proche de la circonférence [31, 133] permettant ainsi de contrôler uniquement la rotation selon l'axe Z.

Un autre problème dans les versions de la *Trackball* et de l'*Arcball* proposées par Chen [31] et Shoemake [133] est la discontinuité. En effet, si le curseur passe de l'intérieur de la sphère à l'extérieur (ou l'inverse), l'angle de rotation selon l'axe X et/ou Y passe d'une certaine valeur non nulle à une valeur nulle puisque la projection du point de la souris n'est plus sur la sphère, ce qui provoque une « discontinuité » dans la rotation [11, 58]. Pour supprimer cet effet, Bell a proposé de ne plus projeter le curseur de souris sur une sphère,

mais sur une surface combinant une sphère et une hyperbole [11]. La surface composée permet alors une variation continue de l'angle de rotation entre un curseur de souris passant de l'intérieur à l'extérieur de la sphère (ou l'inverse). Néanmoins, cette solution ne permet plus de contrôler uniquement la rotation de l'objet selon l'axe Z quand le curseur est en dehors de la sphère.

Certains logiciels 3D utilisent une tout autre approche dans le comportement de la rotation lors d'une interaction en dehors de la sphère. Par exemple, Blender¹ et Unity² permettent à l'utilisateur d'orienter un objet selon l'axe défini entre le centre de la sphère et le curseur de souris. Plus le curseur s'éloigne du centre de la sphère, plus l'objet tourne selon cet axe, permettant même d'effectuer plusieurs rotations complètes (supérieures à 360°) sans débrayages supplémentaires.

1.3 Comparaisons des sphères virtuelles

Plusieurs expérimentations ont été proposées dans la littérature pour comparer les différentes techniques de rotation 3D sur les performances des participants et leurs retours subjectifs dans des tâches d'orientation [10, 31, 62, 159, 70, 130]. Ces tâches affichent généralement deux objets 3D, un objet cible et un objet à manipuler. L'objectif est de faire correspondre l'orientation de l'objet à orienter avec celle de l'objet cible aussi précisément que possible en tournant l'objet 3D manipulé selon les trois axes de rotation, l'orientation de l'objet cible étant souvent définie de manière aléatoire [32, 159]. Chen n'a pas trouvé de différence significative entre les techniques *Two-Axis-Valuator* et *Trackball*, que ce soit pour le temps de réalisation ou la précision [31], de même qu'Hinckley *et al.* [62] entre la *Trackball* de Chen [31] et l'*Arcball* de Shoemake [133], ou que Zhao *et al.* [159] entre les techniques de Shoemake et Bell [11, 133]. Blade *et al.* [10] ont montré que les utilisateurs étaient statistiquement plus rapides et ont préféré utiliser les techniques *Two-Axis-Valuator* [31] et l'*Arcball* de Shoemake [133] que la *Trackball* [31] ou l'*Arcball* de Bell [11]. Plus récemment, Rybicki *et al.* [130] ont montré que leurs participants étaient plus précis en utilisant le manipulateur 3-axes (*Discrete Sliders* pour une rotation simple selon un seul axe de rotation). Ils ont montré aussi que les participants étaient significativement plus rapides pour effectuer une tâche de rotation complexe (plusieurs axes de rotation) avec les techniques basées sur une sphère virtuelle (*Arcball* et *Two-Axis-Valuator*). Encore une fois, il n'y a pas de différences significatives entre les deux techniques de sphères virtuelles qui ont été trouvées.

Enfin, une notion peu abordée dans les articles est la notion de fonction de transfert de rotation. Une telle fonction permet d'appliquer un gain (constant, linéaire, etc) entre l'amplitude de la rotation effectuée en entrée sur la sphère de contrôle et celle appliquée à l'objet 3D contrôlé. Poupyrev *et al.* ont proposé une modélisation du calcul du gain à

1. Blender – <https://www.blender.org>

2. Unity – <https://unity.com/fr>

appliquer sur un quaternion, permettant d'en changer son amplitude [124, 125] en utilisant la fonction *puissance* sans modifier l'orientation. Malheureusement, leur publication ne propose pas de comparaison de différentes fonctions.

2 Étude : influence de la taille de la sphère de contrôle

Les diverses expérimentations proposées dans la littérature se sont principalement intéressées à la comparaison de différentes techniques de rotation 3D sur les performances des utilisateurs. Cependant, d'autres facteurs, qui n'ont pas été étudiés, peuvent aussi influencer les performances et le comportement de l'utilisateur, notamment la taille de la sphère de contrôle. La section suivante décrit donc l'étude que nous avons menée dans le but de déterminer l'influence de la sphère virtuelle de contrôle sur les performances et le comportement des utilisateurs lors d'une tâche de rotation 3D. Notre première hypothèse H1 est que la taille de la sphère virtuelle de contrôle a une influence directe sur les performances des utilisateurs. Notamment, sachant qu'une petite taille de sphère de contrôle devrait permettre des variations d'angles rapides avec un manque de précision, et qu'une grande taille de sphère devrait permettre une plus grande précision, mais une variation d'angle plus lente, alors, étant donné trois tailles de sphères, une petite, une moyenne, une grande, le temps de réalisation des tâches devrait être inférieur pour une taille de sphère moyenne vis-à-vis des petites et grandes tailles. Notre deuxième hypothèse H2, est que la taille de la sphère a une influence sur le comportement des utilisateurs lors d'une tâche de rotation. En effet, la taille de la sphère définit la variation d'angle de rotation, mais aussi la surface de la zone d'interaction affichée à l'écran. Nous supposons alors que plus la surface d'interaction est grande, plus les gestes des utilisateurs devraient être grands pour compenser la perte de vitesse de rotation.

2.1 Méthode d'expérimentation et dispositif

En raison de la situation sanitaire actuelle et afin de faire passer l'expérience à un maximum d'utilisateurs, nous avons développé une application en ligne, en JavaScript, basée sur la librairie 3D Three.js³. Cette application, testée avec les trois navigateurs couramment utilisés⁴ (Google Chrome, Mozilla Firefox et Apple Safari), propose un mode expérience⁵ et un mode bac à sable⁶ afin de tester divers paramètres. L'application communique avec un serveur durant chaque expérience, ce qui permet de sauvegarder à intervalle régulier les données collectées des participants.

3. Three.js – <https://threejs.org>

4. Statistiques d'utilisation des navigateurs sur ordinateurs de bureau – <https://gs.statcounter.com/browser-market-share/desktop/worldwide>

5. Mode expérience – <https://ns.inria.fr/loki/3drotations/?xp=true>

6. Mode bac à sable – <https://ns.inria.fr/loki/3drotations/>

L'expérience s'est déroulée à distance et donc, sans contrôle direct sur les dispositifs utilisés par les participants. La résolution, la taille et la définition des écrans utilisés peuvent alors varier. Afin que la taille physique (*c.-à-d.* affichée à l'écran) de la sphère virtuelle de contrôle soit identique pour chacun des participants lors de l'expérience, chaque participant devait calibrer l'application en faisant correspondre la taille d'un rectangle affiché à l'écran avec la taille d'une carte bleue. Sachant que la taille d'une carte bleue est fixe et connue, il est alors possible de déduire la densité de pixel de l'écran permettant ainsi d'uniformiser les tailles de sphères virtuelles entre les différents dispositifs. Néanmoins, puisque nous testons des tailles de sphère allant jusque 200 mm, nous n'avons recruté que des participants disposant d'un écran d'au moins 15 pouces de diagonale afin que l'utilisateur puisse interagir avec l'intégralité de la sphère.

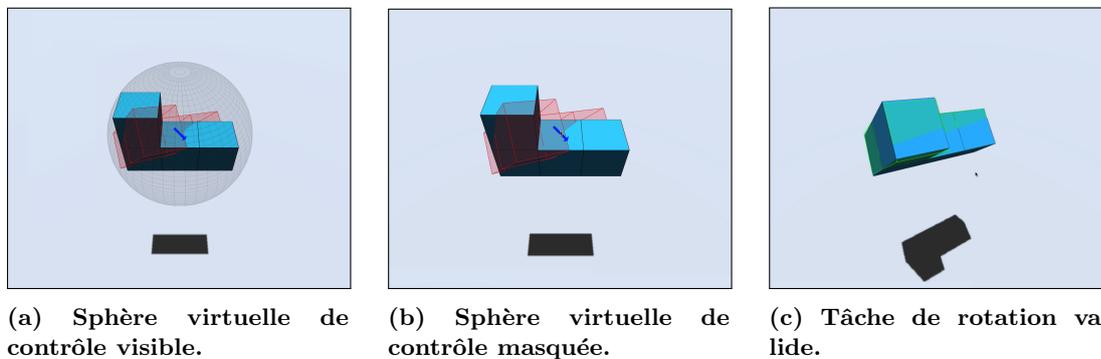


Figure 4.3 – La tâche de rotation consistait à modifier l'orientation de l'objet bleu afin qu'elle corresponde à celle de l'objet cible en rouge transparent. La sphère virtuelle de contrôle (a) n'était pas visible pour l'utilisateur (b). L'objet cible passe en vert quand l'orientation de l'objet est sous le seuil de tolérance de la tâche (c). Enfin, une flèche bleue indiquait la direction de l'axe de rotation optimale pour effectuer la tâche (a) et (b).

2.2 Procédure d'expérimentation

Une fois connecté sur le site de l'expérience, une page d'accueil affichait les instructions aux participants, qui étaient de réaliser une séquence de rotations 3D le plus rapidement possible. Une image animée permettait aux participants de bien comprendre la tâche demandée qui était une tâche dite d'« amarrage » (*cf.* figure 4.3b), dont l'objectif est de faire correspondre l'orientation d'un objet manipulé avec celle d'un objet cible le plus rapidement possible.

Dans notre cas, l'objet à manipuler est affiché en bleu et sans transparence dans la scène 3D, l'objet cible est quant à lui affiché en rouge et avec une certaine transparence afin de ne pas masquer l'objet manipulé (*cf.* figure 4.3b). Un retour visuel est donné à l'utilisateur quand la tâche est valide, c'est-à-dire, quand le seuil de tolérance est atteint, l'objet cible passe alors de couleur rouge à une couleur verte, et inversement si la tâche redevient invalide (*cf.* figure 4.3c). Chaque tâche de rotation est validée lorsque l'utilisateur relâche

le bouton de la souris et que la différence d'orientation entre l'objet manipulé et l'objet cible (*c.-à-d.* la valeur de l'angle de la plus petite rotation à effectuer pour passer de l'objet manipulé à l'objet cible) est inférieure à un certain seuil de tolérance prédéfini en degré (et donc que l'objet cible est affiché en vert). Une animation affichée avec les instructions permettait de visualiser la tâche à effectuer avant de commencer l'expérimentation. Après avoir lu les instructions de la tâche demandée, chaque participant a rempli un formulaire demandant l'âge, le sexe, le dispositif utilisé (souris ou pavé tactile) et enfin, s'il était familier avec les logiciels de modélisation 3D.

La technique de rotation que nous avons utilisée est la technique proposée par Shoemake [133]. Néanmoins, nous nous intéressons dans cette étude principalement à l'influence de la taille de la sphère lorsque l'utilisateur interagit avec celle-ci. Par conséquent, contrairement à l'expérience conduite par Zhao *et al.* [159], nos tâches de rotation présentées aux participants ne nécessitaient pas de rotation selon l'axe Z. De ce fait, nous avons modifié la technique de rotation proposée par Shoemake en désactivant toute interaction en dehors de la sphère, et donc les rotations selon l'axe Z. Enfin, nous avons choisi de ne pas afficher la sphère virtuelle de contrôle, ni les indicateurs visuels classiques pour décrire l'orientation d'un objet 3D tels que les principaux axes de rotation [159] car d'une part, la taille de la sphère virtuelle est un des facteurs que nous étudions et d'autre part, les logiciels actuels proposant un contrôle de rotation via une sphère virtuelle n'affichent pas cette sphère et donc la surface d'interaction disponible pour l'utilisateur.

Cependant, puisque nous nous intéressons uniquement aux performances et au comportement des participants lors d'une tâche de rotation, et en particulier à ces performances d'un point de vue moteur, nous avons rajouté une flèche indiquant la direction optimale de la rotation à effectuer pour compléter la tâche. Cette flèche aide les participants à savoir quel mouvement effectuer avec le curseur pour orienter l'objet, diminue la charge mentale requise et permet aux participants de se concentrer sur la charge moteur (voir la flèche bleue dans la figure 4.3b). La flèche, de taille fixe, est initialement visible au début de chaque tâche et se masque progressivement dès que le participant clique sur la sphère. Elle redevient visible dès que l'utilisateur relâche le bouton de la souris. Si la tâche n'est pas terminée en un seul mouvement de souris, la flèche est mise à jour et indique la nouvelle direction optimale à suivre pour compléter la tâche.

L'expérience a suivi un protocole expérimental $4 \times 2 \times 2 \times 3 \times 4$ de type intra-sujets, composé des facteurs suivants : 4 *tailles* de la sphère virtuelle de contrôle (50, 100, 150 et 200 mm), 2 *amplitudes* de la rotation à effectuer (35° et 70°), 2 *tolérances* de l'orientation finale (6° et 3°), 3 blocs et 4 répétitions utilisant un axe de rotation différent. Chaque répétition nécessitait successivement une rotation selon une des huit directions prédéfinies pour correspondre à une rotation à effectuer selon les axes X et Y uniquement : *Ouest*, *Est*, *Sud-Ouest*, *Nord-Est*, *Sud*, *Nord*, *Sud-Est* et *Nord-Ouest*. Chaque direction d'une tâche était diamétralement opposée à celle de la précédente. L'ordre des tailles de sphères

virtuelles a été contrebalancé entre les participants. Nous avons choisi pour chaque taille de sphère virtuelle que les 4 combinaisons des facteurs *amplitude* et *tolérance* soient ordonnées de la plus facile (petite amplitude et grande tolérance) à la plus difficile (grande amplitude et petite tolérance). Les deux autres conditions ont été choisies de manière arbitraire car nous ne savions pas si augmenter l'amplitude rend la tâche plus difficile que réduire la tolérance. L'ensemble des facteurs donne ainsi un protocole expérimental de $4 \times 2 \times 2 \times 3 \times 4 = 192$ tâches de rotation par participant.

Entre chaque taille de sphère virtuelle, les participants devaient remplir un questionnaire leur demandant leur degré d'accord avec les trois affirmations suivantes : « *J'ai été rapide pour compléter la tâche demandée* », « *J'ai été précis lors de la réalisation de la tâche demandée* » et « *J'ai trouvé la tâche demandée fatigante* ». La réponse à chaque question devait être choisie sur une échelle de Likert à 7 niveaux de « *Pas du tout d'accord* » à « *Tout à fait d'accord* ». Enfin, les participants pouvaient faire une pause après les trois blocs complétés pour chaque combinaison des facteurs *taille*, *amplitude* et *tolérance*, c'est-à-dire toutes les 12 tâches de rotation. Le temps moyen requis pour compléter l'ensemble des 192 tâches de rotation était d'environ 35 minutes.

Afin de pouvoir analyser les données produites par les participants, la résolution de l'écran utilisé et les coordonnées de souris de l'utilisateur ont été enregistrées tout au long de l'expérience.

2.3 Résultats

24 participants (19 hommes, 4 femmes et 1 personne ne préférant pas répondre) ont participé dans cette expérimentation avec un âge moyen de 28.75 ans ($\sigma = 8.46$). 20 participants ont utilisé une souris d'ordinateur et 4 un pavé tactile (interne ou externe). Parmi les 24 participants, seulement 8 ont déclaré être familier avec des outils de modélisation ou de manipulation 3D (*p.ex.* Blender, Maya, Unity, etc). Afin de ne pas tenir compte des temps de réalisation aberrants pour les tâches de rotation, dans lesquelles les utilisateurs ont éventuellement pu rencontrer des problèmes, nous avons supprimé pour chaque participant l'ensemble des tâches dont le temps de réalisation est quatre fois supérieur à l'écart-type du total des temps de réalisation pour ce participant. Nous avons donc supprimé en moyenne 5.8 ($\sigma = 2.5$) tâches par participant, soit un total de 139 tâches sur les 4608 tâches effectuées (3%).

2.3.1 Temps de réalisation

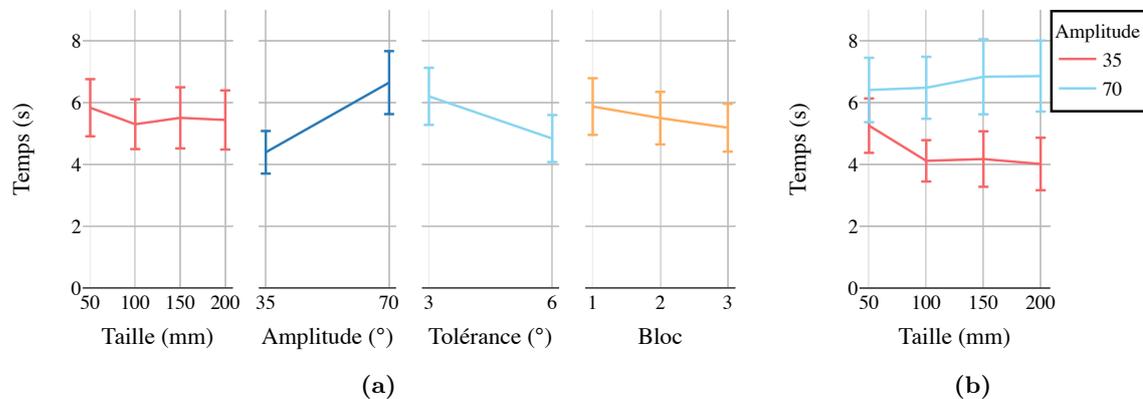


Figure 4.4 – Moyenne des temps de réalisation des tâches de rotation pour les facteurs *taille*, *amplitude*, *tolérance* et *bloc* (a). Interaction des facteurs *taille* et *amplitude* sur le temps de réalisation (b). Les barres d'erreur représentent l'intervalle de confiance à 95%.

Nous avons mesuré le temps effectué par les participants pour chaque tâche de rotation entre le moment où la tâche est affichée et l'orientation validée. Contrairement à ce que nous attendions, les ANOVAs à mesures répétées n'ont pas montré d'effet statistique significatif de la taille de la sphère sur le temps de réalisation de la tâche de rotation ($F_{3,69} = 1.10, p > 0.05$), ce qui ne nous permet donc pas d'accepter l'hypothèse H1 qui suppose que la taille de la sphère a une influence directe sur le temps de réalisation de la tâche de rotation. Néanmoins, nous avons trouvé un effet significatif des facteurs *amplitude* ($F_{1,23} = 95.24, p < 0.001, \eta_p^2 = 0.81$) et *tolérance* ($F_{1,23} = 123.09, p < 0.001, \eta_p^2 = 0.84$) sur le temps de réalisation (cf. figure 4.4a). Les participants ont été 22% plus rapides avec une tolérance de 6° ($4.8s \pm 0.7$) qu'avec une tolérance de 3° ($6.2s \pm 0.9$). De même, ils ont été 34% plus rapides à réaliser la tâche de rotation avec une amplitude de 35° ($4.4s \pm 0.7$) qu'avec une amplitude de 70° ($6.6s \pm 1.0$). Enfin, les ANOVAs à mesures répétées ont aussi montré un effet significatif du facteur *bloc* ($F_{2,46} = 15.37, p < 0.001, \eta_p^2 = 0.40$) suggérant un effet d'apprentissage. Une analyse post-hoc supplémentaire n'a montré une différence significative ($p < 0.005$) qu'entre les blocs 1 ($5.9s \pm 0.9$) et 3 ($5.2s \pm 0.8$), ce qui représente une amélioration de 11% et nous permet de négliger ce faible effet d'apprentissage de la tâche. Enfin, les ANOVAs à mesures répétées ont montré une interaction significative entre les facteurs *taille* et *amplitude* ($F_{3,69} = 9.85, p < 0.001, \eta_p^2 = 0.30$). La figure 4.4b illustre cette interaction et montre que le temps de réalisation augmente doucement avec la taille de la sphère lorsque l'amplitude de la tâche de rotation est de 70° (50mm : $6.4s \pm 1.0$, 100mm : $6.5s \pm 1.0$, 150mm : $6.8s \pm 1.2$ et 200mm : $6.9s \pm 1.1$), mais diminue lorsque l'amplitude de la tâche est de 35° (50mm : $5.2s \pm 0.9$, 100mm : $4.1s \pm 0.7$, 150mm : $4.1s \pm 0.9$ et 200mm : $4.0s \pm 0.8$). De ce fait, la faible vitesse de rotation induite par une plus grande de sphère

semble influencer les performances des utilisateurs lorsqu'une grande rotation est requise (il faut plus de temps pour parcourir la distance) et semble améliorer les performances pour des petites rotations (gain de précision).

2.3.2 Nombre de mouvements de souris

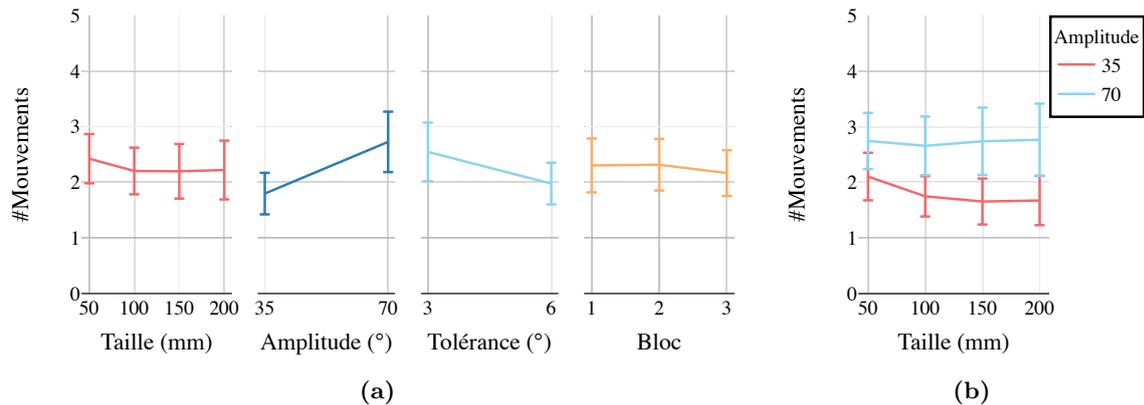


Figure 4.5 – Moyenne du nombre de mouvements du curseur de la souris lors des tâches de rotation pour les facteurs *taille*, *amplitude*, *tolérance* et *bloc* (a). Interaction des facteurs *taille* et *amplitude* sur le nombre de mouvements (b). Les barres d'erreur représentent l'intervalle de confiance à 95%.

Nous avons mesuré le nombre de mouvements du curseur de souris effectués par les participants dans chaque tâche de rotation, c'est-à-dire, le nombre de fois où ils ont cliqué, déplacé et relâché le bouton. Les ANOVAs à mesures répétées ont montré un effet significatif des facteurs *amplitude* ($F_{1,23} = 58.19, p < 0.001, \eta_p^2 = 0.72$) et *tolérance* ($F_{1,23} = 45.33, p < 0.001, \eta_p^2 = 0.66$) sur le nombre de mouvements (*cf.* figure 4.5a). Les participants ont effectué en moyenne 25% de mouvements en plus avec une tolérance de 3° (2.5 ± 0.5) qu'avec une tolérance de 6° (2.0 ± 0.3). Ceci peut s'expliquer par la fait qu'il est plus difficile de valider la tâche de rotation lorsque la tolérance est plus faible et donc, peut nécessiter d'effectuer des mouvements supplémentaires pour corriger l'orientation de l'objet manipulé. De même, comme illustré figure 4.5a ils ont effectué 50% de mouvements de souris en plus avec une amplitude de 70% (2.7 ± 0.5) qu'avec une amplitude de 35° (1.8 ± 0.4). Par ailleurs, les mesures ont montré une interaction entre les facteurs *taille* et *amplitude* ($F_{3,69} = 3.09, p = 0.059, \eta_p^2 = 0.12$). La figure 4.5b illustre cette interaction et montre que le nombre de mouvements augmente avec la taille de la sphère lorsque l'amplitude de la tâche de rotation est de 70°, mais diminue lors que l'amplitude de la tâche est de 35°.

2.3.3 Variation d'angle

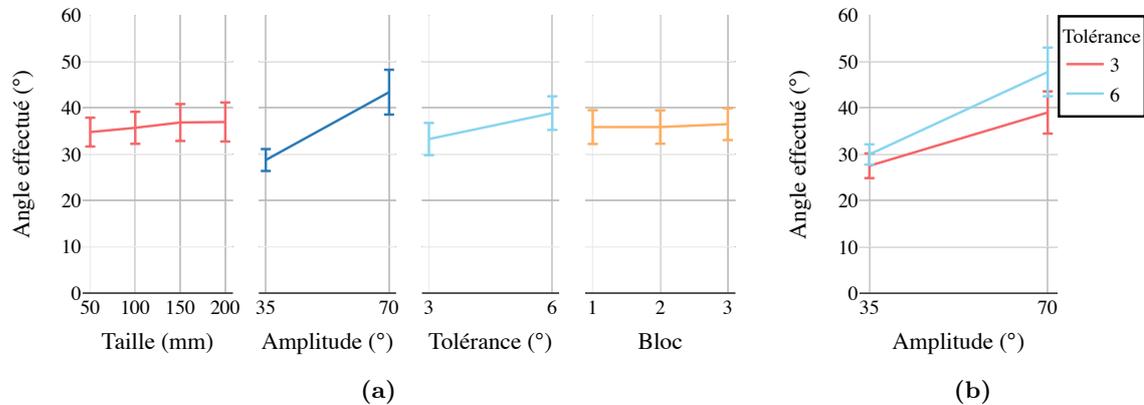


Figure 4.6 – Moyenne des variations d’angles de l’objet manipulé effectuées par les participants lors des tâches de rotation pour les facteurs *taille*, *amplitude*, *tolérance* et *bloc* (a). Interaction des facteurs *tolérance* et *amplitude* sur la variation d’angle (b). Les barres d’erreur représentent l’intervalle de confiance à 95%.

Nous avons mesuré la variation d’angle moyenne effectuée par chaque mouvement de souris, c’est-à-dire la valeur de l’angle entre l’orientation de l’objet au moment du clic de la souris et celle de l’objet lors du relâché du bouton de la souris. Les ANOVAs sur mesures répétées ont montré un effet significatif des facteurs *amplitude* ($F_{1,23} = 110.14$, $p < 0.001$, $\eta_p^2 = 0.83$) et *tolérance* ($F_{1,23} = 148.89$, $p < 0.001$, $\eta_p^2 = 0.87$) sur la variation d’angle moyenne de l’objet manipulé (cf. figure 4.6a). Quant aux facteurs *tolérance* et *amplitude*, les participants ont montré avoir effectué une rotation 17% plus grande avec une tolérance de la tâche de 6° ($38.8^\circ \pm 3.6$) contre 3° ($33.2^\circ \pm 3.5$) et 50% plus grande avec une amplitude de 70° ($43.3^\circ \pm 4.8$) contre une amplitude de 35° ($28.7^\circ \pm 2.7$). Les mesures ont aussi montré une interaction entre les facteurs amplitude et tolérance ($F_{1,23} = 33.57$, $p < 0.001$, $\eta_p^2 = 0.59$). La figure 4.6b illustre cette interaction et montre que la variation d’angle moyenne de l’objet manipulé croît avec l’amplitude plus rapidement avec une tolérance de 6° qu’avec une tolérance de 3°. La différence de variation d’angle peut s’expliquer par le fait que les participants devaient à la fois effectuer de plus grandes rotations pour réaliser la tâche lorsque le l’amplitude augmente, mais aussi utiliser de petites rotations de correction nécessaires pour valider la tâche lorsque la tolérance est faible.

2.3.4 Précision

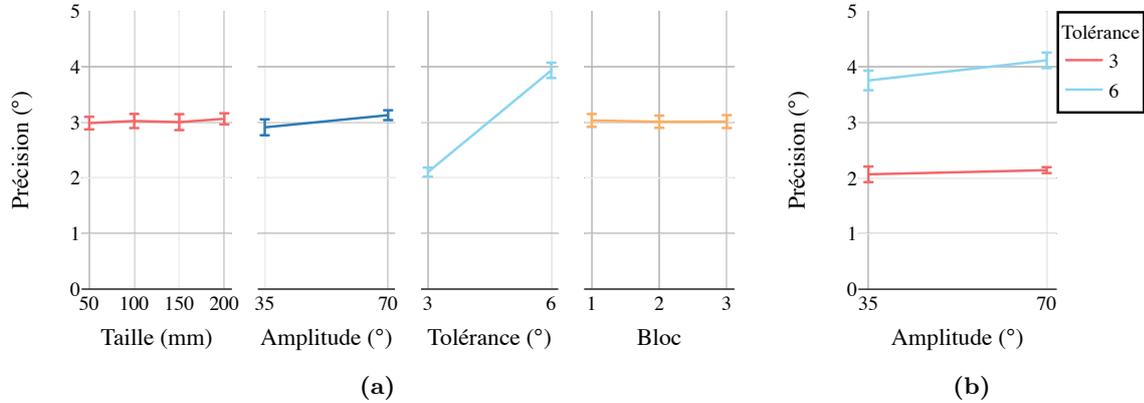


Figure 4.7 – Précision moyenne des participations lors des tâches de rotation pour les facteurs *taille*, *amplitude*, *tolérance* et *bloc* (a). Interaction des facteurs *tolérance* et *amplitude* sur la précision (b). Les barres d'erreur représentent l'intervalle de confiance à 95%.

La tâche de rotation demandée ne pouvait échouer, car les participants devaient faire correspondre l'orientation des deux objets sous le seuil de tolérance. Néanmoins, il est possible de calculer la précision du participant en mesurant la valeur de l'angle entre l'orientation finale et validée effectuée par le participant et l'orientation cible (calculée via une différence de quaternion). Les ANOVAs à mesures répétées ont montré un effet significatif des facteurs *amplitude* ($F_{1,23} = 13.96$, $p = 0.001$, $\eta_p^2 = 0.38$) et *tolérance* ($F_{1,23} = 1538.78$, $p < 0.001$, $\eta_p^2 = 0.99$) sur la précision des participants dans la tâche de rotation (cf. figure 4.7a). En effet, les participants ont été 46% plus précis avec une tolérance de 3° ($2.1^\circ \pm 0.08$) qu'avec une tolérance de 6° ($3.9^\circ \pm 1.14$) et 6% plus précis avec une amplitude de 35° ($2.9^\circ \pm 0.14$) qu'avec une amplitude de 70° ($3.1^\circ \pm 0.09$). Les mesures ont aussi montré une interaction entre les facteurs amplitude et tolérance ($F_{1,23} = 12.60$, $p = 0.002$, $\eta_p^2 = 0.35$). La figure 4.7b illustre cette interaction et montre que la précision diminue avec l'amplitude plus rapidement avec une tolérance de 6° qu'avec une tolérance de 3°. Une explication possible est l'utilisation par les participants de plus grands gestes de rotations lorsque l'amplitude est élevée, entraînant donc une perte de précision.

2.3.5 Retours subjectifs des participants

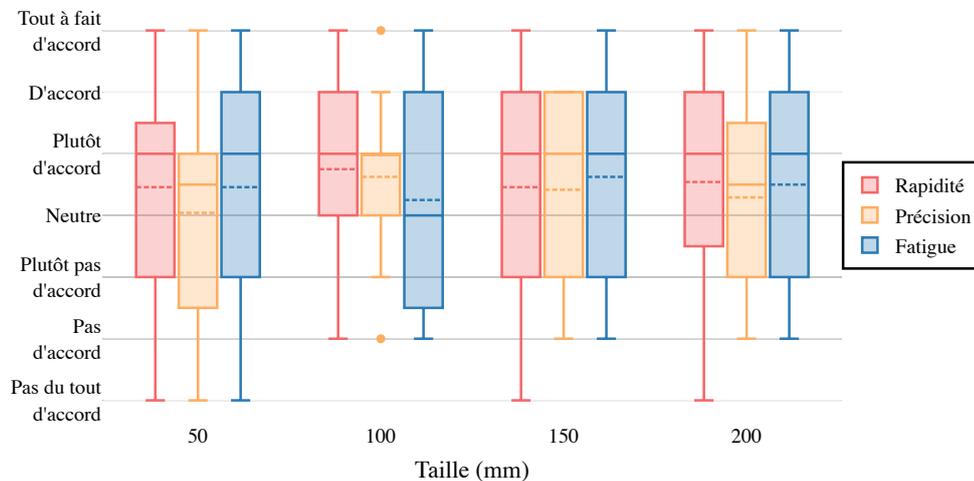


Figure 4.8 – Retours subjectifs des participants sur l’estimation de leur rapidité, leur précision et leur niveau de fatigue des tâches de rotation sur une échelle de Likert à 7 valeurs (de *Pas du tout d’accord* à *Tout à fait d’accord*) pour les quatre tailles de sphère d’interaction.

Nous avons analysé les réponses des participants concernant leur rapidité, leur précision et leur niveau de fatigue lors des tâches de rotation grâce à un test de Friedman. L’analyse n’a montré aucun effet de la taille de la sphère sur les réponses des participants ni pour la rapidité ($p=0.41$), la précision ($p=0.40$) ou le niveau de fatigue ($p=0.45$). Les réponses ont en moyenne été situées entre les valeurs *neutre* et *plutôt d’accord* de l’échelle de Likert à 7 valeurs proposée aux participants pour chacune des conditions (*cf.* figure 4.8).

Discussion

Discussion des résultats

Comme attendu, l’amplitude et la tolérance de la tâche de rotation ont toutes deux eu un effet significatif sur les performances des participants, puisqu’ils correspondent ensemble à l’indice de difficulté de la tâche de rotation. La loi de Fitts pour le pointage à une dimension estime le temps de mouvement DT à $DT = a + b * ID$, avec ID l’indice de difficulté de la tâche décrit par $ID = \log_2(1 + \frac{D}{W})$ [101], avec D la distance entre le point de départ et la cible (ici, l’amplitude de rotation) et W , la largeur de la cible (ici la tolérance de la tâche de rotation). Nous observons clairement l’impact de l’indice de difficulté de la tâche sur les temps de réalisation (24% plus rapides pour la grande tolérance, 34% plus rapides pour la petite amplitude, figure 4.4) et le nombre de mouvements effectués (25% de mouvements en plus pour la petite tolérance, 50% de mouvements en plus pour la grande

amplitude, figure 4.5).

Nous avons par ailleurs observé que la variation d'angle moyenne des mouvements de la souris croît avec l'amplitude plus rapidement avec une tolérance de 6° qu'avec une tolérance de 3°. Cette différence montre que les plus grands gestes effectués par les participants pour la grande amplitude sont compensés par un nombre de petits gestes de correction lorsque la tolérance est faible. Ceci se confirme lorsque nous observons la différence significative du nombre de mouvements en fonction de la tolérance. Les participants ont en effet effectué 25% de mouvements en plus pour la tolérance de 3° que celle de 6°. Bien que la moyenne de l'angle effectué par les mouvements de souris augmente avec la taille de la sphère (cf figure 4.6a), nous sommes étonnés de ne pas avoir observé d'effet significatif de la taille de la sphère. En effet, une plus grande sphère permet théoriquement aux utilisateurs d'effectuer de plus grands mouvements de rotation. La non-présence d'un effet pourrait se justifier dans un premier temps par la non-visibilité de la sphère d'interaction, ne laissant aucune indication visuelle aux utilisateurs concernant la surface d'interaction disponible. Dans un second temps, plus la taille de la sphère augmente, moins la variation de rotation unitaire est importante pour un même déplacement de souris, compensant probablement une surface d'interaction plus grande.

Enfin, bien que nous n'ayons pas observé d'influence directe de la taille de la sphère sur la précision, les interactions entre les facteurs *taille* et *amplitude* sur le temps de réalisation (cf. figure 4.4b) et le nombre de mouvements de la souris (cf. figure 4.5b) montrent que les participants, lorsque la taille de sphère augmente, sont légèrement de plus en plus rapides avec une petite amplitude et effectuent de moins en moins de mouvements, justifiant un gain de précision. À l'inverse, ils sont légèrement de plus en plus lents avec une grande amplitude et effectuent de plus en plus de mouvements, montrant la perte de rapidité entraînée par l'augmentation de la taille de la sphère et la diminution de la vitesse de rotation.

Stratégies de rotation

En raison de la situation sanitaire actuelle, l'étude que nous avons menée auprès de 24 participants s'est déroulée à distance, ne nous permettant pas de visualiser directement le comportement des participants lors des tâches de rotation, notamment les mouvements physiques de la souris des participants ou leurs doigts sur un pavé tactile. Néanmoins, ayant enregistré au cours de l'expérience tous les événements de souris réalisés par chacun des participants pour chacune des tâches de rotation, il nous est possible de rejouer les mouvements réalisés dans l'application et d'observer le comportement des participants. Nous avons observé dans notre étude que la taille de la sphère n'avait pas un impact direct sur les performances des utilisateurs lors des tâches de rotation que ce soit pour le temps de réalisation des tâches de rotations, la précision de l'utilisateur, le nombre de mouvements de la souris ou les variations d'angles effectués. Cependant, la variance

du temps de réalisation, du nombre de mouvements et de l'angle effectué suggère que la taille de la sphère a dû avoir un impact indirect, notamment sur les stratégies de rotation employées par les participants. Voir par exemple la figure 4.4a où les temps de réalisation pour chaque taille de sphère sont les suivants : 50mm : $5.8sec \pm 0.92$; 100mm : $5.2sec \pm 0.8$; 150mm : $5.5sec \pm 1.0$ et 200mm : $5.4sec \pm 0.95$.

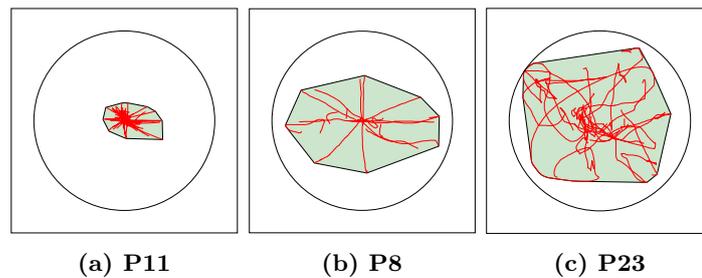


Figure 4.9 – Visualisation des mouvements de souris, mis à l'échelle par rapport à la sphère d'interaction représentée par le cercle, pour les participants P11, P8 et P23 pour les tâches de rotation ayant une taille de sphère de 150mm, une amplitude de 35° et une tolérance de 6° .

Ainsi, en normalisant chaque mouvement de souris par rapport à la taille de la sphère, on observe par exemple dans la figure 4.9 que les mouvements de souris des participants P11, P8, et P23 adoptent une disposition spatiale complètement différentes, illustrant 3 stratégies de rotation. P11 semble n'avoir effectué que des petits débrayages de souris centrés sur la sphère impliquant un grand nombre de mouvements pour compléter les tâches de rotation (*cf.* figure 4.9a). P8 semble avoir effectué de plus larges mouvements (la surface de la sphère étant plus occupée) selon les huit axes de rotation de la tâche en déplaçant le curseur de souris du centre de la sphère vers l'extérieur (*cf.* figure 4.9b). Enfin, P23 ne semble pas avoir appliqué de stratégie spécifique. Ce participant semble avoir été « perdu » dans les rotations à effectuer puisque ses mouvements de souris ne suivent pas du tout les axes de rotations des tâches à effectuer (*cf.* figure 4.9c).

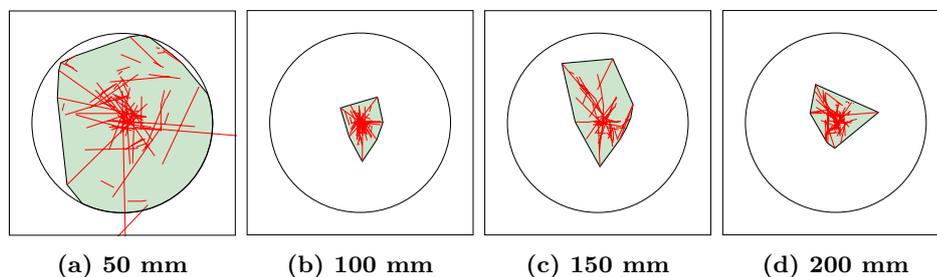


Figure 4.10 – Visualisation des mouvements de souris, mis à l'échelle par rapport à la sphère d'interaction représentée par le cercle, pour le participant P19, pour les tâches de rotations ayant une amplitude de 35° , une tolérance de 3° et une taille de sphère de 50, 100, 150 et 200mm.

Les stratégies de rotation diffèrent entre les participants, mais aussi pour un seul

participant entre plusieurs tailles de sphère. Par exemple, la figure 4.10 montre que le participant P19 a effectué des plus grands mouvements de souris pour une taille de sphère de 50mm que pour les tailles de 100, 150 et 200mm, ce qui est étonnant sachant que plus la taille de sphère la plus petite, plus les variations de rotation sont importantes. Ces visualisations suggèrent que la taille de sphère de contrôle aurait effectivement un impact, certes indirect, sur les performances et le comportement des utilisateurs qu'il conviendrait d'étudier plus en détail dans une future étude.

Conclusion

Nous avons présenté dans ce chapitre un bref état de l'art des techniques actuelles de rotation sur ordinateur de bureau via des sphères virtuelles de contrôles. Les travaux existants se sont exclusivement intéressés à la comparaison des différentes techniques proposées dans la littérature via des expériences contrôlées. Nous avons conduit une première étude qui cherche à mesurer l'impact de la taille de la sphère virtuelle de contrôle sur les performances et le comportement des utilisateurs via des tâches de rotation. Notre étude a donc comparé quatre valeurs de sphère (de 50mm à 100mm), basées sur une même technique de rotation 3D, soit une version modifiée de la sphère virtuelle de contrôle proposée par Shoemake [133]. L'expérience contrôlée, menée à distance auprès de 24 participants n'a pas montré d'effet significatif de la taille de la sphère de contrôle sur les temps de réalisation des tâches de rotation, mais a montré que la taille pouvait avoir un éventuel impact sur leur comportement.

Ce travail a été effectué dans les derniers mois de cette thèse et, de ce fait, nous n'avons malheureusement pas pu approfondir en détail l'analyse des mouvements de souris des participants par manque de temps. Les résultats préliminaires, présentés par des visualisations en discussion de ce chapitre, suggèrent que les participants auraient employé différentes stratégies pour réaliser les tâches de rotation. Par exemple, effectuer de petits mouvements de rotations aux environs du centre de la sphère et débrayer beaucoup, ou alors effectuer de longs mouvements de rotation du centre vers l'extérieur de la sphère de contrôle. Nous pouvons aussi remarquer que beaucoup de participants ne semblent pas avoir appliqué de stratégies de rotation spécifiques, pouvant potentiellement expliquer la variance obtenue dans les temps de réalisation des tâches de rotations.

Conclusion

Nous avons établi dans le chapitre 1 un état de l'art des connaissances et outils pour la production de figures illustratives d'interaction, décomposé en deux parties. La première a permis de faire un tour d'horizon des différents travaux de la littérature portant sur l'illustration de scénarios interactifs. Nous n'avons trouvé que trois taxonomies, deux se concentrant sur des domaines extérieurs à l'IHM tels que le design ou l'éducation et la dernière aux stratégies d'illustration d'une partie de l'IHM, l'interaction gestuelle [109]. Néanmoins, malgré la portée limitée de ces taxonomies, nous avons aussi relié les concepts introduits par ces travaux et ceux du domaine de l'illustration, comme la bande dessinée. La seconde partie de l'état de l'art a, quant à elle, permis de présenter les techniques et outils proposés dans la littérature pour la création de figures à base de tracés. Nous avons séparé ces méthodes en deux catégories, celles qui se basent sur des modèles et celles qui reposent sur de la mise en scène. Nous avons vu que les méthodes basées sur des modèles requièrent un long processus créatif manuel (*p.ex.* le traçage de figures), et qu'elles ne permettent généralement pas de modifier la figure après sa production. Les méthodes basées sur la mise en scène permettent de générer efficacement et simplement des figures, via par exemple une mise en scène à distance par démonstration (*p.ex.* Demodraw [35]) ou via un logiciel spécifique (*p.ex.* ComiPo!). Malheureusement, dans l'ensemble, ces techniques ne permettent de générer des figures que dans un domaine particulier tel que les chorégraphies ou les mangas et ne sont donc pas adaptées pour illustrer un scénario interactif.

Dans le chapitre 2, nous nous sommes intéressés à l'identification des stratégies d'illustration utilisées pour produire des figures illustrant des scénarios interactifs. À cette fin, nous avons d'abord établi une nouvelle taxonomie de figures illustrant des scénarios interactifs. Notre taxonomie a été élaborée selon une approche empirique et un processus de codage de 795 figures issues d'articles de recherche des actes de quatre grandes conférences en IHM. Cette taxonomie unifie les éléments de conception introduits par les travaux de la littérature tout en proposant de nouveaux codes permettant de couvrir un large spectre des interactions illustrées dans la communauté IHM. Via un outil développé spécifiquement pour explorer notre jeu de figures codées, nous avons ensuite identifié une dizaine de stratégies en analysant les liens entre les codes « quoi » et les codes « comment » codés

pour chaque figure de notre jeu de données. Nous avons groupé ces stratégies en deux catégories, les stratégies de structure qui fournissent des indications sur la manière de traiter les bases de la conception des figures et les stratégies d'interaction qui fournissent des indications sur la représentation d'un élément spécifique de la figure. Ces stratégies peuvent être alors utilisées comme lignes directrices pour les chercheurs ou étudiants novices dans l'illustration de figures d'interaction, mais aussi permettre aux chercheurs plus expérimentés de s'améliorer par une mise en perspectives des figures qu'ils produisent par rapport à celles de la littérature. En outre, nous proposons des outils spécifiques disponibles en ligne permettant de visualiser notre taxonomie, notre jeu de données, mais aussi d'explorer de nouvelles stratégies.

Le chapitre 3 a soutenu l'hypothèse que la mise en scène 3D était une alternative intéressante pour la production de figures à base de tracés. En se basant sur les stratégies identifiées dans notre taxonomie, nous avons développé *Esquisse*, un outil qui met en œuvre un nouveau processus créatif basé sur la manipulation d'objets 3D pour mettre en place une scène en 3D et effectuer un rendu sous forme de figure vectorielle. *Esquisse* fournit des techniques d'interaction spécifiques telles que le système d'ancrage ou la téléportation de caméra, pour faciliter la manipulation des objets 3D, et notamment des armatures. *Esquisse* permet aussi d'illustrer la dynamique via le rendu du mouvement d'un objet à travers un effet stroboscopique ou des flèches. L'étude que nous avons menée auprès de 8 participants suggère que les utilisateurs sont capables de s'adapter à la mise en scène 3D, même sans expérience, et à utiliser les fonctionnalités proposées pour produire rapidement toute une variété de figures illustrant des scénarios interactifs. *Esquisse* peut être alors utile aux utilisateurs qui n'ont pas l'habitude de créer leurs figures à base de tracés à la main, mais aussi en tant qu'outil de prototypage rapide. Sachant que les figures générées sont au format vectoriel, des logiciels d'édition peuvent alors être utilisés pour personnaliser les rendus. Néanmoins, *Esquisse* a montré ses limites lorsque des rotations en trois dimensions sont nécessaires pour positionner un objet, ce qui a été le cas de deux participants de l'étude, suggérant qu'il reste encore des axes d'améliorations possibles pour faciliter ces manipulations.

Le chapitre 4 s'est donc intéressé aux techniques de rotation d'objets 3D basées sur des sphères virtuelles de contrôle. Nous avons proposé une étude contrôlée auprès de 24 participants qui visait à évaluer l'hypothèse que la taille de sphère virtuelle de contrôle avait une influence sur les performances et le comportement des utilisateurs lors de tâche de rotation, hypothèse qui n'avait pas encore été étudiée dans la littérature. Les résultats ont montré que la taille de sphère n'avait pas d'effet significatif direct sur les performances des utilisateurs. Néanmoins, la visualisation des mouvements de souris, les mesures de variation d'angles de rotation et le nombre de mouvements de souris des participants permettent de penser que la taille de la sphère peut avoir une influence sur le comportement des utilisateurs, et met en évidence des stratégies différentes utilisées par les participants

(*p.ex.* des mouvements amples de souris, ou de nombreux petits débrayages). Il convient donc d'analyser plus en détails les diverses stratégies de rotation dans une future étude, et éventuellement mettre en évidence l'influence de la taille de la sphère virtuelle de contrôle.

Travaux futurs

Pistes d'amélioration de la taxonomie

Pour établir notre taxonomie, nous avons extrait des caractéristiques visuelles et des stratégies pour illustrer des scénarios interactifs, largement utilisées par la communauté IHM lorsqu'elle produit ses propres figures. Il pourrait être intéressant dans des travaux futurs d'évaluer l'efficacité des figures utilisées dans notre jeu de données, en termes d'apparence visuelle ou de compréhension de la figure. En effet, si nous pouvons supposer un certain niveau de qualité du jeu de figures, étant donné qu'elles ont toutes été extraites d'articles de recherche publiés dans des conférences importantes en IHM, il est également probable que toutes les figures ne soient pas de qualité équivalente et que certaines d'entre elles transmettent plus efficacement leur signification que d'autres. Il serait alors possible de mesurer l'efficacité de certaines stratégies identifiées dans ce travail par rapport à d'autres. Des travaux futurs pourraient examiner cette question par le biais d'études utilisateurs dans lesquelles les participants seraient confrontés à plusieurs figures illustrant un même contexte interactif, mais employant des stratégies différentes et fourniraient un retour d'information subjectif permettant d'évaluer de manière objective les performances globales de chacune des stratégies.

Pistes d'amélioration d'*Esquisse*

L'idée sur laquelle s'appuie *Esquisse* est de créer une figure vectorielle représentant un scénario interactif à partir de l'agencement d'une scène 3D. Un des points forts de l'utilisation de l'agencement de scènes 3D comme support d'illustration est de pouvoir modifier facilement la scène a posteriori et d'en générer un rendu plus ou moins personnalisé. L'agencement de scènes 3D est aussi une faiblesse d'*Esquisse* que nous avons observée lors de notre étude où certains participants ont rencontré des difficultés à manipuler des objets 3D malgré les techniques d'interaction proposées. Proposer de nouvelles techniques d'interaction permettant de manipuler les objets 3D est une perspective de recherche future, telle que la manipulation d'armatures complexes et la rotation d'objets. Une autre piste de recherche pourrait également s'intéresser à la construction d'une scène 3D à partir d'une photographie ou d'un dessin. Ce processus de construction pourrait également s'intégrer dans le processus employé actuellement, reposant sur un modèle photo. Notamment, la scène 3D pourrait être construite de manière automatique via des techniques d'apprentissage, ou semi-automatiques, où l'utilisateur guide le système pour, par exemple, identifier

les objets présents sur la photographie, définir des points de fuites de l'image afin de générer une carte de profondeur de la scène 3D. L'agencement de scènes 3D dans *Esquisse* s'appuie sur la mise à disposition d'objets 3D fournis par l'outil ou disponibles sur internet où il est facile de trouver une multitude d'objets gratuits ou payants. Évidemment, trouver un modèle 3D pour un dispositif prototype spécifique, comme souvent utilisé en IHM, n'est pas toujours possible. La création manuelle d'un modèle 3D dans un logiciel de modélisation est une tâche complexe, mais nous pouvons espérer à l'avenir que des techniques de reconstruction 3D efficaces soient disponibles pour pallier ce problème, en utilisant par exemple des techniques de vision par ordinateur [32, 81, 161], d'infographie [69, 92, 155] ou des capteurs spécifiques comme le capteur LiDAR des derniers iPhones [9]. De plus, l'outil *Esquisse* pourrait également être connecté à une base de données de modèles 3D libres pour enrichir le nombre de dispositifs dont il dispose.

Esquisse supporte aussi l'intégration de fichiers d'interfaces graphiques qui peuvent être automatiquement intégrés aux écrans des appareils dans la scène avec une perspective correcte. Cependant, ces fichiers d'interfaces graphiques doivent encore être produits par l'utilisateur, en utilisant d'autres logiciels. Il existe un nombre croissant d'outils d'édition (tels que Figma⁷ ou Sketch⁸) qui peuvent être utilisés pour produire rapidement des fichiers d'interfaces graphiques en 2D qui seront intégrés dans l'exportation SVG par *Esquisse*. Ce genre d'outils pourrait d'ailleurs être directement intégré dans le processus créatif d'*Esquisse*, permettant de modifier directement les interfaces graphiques dans la scène 3D, et ainsi permettre aux utilisateurs de plus facilement faire correspondre la position ou le mouvement des objets 3D dans la scène avec les interfaces affichées sur les objets écran.

Enfin, comme vu dans notre taxonomie, la plupart des figures contiennent des méta-annotations. Ces annotations sont généralement des formes géométriques simples ou des couches de texte en superposition des figures. *Esquisse* ne propose pas actuellement de moyen d'ajouter ces annotations et l'utilisateur doit toujours ouvrir le fichier SVG produit dans un logiciel d'édition graphique pour les ajouter ensuite. L'ajout d'un système de méta-annotation à *Esquisse* pourrait permettre aux utilisateurs d'avoir une visualisation complète de leur figure. Ce système pourrait s'appuyer par exemple sur le système d'ancrage où la méta-annotation serait une forme ou un texte qui serait positionné dans la scène 3D et serait lié à un objet 3D spécifique.

Il est à noter qu'un contrat d'ingénieur d'un an sera financé par Inria à partir de janvier 2021 pour développer une version autonome et en ligne d'*Esquisse* en javascript via la librairie Three.js. Cette nouvelle version d'*Esquisse*, libérée des contraintes de l'environnement proposé par Blender, va pouvoir implémenter toutes ces fonctionnalités.

7. <https://www.figma.com/>

8. <https://www.sketchapp.com/>

Pistes d'amélioration des sphères virtuelles

La visualisation des mouvements de souris effectués par les participants lors de l'étude que nous avons menée sur l'influence de la taille de la sphère de contrôle lors d'une tâche de rotation laisse penser que les participants ont employé différentes stratégies en fonction des conditions de rotation (taille de la sphère, amplitude et tolérance de la tâche). Une future étude pourrait s'intéresser à l'analyse approfondie du comportement des utilisateurs lors des rotations. Notamment, identifier s'il existe des stratégies spécifiques employées par des participants dans des conditions particulières, et, si oui, il serait alors intéressant de trouver des métriques permettant de distinguer les participants, mesurer l'efficacité ou non de ces stratégies, et d'expliquer pourquoi elles sont employées.

Un des objectifs de ce travail est de proposer une nouvelle technique de rotation. Nous avons vu dans ce chapitre que la taille de sphère virtuelle de contrôle définissait à la fois la vitesse de rotation et la surface d'interaction disponible pour manipuler l'objet. Nous pensons que c'est une mauvaise façon de faire, et qu'il faudrait dissocier ces deux caractéristiques. Notamment, alors qu'il paraît évident que la taille de la sphère d'interaction doit définir la zone d'interaction, nous supposons que la vitesse de rotation doit dépendre de la vitesse du pointeur de souris. En utilisant les stratégies identifiées dans une seconde étude, il serait possible de définir des fonctions de transfert spécifiques permettant d'associer de manière linéaire ou non une vitesse de rotation à une vitesse du pointeur de souris donnée. Cette technique devrait permettre aux utilisateurs de pouvoir avoir le même comportement de rotation qu'importe la position du curseur sur la sphère. Une troisième future étude permettrait ainsi de savoir si cette technique permet d'améliorer les performances des utilisateurs lors de tâches de rotation d'objets 3D.

Vers un système idéal de création de figures ?

Les travaux effectués dans cette thèse peuvent être rassemblés pour imaginer, conceptualiser un système idéal pour créer des figures à base de tracés illustrant un scénario interactif. Dans ce système idéal, l'utilisateur pourrait prendre une photographie du monde réel qui met en scène ce qu'il veut représenter, ou, grâce à notre taxonomie, chercherait dans une bibliothèque de figures de la littérature celles qui se rapprochent le plus de ce qu'il souhaite en sélectionnant les éléments qui constituent son scénario. Pour éviter l'agencement d'une scène 3D, des techniques de reconstruction pourraient être alors utilisées pour générer une scène 3D à partir d'une photographie. Il est aussi possible d'imaginer que la phase de reconstruction de la scène soit semi-automatique pour optimiser les calculs en demandant par exemple à l'utilisateur d'identifier les objets présents sur la photographie, définir des points de fuites de l'image afin de générer une scène sans artefacts (mauvaise perspective, mauvaise position des objets, etc.).

La principe de base de la création de figures se baserait donc toujours sur l'agencement

3D, mais celle-ci serait masquée de l'utilisateur. En effet, l'utilisateur ne verrait que le rendu 2D de la scène, soit sa figure finale. D'une manière similaire au système proposé par Kalnins *et al.* [74], qui permet une stylisation en temps temps réel d'une scène 3D en n'affichant que son rendu, l'interaction avec les objets 3D serait toujours possible, non plus dans la scène 3D où les utilisateurs ont des difficultés à manipuler les objets, mais dans l'espace en deux dimensions du rendu, via diverses techniques d'interaction en 2D. Par exemple, un clic et un déplacement du curseur de la souris pourraient déplacer et orienter les objets grâce à une technique adaptée de sphère virtuelle, manipuler les armatures via une adaptation du système d'ancrage proposé par *Esquisse* en 2D ou simplement par démonstration via le LeapMotion, la Microsoft Kinect [35] ou TensorFlow [157, 114].

Ce nouvel outil aurait aussi un système intégré permettant d'expérimenter des stratégies d'illustration. L'utilisateur indiquerait les objectifs de sa figure et le système proposerait des variations de sa figure utilisant ces stratégies. Par exemple, l'utilisateur indiquerait qu'il souhaite mettre en mouvement la main d'un avatar de sa figure d'un point A à un point B, et le système lui proposerait une prévisualisation des variantes de sa figure avec divers effets tels qu'une décomposition de sa figure en plusieurs vignettes pour former un storyboard ou le mouvement de la main avec des flèches, un effet stroboscopique ou un effet de flou. Il pourrait aussi indiquer qu'il souhaite mettre en avant tels objets dans sa figure et le système lui montrerait alors d'autres variations de sa figure avec des techniques de regroupement telles que l'utilisation de couleurs spécifiques. De cette manière, l'utilisateur pourrait explorer diverses stratégies d'illustration directement avec sa figure et choisir celles qui lui paraissent les meilleures.

En travaillant directement en deux dimensions avec le rendu de la figure, ce processus créatif idéal pourrait permettre l'annotation directe de la figure tel qu'ajouter des légendes ou des annotations textuelles liées à des éléments qui composent la figure. Nous pouvons imaginer aussi que ces annotations seraient disposées automatiquement dans la figure via un solveur de contrainte pour ne pas masquer les objets représentés et éviter les intersections entre les flèches ou les lignes. Le système pourrait aussi mettre à jour automatiquement la position des annotations en fonction du point de vue de la caméra et de la position des objets.

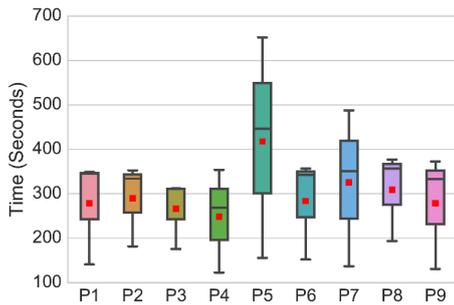
Bien qu'*Esquisse* offre aux utilisateurs la possibilité de personnaliser les figures à base de tracés rendues en modifiant la couleur, l'épaisseur et le style des contours dans le fichier SVG, un système idéal pourrait proposer des méthodes de stylisation automatique pour les graphiques vectoriels [41, 75, 50] pour produire des figures à base de tracés avec différents styles, tels que des effets de « coups de crayon » ou des effets d'éclairage (ajouter le rendu des ombres par exemple) ce qui permettrait alors aux utilisateurs de personnaliser leurs figures. Les travaux présentés dans cette thèse fournissent ainsi un premier cadre théorique et pratique pour informer les futures recherches et outils portant sur la production de figures illustrant des scénarios interactifs.

Annexes

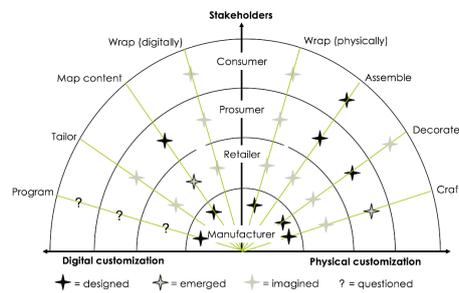
A

Exemples d'illustrations extraites des articles de recherche

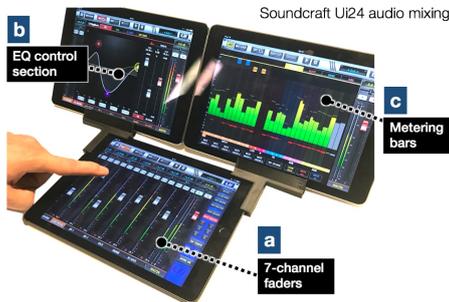
Pour réaliser notre taxonomie et constituer notre jeu de données (*cf.* chapitre 2), nous avons automatiquement extrait 7243 figures des articles de recherche de quatre conférences en IHM, en utilisant l'outil *pdffigure2*¹. Ces figures ont été triées manuellement et classées dans 8 catégories : graphiques, diagrammes, systèmes et environnements, tableaux, textes, interfaces, scénarios interactifs et artefacts. La figure A.1 ci-après montre des exemples de figures pour chacune des catégories.



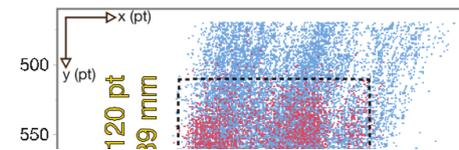
(a) *Graphiques* : Boîte à moustache qui montre le temps de réalisation d'un tâche pour chaque participant [72, Fig. 3]



(b) *Diagrammes* : Carte de personnalisation du calendrier de l'Avent [13, Fig. 10]

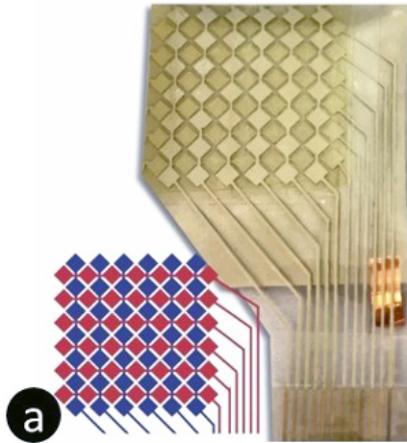


(c) *Scénarios interactifs* : Interaction d'un utilisateur avec un système multi-écrans [102, Fig. 16]



(d) *Artefacts* : Partie non complète d'un graphique [36, Fig. 9]

1. <https://github.com/allenai/pdffigures2>



(e) *Systèmes et Environnements* : Photographie d'un prototype électronique [117, Fig. 4]

```
void foo(int n) {
    int sum,i;
    assume(1 <= n);
    sum = 0;
    i = 1;
    while (i <= n)
        // invariant: a condition that holds here
        // at each iteration
        {
            sum = sum + i;
            i = i + 1;
        }
    assert(2*sum == n*(n+1));
}
```

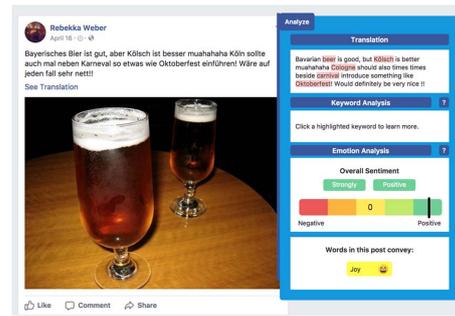
(g) *Textes* : Pseudo-code d'une fonction. [21, Fig. 1]

Question	Response (1 to 5)				
	1	2	3	4	5
1. How easy is it to rotate the Dial compared to panning with the mouse?	0	0	2	8	5
2. How easy is it to learn SteeringWheel?	1	2	0	4	8
3. How important is it to view the segment boundaries?	0	0	4	3	7
4. How useful is audio feedback?	0	1	6	6	2
5. How useful is haptic feedback?	0	0	0	6	9
6. How easy is it to perform gestures?	1	1	6	5	2
7. How easy is it to use Dial and Mouse together with both hands?	1	2	2	3	7
8. How easy is it to fill forms with SteeringWheel?	0	0	1	7	7
9. How easy is it to customize the interface with the Dashboard?	0	1	2	7	5
10. How noticeable were the effects of locality preservation?	0	0	1	9	5

Color Legends (5 color bins):

0-3	4-6	7-9	10-12	13-15
-----	-----	-----	-------	-------

(f) *Tableaux* : Table de réponses de participants aux questions d'une expérimentation [18, Tab. 3]



(h) *Interfaces* : Interface d'un logiciel de traduction de texte [96, Fig. 1]

Figure A.1 – Exemples de figures pour chacune des huit catégories d'illustration identifiées lors de la phase de tri des 7243 figures extraites des articles de recherche.

Composition				
Point de vue	1 ^{ère} personne		La scène est vue à travers les yeux de l'utilisateur.	
	3 ^{ème} personne		La scène est vue en face ou sur le côté de l'utilisateur.	
	Par-dessus l'épaule		La scène est vue par-dessus l'épaule de l'utilisateur.	
	Vue du dessus		La scène est vue du dessus.	
	Interface utilisateur seule		La figure montre uniquement une interface utilisateur. Des doigts ou des mains peuvent être représentés.	
Disposition	Nombre de vignettes	Une vignette	La figure ne comporte qu'une vignette.	
		Plusieurs vignettes	La figure comporte plusieurs vignettes.	
	Sous-vignettage	Image dans l'image		Une vignette est incluse dans une autre vignette avec un point de vue différent.
		Agrandissement		Une vignette est incluse dans une autre vignette et représente une portion détaillée de la vignette parente.
		Incrustation d'interface		Une interface utilisateur est intégrée dans la figure (moyennant une perspective).
		Interface superposée		Une interface utilisateur est superposée dans la figure, souvent avec transparence.
		Juxtaposition		Une vignette parente est composée de plusieurs sous-vignettes juxtaposées.

Table B.1 – Hiérarchie des codes pour la catégorie *Composition*.

Utilisateurs			
Nombre	Un utilisateur		Un seul utilisateur interagit avec le système.
	Utilisateurs multiples		Plusieurs utilisateurs interagissent avec le système.
Partie du corps	Membre spécifique	Main	Seules les mains de l'utilisateur sont visibles.
		Doigt	Seuls les doigts de l'utilisateur sont visibles.
		Tête	Seule la tête de l'utilisateur est visible.
		Pied	Seuls les pieds de l'utilisateur sont visibles.
	Bas du corps		Le bas du corps de l'utilisateur est visible.
	Haut du corps		Le haut du corps de l'utilisateur est visible.
Corps entier		Le corps entier de l'utilisateur est visible.	

Table B.2 – Hiérarchie des codes pour la catégorie *Utilisateurs*.

Techniques visuelles				
Mise en évidence	Région	Zone colorée		Coloration d'une portion de la figure pour mettre en avant une zone spécifique.
	Élément	Flèche		Des flèches pointent vers un objet pour le mettre en avant.
		Élément coloré		Une couleur spécifique est utilisée pour un objet.
		Contour	Contour de l'objet	Le contour d'un objet est mis en avant.
	Contour géométrique		Un objet est entouré par un rectangle, un cercle, etc.	
Arrière-plan	Grisé/Flouté		L'arrière-plan de la figure est grisé ou flouté.	
	Supprimé		L'arrière-plan de la figure est supprimé.	
Anonymisation	Floutage		Des portions de la figure sont floutées pour masquer le contenu.	
	Remplissage		Des portions de la figure sont colorées pour masquer le contenu.	
Grouper et lier	Couleurs		Des couleurs identiques sont utilisées pour grouper des objets entre eux.	
	Indicateurs		Des indicateurs de type chiffre ou lettre sont utilisés pour relier des objets entre eux.	
	Flèches		Des lignes ou des flèches sont utilisées pour relier des objets entre eux.	
	Annotations textuelles		Du texte (<i>p.ex.</i> étiquette, définition) est associé à un élément via une ligne ou des flèches.	
Dynamique	Effet dynamique	Effet stroboscopique		Un effet stroboscopique est utilisé pour montrer les différentes étapes d'un mouvement.
		Onomatopées		Effets visuels (<i>p.ex.</i> vagues, ondes) ou textuels permettant d'illustrer le mouvement.
		Flou de mouvement		Un effet de flou directionnel est utilisé pour indiquer un mouvement.
	Ligne et flèche	Trajectoire		Des flèches ou des lignes sont utilisées pour illustrer la trajectoire ou le chemin d'un mouvement.
		Direction		Des flèches ou des lignes sont utilisées pour montrer la direction d'un mouvement.
		Transition		Des flèches ou des lignes sont utilisées pour montrer la transition entre deux états d'un système.
		Projection		Des flèches ou des lignes sont utilisées pour montrer la projection d'un objet.
	Point de contact		Des formes géométriques, souvent des cercles, sont utilisées pour représenter la zone de contact entre deux éléments.	
Structure	Indicateur	Chiffre		Des chiffres sont utilisés pour ordonner une séquence ou diverses parties de la figure (souvent dans un coin).
		Lettre		Des lettres sont utilisées pour ordonner une séquence ou diverses parties de la figure (souvent dans un coin).
		Sous-titre		Des sous-titres sont utilisés pour annoter les vignettes.
Mesure	Flèche		Des flèches sont utilisées pour indiquer une mesure (<i>p.ex.</i> un angle).	
	Indicateur textuel		Du texte est utilisé pour indiquer la valeur d'une mesure.	

Table B.3 – Hiérarchie des codes pour la catégorie *Techniques visuelles*.

Scénario interactif		
Objectif	Espace de conception	La figure illustre l'ensemble des fonctionnalités, des commandes possibles ou les cas d'usage d'un système interactif.
	Séquence d'interaction	La figure illustre une séquence d'interaction (ou storyboard), <i>c.-à-d.</i> une séquence d'actions qui doivent être exécutées dans un ordre spécifique.
	Système interactif	La figure illustre un système interactif et se concentre sur le dispositif mis en place (<i>p.ex.</i> les capteurs).
Temporalité	Évolutive	La temporalité évolue au sein de la figure, <i>c.-à-d.</i> nous pouvons y voir une transition, une progression entre les actions d'un utilisateur et/ou les réponses du système.
	Fixe	La temporalité est fixe au sein de la figure, <i>c.-à-d.</i> nous observons un utilisateur et/ou un système à un moment spécifique sans aucune information sur les états précédents ou suivants.
Activité	Production	Production : programmation, entrée de texte, etc.
	Divertissement	Vidéo, musique, jeux vidéo, etc.
	Médical	Opération chirurgicale, etc.
	Conduite	Conduite automobile, interaction dans une voiture, etc.
	Création 2D/3D	Dessin 2D, modélisation 3D, mise en scène 3D, etc.
	Fabrication	Prototypage physique, conception, etc.
	Communication	Communication, coopération, échanges entre des utilisateurs, etc.
Manipulation de données	Manipulation de cartes, de graphiques, etc.	

Table B.4 – Hiérarchie des codes pour la catégorie *Scénario interactif*.

Caractéristiques visuelles				
Type	Dessin	Réalisme	Simpliste	La figure contient des éléments dessinés de manière simpliste.
			Réaliste	La figure contient des éléments dessinés de manière réaliste.
	UI	Dessin		La figure contient une interface utilisateur dessinée à la main ou via un logiciel.
		Capture ou rendu		La figure contient une interface utilisateur générée via <i>p.ex.</i> un rendu 3D ou une capture d'écran.
	Photographie			La figure contient des éléments tirés d'une photographie.
	Visualisation de données			La figure contient un graphique, un diagramme, etc.
	Clipart/icône			La figure contient des cliparts, des icônes.
	Texte			La figure contient du texte.
Coloration	Transparence			La transparence est utilisée dans la figure.
	Teinte	Couleurs		La figure utilise plusieurs couleurs.
		Niveaux de gris		La figure utilise des niveaux de gris.
		Monochrome		La figure utilise une seule couleur (<i>p.ex.</i> noir et blanc).
Style de ligne	Pointillée			La figure présente des lignes pointillées.
	Épaisseur			La figure présente des lignes avec des épaisseurs différentes.

Table B.5 – Hiérarchie des codes pour la catégorie *Caractéristiques visuelles*.

Système interactif			
Action	Discrète	Pointage	L'utilisateur pointe vers une cible.
		Frappe de touches	L'utilisateur presse une touche ou un bouton physique.
		Geste symbolique	L'utilisateur exécute un geste prédéfini.
	Continue	Translation	L'utilisateur déplace un objet.
		Rotation	L'utilisateur oriente un objet.
		Mise à l'échelle	L'utilisateur change la taille d'un objet.
		Défilement	L'utilisateur défile un contenu.
		Déformation	L'utilisateur déforme un objet.
	Écriture/Dessin	L'utilisateur dessine ou écrit.	
Type de l'interaction	Interaction tactile		L'interaction se déroule sur une surface tactile.
	Interaction dans l'air		L'interaction se déroule dans l'air.
	Interaction à distance		L'interaction se déroule à distance.
	Interaction sur le corps		L'interaction se déroule sur le corps de l'utilisateur.
	Interaction sur ordinateur		L'interaction se déroule avec un ordinateur de bureau en utilisant le couple clavier/souris/pavé tactile.
	Interaction tangible		L'interaction avec le système se fait via des objets tangibles.
	Interaction via des contrôleurs		L'interaction avec le système se fait via des contrôleurs (télécommandes, manettes, etc.).
	Interaction avec crayons		L'interaction avec le système se fait via un crayon/styler.
	Interaction via le regard		L'interaction avec le système se fait avec les yeux de l'utilisateur.
Canal d'entrée	Capture	Capture externe	L'utilisateur est suivi par le système avec des capteurs externes (<i>p.ex.</i> Microsoft Kinect).
		Capture interne	L'utilisateur est suivi par le système avec des capteurs internes ou sur lui (<i>p.ex.</i> une montre connectée).
	Dispositif	Dispositif mobile	L'utilisateur interagit avec le système via un dispositif mobile (smartphone, tablette, montre connectée, etc.).
		Grande surface	L'utilisateur interagit avec le système via une grande surface (mur d'écrans, télévision, table interactive, etc.).
		Périphérique de bureau	L'utilisateur interagit avec le système via un périphérique d'ordinateur de bureau (clavier, pavé tactile, souris, etc.).
		Périphérique portable	L'utilisateur interagit avec le système via un périphérique d'ordinateur portable (clavier intégré, pavé tactile intégré, etc.).
		Contrôleur	L'utilisateur interagit avec le système via un contrôleur (télécommande, manette, etc.).
		Objet tangible	L'utilisateur interagit avec un objet tangible.
		Dispositifs de RA	L'utilisateur interagit avec des dispositifs de RA.
	Dispositifs de RV	L'utilisateur interagit avec des dispositifs de RV.	
	Son	Entrée vocale	L'utilisateur interagit avec le système via un microphone.
Modalité de sortie	Haptique		Le système fournit un retour haptique à l'utilisateur.
	Visuelle	Écran large	Le retour visuel se fait via un écran large (télévision, mur d'écrans, table interactive, etc.).
		Écran d'ordinateur	Le retour visuel se fait via un écran d'ordinateur (ordinateur portable, ordinateur de bureau).
		Écran mobile	Le retour visuel se fait via un dispositif mobile (smartphone, tablette, montre connectée, écran de voiture, etc.).
		Écran projeté	Le retour visuel se fait via un écran projeté.
		HMD	Casque de RA ou de RV.
		Lumière	Le retour visuel se fait via des lampes, des LEDs, des ampoules, etc.
	Audible	Son	Le système émet un retour sonore spécifique.
Sortie vocale		Le système parle à l'utilisateur dans sa langue.	
Situation	Bureau		Sur une table, un bureau, etc.
	Transport public ou privé		Voiture, avion, vélo, autobus, etc.
	Extérieur		Parc, rue, jardin, etc.
	Intérieur privé		À la maison, au bureau, etc.
	Intérieur public		Centre commercial, gare, musée, salle de sport, etc.

Table B.6 – Hiérarchie des codes pour la catégorie *Système interactif*.

Liste des acronymes

- ACP** Analyse en Composantes Principales. 70
- CHI** ACM Conference on Human Factors in Computing Systems. 9, 33
- CSCW** ACM Conference on Computer Supported Cooperative Work. 9, 33
- HMD** Head Mounted Display. 40, 47, 132
- IHM** Interaction Homme-Machine. 8, 10–12, 19, 24, 29, 31–34, 36, 37, 65, 71, 72, 74, 76, 80, 95, 96, 101, 120, 122, 127
- RA** Réalité Augmentée. 9, 40, 47, 49, 65, 132
- RV** Réalité Virtuelle. 9, 40, 47, 49, 54, 55, 70, 132
- Ubicomp** ACM International Joint Conference on Pervasive and Ubiquitous Computing. 9, 33
- UI** Interface Utilisateur. 41
- UIST** ACM Symposium on User Interface Software and Technology. 9, 33, 74

- Clipart** Anglicisme. Collection d'images numériques ou de dessins prêts à être insérés dans des documents multimédias. 27, 41, 78
- Débrayage** Le débrayage (an anglais, *clutch*) est un mécanisme qui permet d'augmenter l'étendue d'utilisation d'un dispositif [60] (*p.ex.* soulever et repositionner la souris sur la table, ou le doigt sur une surface tactile). 104, 107, 118
- Depth buffer** Le depth buffer (anglais) ou tampon de profondeur est un tampon utilisé dans les rendus 3D pour stocker la profondeur des pixels des objets rendus et savoir quelle couleur rendre pour chacun des pixels. 81
- Hystérésis** Lors d'une rotation d'un objet via une sphère virtuelle, le phénomène d'hystérésis apparaît si le déplacement du curseur de souris d'un point A à un point B de la sphère en utilisant des chemins différents mène à des orientations de l'objet 3D différentes.. 106
- Rotoscopie** La rotoscopie est une technique cinématographique qui consiste à relever image par image les contours d'une figure filmée en prise de vue réelle pour en transcrire la forme et les actions dans un film d'animation. 26
- Storyboard** Un storyboard (anglais) ou scénarimage est une figure comportant plusieurs vignettes successives relatant un enchaînement d'actions entre l'utilisateur et un système interactif. 16, 19–21, 131
- SVG** De l'anglais Scalable Vector Graphics, un fichier SVG est un document XML où les formes géométriques dessinées sont décrites à l'aide de propriétés : coordonnées des points, épaisseur des traits, couleurs, etc. 26, 79, 81, 89, 90, 93–96, 99–101, 123, 125
- Trackball** La trackball (anglais) ou sphère virtuelle de contrôle est une technique de rotation d'un objet 3D permettant de manipuler l'objet suivant les trois axes de rotation à partir des deux degrés de liberté du curseur de souris. 13, 104
- Widget** En informatique, un widget est un composant, un élément visuel d'une interface graphique (un bouton, une barre de défilement, etc). Ce composant peut être interactif ou non. 66, 67, 103, 104

Bibliographie

- [1] 2015. *UIST '15 : Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA.
- [2] 2016. *UIST '16 : Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, New York, NY, USA.
- [3] 2017. *UIST '17 : Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA.
- [4] Merwan Achibet, Géry Casiez, Anatole Lécuyer, and Maud Marchal. 2015. THING : Introducing a Tablet-based Interaction Technique for Controlling 3D Hand Models. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 317–326. DOI : <http://dx.doi.org/10.1145/2702123.2702158>
- [5] Maneesh Agrawala, Doantam Phan, Julie Heiser, John Haymaker, Jeff Klingner, Pat Hanrahan, and Barbara Tversky. 2003. Designing Effective Step-by-Step Assembly Instructions. In *ACM SIGGRAPH 2003 Papers (SIGGRAPH '03)*. Association for Computing Machinery, New York, NY, USA, 828–837. DOI : <http://dx.doi.org/10.1145/1201775.882352>
- [6] Emre Aksan, Fabrizio Pece, and Otmar Hilliges. 2018. DeepWriting. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173779>
- [7] Fouad Alallah, Ali Neshati, Nima Sheibani, Yumiko Sakamoto, Andrea Bunt, Pourang Irani, and Khalad Hasan. 2018. Crowdsourcing vs Laboratory-Style Social Acceptability Studies?. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173884>
- [8] Arthur Appel. 1967. The Notion of Quantitative Invisibility and the Machine Rendering of Solids. In *Proceedings of the 1967 22Nd National Conference (ACM '67)*. ACM, New York, NY, USA, 387–393. DOI : <http://dx.doi.org/10.1145/800196.806007>
- [9] Apple. 2020. Developer Documentation - Visualizing and Interacting with a Reconstructed Scene. (2020). Retrieved Decembre 3rd, 2020 from https://developer.apple.com/documentation/arkit/world_tracking/visualizing_and_interacting_with_a_reconstructed_scene

- [10] Ragnar Bade, Felix Ritter, and Bernhard Preim. 2005. Usability Comparison of Mouse-Based Interaction Techniques for Predictable 3d Rotation, Vol. 3638. 138–150. DOI : http://dx.doi.org/10.1007/11536482_12
- [11] G. Bell. Bell’s Trackball. (? ? ? ?). http://members.tripod.com/professor_tom/index.html, 1988
- [12] Mary Bellis. 2019. The Rules for Patents Drawings. (2019). Retrieved May 25th, 2020 from <https://www.thoughtco.com/rules-for-patent-drawings-1992228>
- [13] Steve Benford, Boriana Koleva, William Westwood Preston, Alice Angus, Emily-Clare Thorn, and Kevin Glover. 2018. Customizing Hybrid Products. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173604>
- [14] Joanna Bergstrom-Lehtovirta, Kasper Hornbæk, and Sebastian Boring. 2018. It's a Wrap. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3174138>
- [15] Jacques Bertin. 1983. *Semiology of graphics ; diagrams networks maps*. Technical Report.
- [16] James C. Bezdek, Robert Ehrlich, and William Full. 1984. FCM : The fuzzy c-means clustering algorithm. *Computers and Geosciences* 10, 2 (1984), 191 – 203. DOI : [http://dx.doi.org/https://doi.org/10.1016/0098-3004\(84\)90020-7](http://dx.doi.org/https://doi.org/10.1016/0098-3004(84)90020-7)
- [17] Eric Allan Bier. 1987. Skitters and Jacks : Interactive 3D Positioning Tools. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics (I3D '86)*. Association for Computing Machinery, New York, NY, USA, 183–196. DOI : <http://dx.doi.org/10.1145/319120.319135>
- [18] Syed Masum Billah, Vikas Ashok, Donald E. Porter, and I.V. Ramakrishnan. 2018. SteeringWheel. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173594>
- [19] Richard A. Bolt. 1980. ”Put-that-there” : Voice and Gesture at the Graphics Interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '80)*. ACM, New York, NY, USA, 262–270. DOI : <http://dx.doi.org/10.1145/800250.807503>
- [20] M. Bostock, V. Ogievetsky, and J. Heer. 2011. D³ Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2301–2309.
- [21] Dimitar Bounov, Anthony DeRossi, Massimiliano Menarini, William G. Griswold, and Sorin Lerner. 2018. Inferring Loop Invariants through Gamification. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173805>

- [22] Michael Brock, Aaron Quigley, and Per Ola Kristensson. 2018. Change Blindness in Proximity-Aware Mobile Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–7. DOI: <http://dx.doi.org/10.1145/3173574.3173617>
- [23] Frederik Brudy, Joshua Kevin Budiman, Steven Houben, and Nicolai Marquardt. 2018. Investigating the Role of an Overview Device in Multi-Device Collaboration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI: <http://dx.doi.org/10.1145/3173574.3173874>
- [24] Simon Butscher, Sebastian Hubenschmid, Jens Müller, Johannes Fuchs, and Harald Reiterer. 2018. Clusters, Trends, and Outliers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI: <http://dx.doi.org/10.1145/3173574.3173664>
- [25] David C. Buuck. 2013. Providing tactile output and interaction. (Nov. 3 2013). <https://patentimages.storage.googleapis.com/af/b0/b3/241c27d8e77d55/US8593420.pdf> US Patent 8,593,420 B1.
- [26] Bill Buxton. 2007. *Sketching User Experiences : Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [27] John Canny. 1987. A computational approach to edge detection. In *Readings in Computer Vision*. Elsevier, 184–203.
- [28] Sheelagh Carpendale. 2003. *Considering Visual Variables as a Basis for Information Visualisation*. Technical Report. University of Calgary. DOI: <http://dx.doi.org/10.11575/PRISM/30495>
- [29] Hong-Yu Chang, Wen-Jie Tseng, Chia-En Tsai, Hsin-Yu Chen, Roshan Lalintha Peiris, and Liwei Chan. 2018. FacePush. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI: <http://dx.doi.org/10.1145/3242587.3242588>
- [30] Youli Chang, Sehi L'Yi, Kyle Koh, and Jinwook Seo. 2015. Understanding Users' Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1499–1508. DOI: <http://dx.doi.org/10.1145/2702123.2702425>
- [31] Michael Chen, S. Joy Mountford, and Abigail Sellen. A Study in Interactive 3-D Rotation Using 2-D Control Devices. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques (1988) (SIGGRAPH '88)*. ACM, 121–129. DOI: <http://dx.doi.org/10.1145/54852.378497>
- [32] Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. 2013. 3-sweep : Extracting editable objects from a single photo. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 195.

- [33] Xiang 'Anthony' Chen, Ye Tao, Guanyun Wang, Runchang Kang, Tovi Grossman, Stelian Coros, and Scott E. Hudson. 2018. Forte. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI: <http://dx.doi.org/10.1145/3173574.3174070>
- [34] Pei-Yu Chi. 2016. *Designing Video-Based Interactive Instructions*. Ph.D. Dissertation. University of California, Berkeley.
- [35] Pei-Yu (Peggy) Chi, Daniel Vogel, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2016. Authoring Illustrations of Human Movements by Iterative Physical Demonstration. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 809–820. DOI: <http://dx.doi.org/10.1145/2984511.2984559>
- [36] Christian Corsten, Simon Voelker, Andreas Link, and Jan Borchers. 2018. Use the Force Picker, Luke. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI: <http://dx.doi.org/10.1145/3173574.3174235>
- [37] James E Cutting. 2002. Representing Motion in a Static Image : Constraints and Parallels in Art, Science, and Popular Culture. *Perception* 31, 10 (2002), 1165–1193. DOI: <http://dx.doi.org/10.1068/p3318> PMID : 12430945.
- [38] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. 2003. Suggestive Contours for Conveying Shape. In *ACM SIGGRAPH 2003 Papers (SIGGRAPH '03)*. ACM, New York, NY, USA, 848–855. DOI: <http://dx.doi.org/10.1145/1201775.882354>
- [39] Alix Ducros, Clemens N. Klokmoose, and Aurélien Tabard. 2019. Situated Sketching and Enactment for Pervasive Displays. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces (ISS '19)*. Association for Computing Machinery, New York, NY, USA, 217–228. DOI: <http://dx.doi.org/10.1145/3343055.3359702>
- [40] J. C. Dunn. 1973. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics* 3, 3 (1973), 32–57. DOI: <http://dx.doi.org/10.1080/01969727308546046>
- [41] Elmar Eisemann, Holger Winnemöller, John C Hart, and David Salesin. 2008. Stylized vector art from 3D models with region support. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1199–1207.
- [42] Elsevier. 2020. Graphical abstracts. (2020). Retrieved June 9th, 2020 from <https://www.elsevier.com/authors/journal-authors/graphical-abstract>
- [43] Jean-Dominique Favreau, Florent Lafarge, and Adrien Bousseau. 2017. Photo2clipart : Image Abstraction and Vectorization Using Layered Linear

- Gradients. *ACM Trans. Graph.* 36, 6, Article 180 (Nov. 2017), 11 pages. DOI : <http://dx.doi.org/10.1145/3130800.3130888>
- [44] Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76, 5 (1971), 378–382. DOI : <http://dx.doi.org/10.1037/h0031619> Place : US Publisher : American Psychological Association.
- [45] Malcolm Fleming. 1967. Classification and Analysis of Instructional Illustrations. *AV Communication Review* 15, 3 (1967), 246–258. <http://www.jstor.org/stable/30217383>
- [46] Euan Freeman, Julie Williamson, Sriram Subramanian, and Stephen Brewster. 2018. Point-and-Shake. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173592>
- [47] Emmanouil Giannidakis, Gilles Bailly, Sylvain Malacria, and Fanny Chevalier. 2017. IconHK : Using Toolbar Button Icons to Communicate Keyboard Shortcuts. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 4715–4726. DOI : <http://dx.doi.org/10.1145/3025453.3025595>
- [48] Alix Goguey, Géry Casiez, Daniel Vogel, and Carl Gutwin. 2018a. Characterizing Finger Pitch and Roll Orientation During Atomic Touch Actions. In *CHI 2018-ACM Conference on Human Factors in Computing Systems*. 1–12.
- [49] Alix Goguey, Sylvain Malacria, and Carl Gutwin. 2018b. Improving Discoverability and Expert Performance in Force-Sensitive Text Selection for Touch Devices with Mode Gauges. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3174051>
- [50] Stéphane Grabli, Emmanuel Turquin, Frédo Durand, and François X. Sillion. 2004. Programmable Style for NPR Line Drawing. In *Proceedings of the Fifteenth Eurographics Conference on Rendering Techniques (EGSR'04)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 33–44. DOI : <http://dx.doi.org/10.2312/EGWR/EGSR04/033-044>
- [51] Saul Greenberg, Sheelagh Carpendale, Nicolai Marquardt, and Bill Buxton. 2011. *Sketching User Experiences : The Workbook* (1st ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [52] Saul Greenberg, Sheelagh Carpendale, Nicolai Marquardt, and Bill Buxton. 2012. The Narrative Storyboard : Telling a Story about Use and Context over Time. *Interactions* 19, 1 (Jan. 2012), 64–69. DOI : <http://dx.doi.org/10.1145/2065327.2065340>
- [53] Tovi Grossman, Fanny Chevalier, and Rubaiat Habib Kazi. 2015. Your Paper is Dead ! Bringing Life to Research Articles with Animated Figures. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing*

- Systems (CHI EA '15)*. Association for Computing Machinery, New York, NY, USA, 461–475. DOI : <http://dx.doi.org/10.1145/2702613.2732501>
- [54] Jan Gugenheimer, Evgeny Stemasov, Harpreet Sareen, and Enrico Rukzio. 2018. Face-Display. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173628>
- [55] Jaehyun Han and Geehyuk Lee. 2015. Push-Push : A Drag-like Operation Overlapped with a Page Transition Operation on Touch Interfaces. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 313–322. DOI : <http://dx.doi.org/10.1145/2807442.2807457>
- [56] Y.. Hansen. 2000. *Visualization for Thinking, Planning, and Problem Solving*. 193–220 pages. <https://mediosavanzados.files.wordpress.com/2007/06/visualization-for-thinking.pdf>
- [57] S. Haroz, R. Kosara, and S. L. Franconeri. 2016. The Connected Scatterplot for Presenting Paired Time Series. *IEEE Transactions on Visualization and Computer Graphics* 22, 9 (2016), 2174–2186.
- [58] K. Henriksen, J. Sporning, and K. Hornbaek. Virtual trackballs revisited. 10, 2 (????), 206–216. DOI : <http://dx.doi.org/10.1109/TVCG.2004.1260772>
- [59] Aaron Hertzmann and Denis Zorin. 2000. Illustrating Smooth Surfaces. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 517–526. DOI : <http://dx.doi.org/10.1145/344779.345074>
- [60] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. 1994. Passive Real-World Interface Props for Neurosurgical Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*. Association for Computing Machinery, New York, NY, USA, 452–458. DOI : <http://dx.doi.org/10.1145/191666.191821>
- [61] Ken Hinckley, Gonzalo Ramos, Francois Guimbretiere, Patrick Baudisch, and Marc Smith. 2004. Stitching : Pen Gestures That Span Multiple Displays. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. ACM, New York, NY, USA, 23–31. DOI : <http://dx.doi.org/10.1145/989863.989866>
- [62] Ken Hinckley, Joe Tullio, Randy Pausch, Dennis Proffitt, and Neal Kassell. 1997. Usability Analysis of 3D Rotation Techniques. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST '97)*. Association for Computing Machinery, New York, NY, USA, 1–10. DOI : <http://dx.doi.org/10.1145/263407.263408>
- [63] Kasper Hornbæk and Antti Oulasvirta. 2017. What Is Interaction?. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5040–5052. DOI : <http://dx.doi.org/10.1145/3025453.3025765>

- [64] Donny Huang, Xiaoyi Zhang, T. Scott Saponas, James Fogarty, and Shyamnath Gollakota. 2015. Leveraging Dual-Observable Input for Fine-Grained Thumb Interaction Using Forearm EMG. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 523–528. DOI : <http://dx.doi.org/10.1145/2807442.2807506>
- [65] Jin Huang, Feng Tian, Xiangmin Fan, Xiaolong (Luke) Zhang, and Shumin Zhai. 2018. Understanding the Uncertainty in 1D Unidirectional Moving Target Selection. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. DOI : <http://dx.doi.org/10.1145/3173574.3173811>
- [66] Jessica Hullman and Benjamin Bach. 2018. Picturing Science : Design Patterns in Graphical Abstracts. In *International Conference on Theory and Application of Diagrams*. Springer.
- [67] Ke Huo, Yuanzhi Cao, Sang Ho Yoon, Zhuangying Xu, Guiming Chen, and Karthik Ramani. 2018a. Scenariot. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173793>
- [68] Ke Huo, Tianyi Wang, Luis Paredes, Ana M. Villanueva, Yuanzhi Cao, and Karthik Ramani. 2018b. SynchronizAR. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3242587.3242595>
- [69] Emmanuel Iarussi, David Bommès, and Adrien Bousseau. 2015. BendFields : Regularized Curvature Fields from Rough Concept Sketches. *ACM Transactions on Graphics* (2015). <http://www-sop.inria.fr/reves/Basilic/2015/IBB15>
- [70] Inés Jacob and Javier Oliver. 1996. Evaluation of Techniques for Specifying 3D Rotations with a 2D Input Device. In *Proceedings of the HCI'95 Conference on People and Computers X (HCI '95)*. Cambridge University Press, USA, 63–76.
- [71] Yunwoo Jeong, Han-Jong Kim, and Tek-Jin Nam. 2018. Mechanism Perfboard. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173985>
- [72] Hernisa Kacorri, Eshed Ohn-Bar, Kris M. Kitani, and Chieko Asakawa. 2018. Environmental Factors in Indoor Navigation Based on Real-World Trajectories of Blind Users. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173630>
- [73] Robert E. Kahn and John H. Cafarella. 1998. Unconstrained pointing interface for natural human interaction with a display-based computer system. (Aug. 3 1998). <https://patentimages.storage.googleapis.com/e0/a6/26/bb6514edf9149e/US5793361.pdf> US Patent 5,793,361.

- [74] 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. 2002. WYSIWYG NPR : Drawing Strokes Directly on 3D Models. *ACM Trans. Graph.* 21, 3 (July 2002), 755–762. DOI : <http://dx.doi.org/10.1145/566654.566648>
- [75] Kevin Karsch and John C. Hart. 2011. Snaxels on a Plane. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering (NPAR '11)*. ACM, New York, NY, USA, 35–42. DOI : <http://dx.doi.org/10.1145/2024676.2024683>
- [76] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George Fitzmaurice. 2014a. Kitty : Sketching Dynamic and Interactive Illustrations. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. Association for Computing Machinery, New York, NY, USA, 395–405. DOI : <http://dx.doi.org/10.1145/2642918.2647375>
- [77] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George Fitzmaurice. 2014b. Kitty : Sketching Dynamic and Interactive Illustrations. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 395–405. DOI : <http://dx.doi.org/10.1145/2642918.2647375>
- [78] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, Shengdong Zhao, and George Fitzmaurice. 2014c. Draco : Bringing Life to Illustrations with Kinetic Textures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 351–360. DOI : <http://dx.doi.org/10.1145/2556288.2556987>
- [79] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, Shengdong Zhao, and George Fitzmaurice. 2014d. DRACO : Sketching Animated Drawings with Kinetic Textures. In *ACM SIGGRAPH 2014 Posters (SIGGRAPH '14)*. ACM, New York, NY, USA, Article 5, 1 pages. DOI : <http://dx.doi.org/10.1145/2614217.2614221>
- [80] Mohamed Khamis, Ludwig Trotter, Ville Mäkelä, Emanuel von Zeszschwitz, Jens Le, Andreas Bulling, and Florian Alt. 2018. CueAuth. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (dec 2018), 1–22. DOI : <http://dx.doi.org/10.1145/3287052>
- [81] Natasha Kholgade, Tomas Simon, Alexei Efros, and Yaser Sheikh. 2014. 3D object manipulation in a single photograph using stock 3D models. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 127.
- [82] Rushil Khurana, Karan Ahuja, Zac Yu, Jennifer Mankoff, Chris Harrison, and Mayank Goel. 2018. GymCam. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (dec 2018), 1–17. DOI : <http://dx.doi.org/10.1145/3287063>
- [83] Han-Jong Kim, Chang Min Kim, and Tek-Jin Nam. 2018. SketchStudio : Experience Prototyping with 2.5-Dimensional Animated Design Scenarios. In *Proceedings of the*

- 2018 *Designing Interactive Systems Conference (DIS '18)*. ACM, New York, NY, USA, 831–843. DOI : <http://dx.doi.org/10.1145/3196709.3196736>
- [84] K. Koffka. *Principles of Gestalt psychology*. Harcourt, Brace. Pages : 720.
- [85] Thomas Kosch, Mariam Hassib, Pawel W. Woźniak, Daniel Buschek, and Florian Alt. 2018. Your Eyes Tell. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3174010>
- [86] Klaus Krippendorff. 1970. Estimating the Reliability, Systematic Error and Random Error of Interval Data. *Educational and Psychological Measurement* 30, 1 (1970), 61–70. DOI : <http://dx.doi.org/10.1177/001316447003000105>
- [87] Christian Lander, Sven Gehring, Antonio Krüger, Sebastian Boring, and Andreas Bulling. 2015. GazeProjector : Accurate Gaze Estimation and Seamless Gaze Interaction Across Multiple Displays. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 395–404. DOI : <http://dx.doi.org/10.1145/2807442.2807479>
- [88] J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33, 1 (1977), 159–174. <http://www.jstor.org/stable/2529310>
- [89] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2010. *Research Methods in Human-Computer Interaction*. Wiley Publishing.
- [90] Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. 2018. PalmTouch. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173934>
- [91] Joon Hyub Lee, Sang-Gyun An, Yongkwan Kim, and Seok-Hyung Bae. 2018. Projective Windows. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173792>
- [92] Changjian Li, Hao Pan, Adrien Bousseau, and Niloy J. Mitra. 2020. Sketch2CAD : Sequential CAD Modeling by Sketching in Context. *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2020)* 39, 6 (2020), 164 :1–164 :14. DOI : <http://dx.doi.org/https://doi.org/10.1145/3414685.3417807>
- [93] Wenbin Li, Fabio Viola, Jonathan Starck, Gabriel J. Brostow, and Neill D. F. Campbell. 2016. Roto++ : Accelerating Professional Rotoscoping Using Shape Manifolds. *ACM Trans. Graph.* 35, 4, Article 62 (July 2016), 15 pages. DOI : <http://dx.doi.org/10.1145/2897824.2925973>
- [94] Rong-Hao Liang, Meng-Ju Hsieh, Jheng-You Ke, Jr-Ling Guo, and Bing-Yu Chen. 2018. RFIMatch. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3242587.3242620>

- [95] Hyunchul Lim, Jungmin Chung, Changhoon Oh, SoHyun Park, Joonhwan Lee, and Bongwon Suh. 2018a. Touch+Finger. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3242587.3242651>
- [96] Hajin Lim, Dan Cosley, and Susan R. Fussell. 2018b. Beyond Translation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173791>
- [97] Jhe-Wei Lin, Chiuan Wang, Yi Yao Huang, Kuan-Ting Chou, Hsuan-Yu Chen, Wei-Luan Tseng, and Mike Y. Chen. 2015. BackHand : Sensing Hand Gestures via Back of the Hand. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 557–564. DOI : <http://dx.doi.org/10.1145/2807442.2807462>
- [98] Wanyu Liu, Olivier Rioul, Joanna McGrenere, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2018. BIGFile. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173959>
- [99] Joanne Lo, Cesar Torres, Isabel Yang, Jasper O’Leary, Danny Kaufman, Wilmot Li, Mira Dontcheva, and Eric Paulos. 2016. Aesthetic Electronics : Designing, Sketching, and Fabricating Circuits Through Digital Exploration. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 665–676. DOI : <http://dx.doi.org/10.1145/2984511.2984579>
- [100] Pedro Lopes, Sijing You, Alexandra Ion, and Patrick Baudisch. 2018. Adding Force Feedback to Mixed Reality Experiences and Games using Electrical Muscle Stimulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3174020>
- [101] I. Scott MacKenzie. 1992. Fitts’ Law as a Research and Design Tool in Human-Computer Interaction. *Hum.-Comput. Interact.* 7, 1 (March 1992), 91–139. DOI : http://dx.doi.org/10.1207/s15327051hci0701_3
- [102] Nicolai Marquardt, Frederik Brudy, Can Liu, Ben Bengler, and Christian Holz. 2018. SurfaceConstellations. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173928>
- [103] Nicolai Marquardt, Ken Hinckley, and Saul Greenberg. 2012. Cross-Device Interaction via Micro-Mobility and f-Formations. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. Association for Computing Machinery, New York, NY, USA, 13–22. DOI : <http://dx.doi.org/10.1145/2380116.2380121>

- [104] Patricia I. Cornelio Martinez, Emanuela Maggioni, Kasper Hornbæk, Marianna Obrist, and Sriram Subramanian. 2018. Beyond the Libet Clock. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3174115>
- [105] Sebastian Marwecki and Patrick Baudisch. 2018. Scenograph. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3242587.3242648>
- [106] Sebastian Marwecki, Maximilian Brehm, Lukas Wagner, Lung-Pan Cheng, Florian 'Floyd' Mueller, and Patrick Baudisch. 2018. VirtualSpace - Overloading Physical Space with Multiple Virtual Reality Users. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173815>
- [107] Damien Masson, Sylvain Malacria, Edward Lank, and Géry Casiez. 2020. Chameleon : Bringing Interactivity to Static Digital Documents. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. DOI : <http://dx.doi.org/10.1145/3313831.3376559>
- [108] Sven Mayer, Valentin Schwind, Robin Schweigert, and Niels Henze. 2018. The Effect of Offset Correction and Cursor on Mid-Air Pointing in Real and Virtual Environments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3174227>
- [109] Erin McAweeney, Haihua Zhang, and Michael Nebeling. 2018a. User-Driven Design Principles for Gesture Representations. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3174121>
- [110] Erin McAweeney, Haihua Zhang, and Michael Nebeling. 2018b. User-Driven Design Principles for Gesture Representations. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 547, 13 pages. DOI : <http://dx.doi.org/10.1145/3173574.3174121>
- [111] Scott McCloud. 1993. *Understanding Comics : The Invisible Art*. Tundra Publishing Ltd.
- [112] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and Interrater Reliability in Qualitative Research : Norms and Guidelines for CSCW and HCI Practice. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 72 (Nov. 2019), 23 pages. DOI : <http://dx.doi.org/10.1145/3359174>
- [113] Eric N. Mortensen and William A. Barrett. 1995. Intelligent Scissors for Image Composition. In *Proceedings of the 22Nd Annual Conference on Computer Graphics*

- and Interactive Techniques (SIGGRAPH '95)*. ACM, New York, NY, USA, 191–198. DOI : <http://dx.doi.org/10.1145/218380.218442>
- [114] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2017. Real-Time Hand Tracking Under Occlusion from an Egocentric RGB-D Sensor. 1284–1293. DOI : <http://dx.doi.org/10.1109/ICCVW.2017.82>
- [115] Mathieu Nancel, Daniel Vogel, Bruno De Araujo, Ricardo Jota, and Géry Casiez. 2016. Next-Point Prediction Metrics for Perceived Spatial Errors. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 271–285. DOI : <http://dx.doi.org/10.1145/2984511.2984590>
- [116] Eric Neufeld, Haruna Popoola, David Callele, and David Mould. 2003. Mixed Initiative Interactive Edge Detection.. In *Graphics Interface*. 177–184.
- [117] Aditya Shekhar Nittala, Anusha Withana, Narjes Pourjafarian, and J  rgen Steimle. 2018. Multi-Touch Skin. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173607>
- [118] Fabrizio Pece, Juan Jose Zarate, Velko Vechev, Nadine Besse, Olexandr Gudozhnik, Herbert Shea, and Otmar Hilliges. 2017. MagTics : Flexible and Thin Form Factor Magnetic Actuators for Dynamic and Wearable Haptic Feedback. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 143–154. DOI : <http://dx.doi.org/10.1145/3126594.3126609>
- [119] Eujin Pei, R.I. Campbell, and Mark Evans. 2011. A Taxonomic Classification of Visual Design Representations Used by Industrial Designers and Engineering Designers. *The Design Journal* 14 (2011), 64–91. DOI : <http://dx.doi.org/10.2752/175630610X12877385838803>
- [120] David Petrou. 2012. Head-mounted display that displays a visual representation of physical interaction with an input interface located outside of the field of view. (July 3 2012). <https://patentimages.storage.googleapis.com/5a/23/22/d3db8fd8048482/US8217856.pdf> US Patent 8,217,856 B1.
- [121] Henning Pohl and Kasper Hornb  k. 2018. ElectricItch. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3242587.3242647>
- [122] Ivan Poupyrev and Gaetano Roberto Aiello. 2015. Radar-based gesture-recognition through a wearable device. (Dec. 3 2015). <https://patentimages.storage.googleapis.com/77/69/61/d96948a8b469ba/US20150346820A1.pdf> US Patent 2015/0346820A1.
- [123] Ivan Poupyrev and Shigeaki Maruyama. 2011. Apparatus and method for touch screen interaction based on tactile feedback and pressure measurement. (May 3 2011).

- <https://patentimages.storage.googleapis.com/b4/7e/9f/37227d90b5fa6c/US7952566B2.pdf> US Patent 7,952,566 B2.
- [124] Ivan Poupyrev, Suzanne Weghorst, and Sidney Fels. Non-isomorphic 3D Rotational Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (2000) (CHI '00)*. ACM, 540–547. DOI:<http://dx.doi.org/10.1145/332040.332497>
- [125] Ivan Poupyrev, Suzanne Weghorst, Takahiro Otsuka, and Tadao Ichikawa. 1999. Amplifying Spatial Rotations in 3D Interfaces. In *CHI '99 Extended Abstracts on Human Factors in Computing Systems (CHI EA '99)*. Association for Computing Machinery, New York, NY, USA, 256–257. DOI:<http://dx.doi.org/10.1145/632716.632874>
- [126] Abbas Rafii. 2014. Two-dimensional method and system enabling three-dimensional user interaction with a device. (April 3 2014). <https://patentimages.storage.googleapis.com/4b/15/08/06d12780bfec5d/US8686943.pdf> US Patent 8,686,943 B1.
- [127] Nimesha Ranasinghe, Adrian David Cheok, Owen Noel Newton Fernando, Hideaki Nii, and Gopalakrishnakone Ponnampalam. 2011. Electronic Taste Stimulation. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp '11)*. Association for Computing Machinery, New York, NY, USA, 561–562. DOI:<http://dx.doi.org/10.1145/2030112.2030213>
- [128] Nimesha Ranasinghe, Adrian David Cheok, and Ryohei Nakatsu. 2012. Taste/IP : The Sensation of Taste for Digital Communication. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction (ICMI '12)*. Association for Computing Machinery, New York, NY, USA, 409–416. DOI:<http://dx.doi.org/10.1145/2388676.2388768>
- [129] Jaime Ruiz and Yang Li. 2011. DoubleFlip : A Motion Gesture Delimiter for Mobile Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2717–2720. DOI:<http://dx.doi.org/10.1145/1978942.1979341>
- [130] Steven Rybicki, Brian DeRenzi, and James Gain. 2016. Usability and Performance of Mouse-Based Rotation Controllers. In *Proceedings of the 42nd Graphics Interface Conference (GI '16)*. Canadian Human-Computer Communications Society, Waterloo, CAN, 93–100.
- [131] Oliver Schneider, Jotaro Shigeyama, Robert Kovacs, Thijs Jan Roumen, Sebastian Marwecki, Nico Boeckhoff, Daniel Amadeus Gloeckner, Jonas Bounama, and Patrick Baudisch. 2018. DualPanto. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI:<http://dx.doi.org/10.1145/3242587.3242604>
- [132] Youngbo Aram Shim, Jaeyeon Lee, and Geehyuk Lee. 2018. Exploring Multimodal Watch-back Tactile Display using Wind and Vibration. In *Proceedings of the 2018*

- CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173706>
- [133] Ken Shoemake. ARCBALL : A User Interface for Specifying Three-dimensional Orientation Using a Mouse. In *Proceedings of the Conference on Graphics Interface '92 (1992)*. Morgan Kaufmann Publishers Inc., 151–156. <http://dl.acm.org/citation.cfm?id=155294.155312>
- [134] Nikos Sidiropoulos, Sina Hadi Sohi, Thomas Lin Pedersen, Bo Torben Porse, Ole Winther, Nicolas Rapin, and Frederik Otzen Bagger. 2018. SinaPlot : an enhanced chart for simple and truthful representation of single observations over multiple classes. *Journal of Computational and Graphical Statistics* 27, 3 (2018), 673–676.
- [135] Alexa F. Siu, Eric J. Gonzalez, Shenli Yuan, Jason B. Ginsberg, and Sean Follmer. 2018. shapeShift. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173865>
- [136] Kinga Skorupska, Manuel Nunez, Wieslaw Kopec, and Radoslaw Nielek. 2018. Older Adults and Crowdsourcing : Android TV App for Evaluating TEDx Subtitle Quality. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 159 (Nov. 2018), 23 pages. DOI : <http://dx.doi.org/10.1145/3274428>
- [137] Brian A. Smith and Shree K. Nayar. 2018. The RAD. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3174090>
- [138] Marco Speicher, Anna Maria Feit, Pascal Ziegler, and Antonio KrÃ¼ger. 2018. Selection-based Text Entry in Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3174221>
- [139] Ryo Suzuki, Jun Kato, Mark D. Gross, and Tom Yeh. 2018a. Reactile. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173773>
- [140] Ryo Suzuki, Junichi Yamaoka, Daniel Leithinger, Tom Yeh, Mark D. Gross, Yoshihiro Kawahara, and Yasuaki Kakehi. 2018b. Dynablock. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3242587.3242659>
- [141] Shan-Yuan Teng, Tzu-Sheng Kuo, Chi Wang, Chi huan Chiang, Da-Yuan Huang, Liwei Chan, and Bing-Yu Chen. 2018. PuPoP. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3242587.3242628>

- [142] M. Tovey, S. Porter, and R. Newman. 2003. Sketching, concept development and automotive design. *Design Studies* 24, 2 (2003), 135 – 153. DOI : [http://dx.doi.org/https://doi.org/10.1016/S0142-694X\(02\)00035-2](http://dx.doi.org/https://doi.org/10.1016/S0142-694X(02)00035-2)
- [143] Daniel Trindade, André Rodrigues, Tiago Guerreiro, and Hugo Nicolau. 2018. Hybrid-Braille. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173601>
- [144] Erick Tseng, Daniel Johansson, Michael A. Cleron, and Indika C. Mendis. 2015. Long press interface interactions. (Feb. 3 2015). <https://patentimages.storage.googleapis.com/1a/34/db/59a5f539644bce/US8954887.pdf> US Patent 8,954,887 B1.
- [145] USPTO. 2018. Drawings section. (2018). Retrieved May 25th, 2020 from <https://www.uspto.gov/web/offices/pac/mpep/s1825.html>
- [146] Claus O Wilke. 2019. *Fundamentals of data visualization : a primer on making informative and compelling figures*. O'Reilly Media.
- [147] Anusha Withana, Daniel Groeger, and Jürgen Steimle. 2018. Tacttoo. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3242587.3242645>
- [148] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. 2009. User-defined Gestures for Surface Computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1083–1092. DOI : <http://dx.doi.org/10.1145/1518701.1518866>
- [149] Pui Chung Wong, Kening Zhu, and Hongbo Fu. 2018. FingerT9. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173752>
- [150] Ying Wu and Thomas S Huang. 2001. Hand modeling, analysis and recognition. *IEEE Signal Processing Magazine* 18, 3 (2001), 51–60.
- [151] Yu-Chian Wu, Te-Yen Wu, Paul Taele, Bryan Wang, Jun-You Liu, Pin sung Ku, Po-En Lai, and Mike Y. Chen. 2018. ActiveErgo. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3174132>
- [152] Haijun Xia, Nathalie Henry Riche, Fanny Chevalier, Bruno De Araujo, and Daniel Wigdor. 2018. DataInk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. DOI : <http://dx.doi.org/10.1145/3173574.3173797>
- [153] Jun Xie, Aaron Hertzmann, Wilmot Li, and Holger Winnemöller. 2014. PortraitSketch : Face Sketching Assistance for Novices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 407–417. DOI : <http://dx.doi.org/10.1145/2642918.2647399>

- [154] Jun Xie, Holger Winnemöller, Wilmot Li, and Stephen Schiller. 2017. Interactive Vectorization. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 6695–6705. DOI: <http://dx.doi.org/10.1145/3025453.3025872>
- [155] Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh. 2014. True2Form : 3D Curve Networks from 2D Sketches via Selective Regularization. *ACM Trans. Graph.* 33, 4, Article 131 (July 2014), 13 pages. DOI: <http://dx.doi.org/10.1145/2601097.2601128>
- [156] Keng-Ta Yang, Chiu-Hsuan Wang, and Liwei Chan. 2018. ShareSpace. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI: <http://dx.doi.org/10.1145/3242587.3242630>
- [157] yemount. 2020. Github, Pose-Animator. (2020). Retrieved Decembre 3rd, 2020 from <https://github.com/yemount/pose-animator>
- [158] D. Yoon, J. H. Lee, K. Yeom, and J. Park. 2011. Mobiature : 3D model manipulation technique for large displays using mobile devices. In *2011 IEEE International Conference on Consumer Electronics (ICCE)*. 495–496. DOI: <http://dx.doi.org/10.1109/ICCE.2011.5722702>
- [159] Yao Jun Zhao, Dmitri Shuralyov, and Wolfgang Stuerzlinger. Comparison of multiple 3D rotation methods. In *2011 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems Proceedings (2011-09)*. IEEE, 1–5. DOI: <http://dx.doi.org/10.1109/VECIMS.2011.6053855>
- [160] Jingjie Zheng, Blaine Lewis, Jeff Avery, and Daniel Vogel. 2018. FingerArc and FingerChord. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. DOI: <http://dx.doi.org/10.1145/3242587.3242589>
- [161] Youyi Zheng, Xiang Chen, Ming-Ming Cheng, Kun Zhou, Shi-Min Hu, and Niloy J Mitra. 2012. Interactive images : cuboid proxies for smart image manipulation. *ACM Trans. Graph.* 31, 4 (2012), 99–1.
- [162] Junhan Zhou, Yang Zhang, Gierad Laput, and Chris Harrison. 2016. AuraSense : Enabling Expressive Around-Smartwatch Interactions with Electric Field Sensing. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 81–86. DOI: <http://dx.doi.org/10.1145/2984511.2984568>