



THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Interaction gestuelle avec un Leap Motion état des lieux

Willemet, Alain

Award date:
2018

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Université de Namur
Faculté d'informatique
Année académique 2017–2018

**Interaction gestuelle
avec un *Leap Motion* :
état des lieux**

Alain Willemet



Promoteur : Bruno Dumas (Signature pour approbation du dépôt - REE art. 40)

Mémoire présenté
en vue de l'obtention du grade de
Master en Sciences Informatiques.

Remerciements

Je remercie mon promoteur Monsieur Bruno Dumas, professeur à l'université de Namur, pour sa direction de mémoire.

Sa disponibilité, ses conseils en méthodologie, ses commentaires et sa rigueur scientifique m'ont guidé pour mener à bien cette étude.

Je remercie également les membres du jury d'évaluation.

Résumé

Le *Leap Motion* est un périphérique de reconnaissance de gestes. Introduit en 2013, il est le premier appareil compact qui permet de reconnaître les gestes de la main.

Pour analyser le *Leap Motion* de manière critique, trois questions de départ se posent :

- répond-il aux attentes des utilisateurs en matière de précision, de fiabilité et de robustesse ?
- quelles sont les applications les plus prometteuses du *Leap Motion* ?
- quelles sont les bonnes pratiques dans la création d'une application intégrant le *Leap Motion* ?

Pour explorer le sujet, le domaine de reconnaissance de gestes est établi : définitions, classifications, catégorisation des périphériques et types de reconnaissance. Le *Leap Motion* est présenté comme périphérique de reconnaissance de gestes par la vision et son intérêt est démontré.

Afin de répondre à la première question, des articles scientifiques récents traitant des qualités du *Leap Motion* sont analysés. Sur base de cette analyse, la précision, la robustesse, les capacités de pointage et d'interaction et les facultés de reconnaissance du capteur *Leap Motion* sont évalués.

Pour répondre à la deuxième question, les applications observées du *Leap Motion* sont classifiées dans l'état de l'art par thème, sur base d'articles scientifiques. Cette classification permet ensuite de les catégoriser. Enfin, une analyse critique identifie les plus prometteuses et suggère des améliorations.

Afin de répondre à la troisième question, l'interface de programmation API du *Leap Motion* est analysée : sa librairie dynamique, ses classes, ses six types de données et leurs paramètres. Des tests menés à l'aide de programmes java permettent d'évaluer ses capacités de reconnaissance.

Ensuite, un cas d'utilisation de l'application sur une carte est choisi. Il consiste à déplacer cette carte dans les quatre directions puis à la zoomer et à la dézoomer.

Afin d'identifier les gestes les plus intuitifs pour effectuer les opérations du cas d'application, les comportements de différents utilisateurs sont collectés lors d'une enquête. La carte est présentée à l'aide d'un ordinateur sur lequel aucun programme n'est implémenté. Les attentes d'interaction gestuelles sont expliquées puis les résultats sont synthétisés.

Sur base des gestes les plus utilisés lors de l'enquête, un prototype logiciel d'optimisation de l'interaction gestuelle est créé. Ce prototype est présenté aux utilisateurs de l'enquête afin de le tester. Ces tests vérifient la satisfaction des utilisateurs et identifient les éléments à améliorer en ce qui concerne l'interaction gestuelle.

Abstract

The *Leap Motion* is a gesture recognition device. Introduced in 2013, it is the first compact device to recognize hand gestures.

To critically analyze *Leap Motion*, three starting questions are:

- Meet user expectations for accuracy, reliability and robustness?
- What are the most promising applications of *Leap Motion*?
- What are the best practices in creating an application that integrates the *Leap Motion*?

To explore the subject, the gesture recognition domain is established: definitions, classifications, categorization of devices and types of recognition. The *Leap Motion* is presented as a gesture recognition device by the vision and its interest is demonstrated.

In order to answer the first question, recent scientific articles dealing with the qualities of the *Leap Motion* are analyzed. Based on this analysis, accuracy, robustness, pointing and interaction capabilities, and sensor recognition capabilities are evaluated.

To answer the second question, the observed applications of the *Leap Motion* are classified in the state of the art by theme, based on scientific articles. This classification then makes it possible to categorize it. Finally, a critical analysis identifies the most promising and suggests improvements.

API of *Leap Motion* is analyzed: its dynamic library, its classes, its six data types and their parameters. Tests conducted using Java programs can evaluate its recognition capabilities.

Then, a use case of the application on a map is chosen. He composes this map in all four directions and then zooms in and out.

In order to identify the most intuitive gestures to perform the application operations, the behaviors of different users are collected during an investigation. The map is presented using a computer on which no program is applied. The expectations of gestural interaction are explained then the results are synthesized.

Based on the gestures most used during the investigation, a gesture interaction optimization software prototype is created. This prototype is presented to the users of the survey to test it. These tests check the satisfaction of the users and identify the elements to be improved regarding gestural interaction.

Table des matières

Résumé	4
1. Vue d'ensemble de la reconnaissance de gestes.....	13
1.1 Définitions.....	13
1.2 Classifications des gestes	14
1.3 Les périphériques de reconnaissance de gestes	16
1.4 La reconnaissance des gestes	21
1.4.1 La reconnaissance des gestes par la vision	22
1.4.2 La reconnaissance des gestes par le toucher	23
2. Le contrôleur Leap Motion	24
2.1 Intérêt de l'étude du contrôleur Leap Motion.....	24
2.2 Présentation du contrôleur Leap Motion.....	24
2.3 La plate-forme du contrôleur Leap Motion	29
3. Evaluation des performances du Leap Motion.....	32
3.1 Précision, fiabilité et robustesse du contrôleur Leap Motion	32
3.1.1 Précision et fiabilité du Leap Motion pour la mesure de position des doigts.....	32
3.1.2. Analyse de la précision, de la fiabilité du capteur Leap Motion et son aptitude pour le suivi statique, dynamique.....	33
3.1.3 Analyse de la précision et de la robustesse du contrôleur Leap Motion.....	33
3.2 Evaluation du contrôleur Leap Motion comme appareil de pointage.....	35
3.2.1 Evaluation du contrôleur Leap Motion comme nouvel appareil de pointage sans contact	35
3.3 Interaction 3D en utilisant le contrôleur Leap Motion	37
3.3.1 Exploration de l'interaction 3D en utilisant le contrôleur Leap Motion.....	37
3.4 Comparaison entre le Leap Motion et le Microsoft Kinect	38

3.4.1	Evaluation comparative du Leap Motion et du Microsoft Kinect	38
3.4.2	Reconnaissance des gestes de la main avec le Leap Motion et le Kinect...	38
3.5	Amélioration des facultés de reconnaissance de gestes du Leap Motion ..	39
3.5.1	Le Leap Motion pour la capture de mouvement 3D par spline	39
3.5.2	Modélisation du squelette de la main.....	39
3.5.3	Utilisation d'un algorithme non paramétrique	39
3.5.4	Intégration d'autres capteurs pour percevoir d'autres types de signaux.....	40
4.	Etat de l'art des applications du Leap Motion	43
4.1	Compte-rendu des applications possibles avec le Leap Motion.....	43
4.1.1	Peinture	43
4.1.2	Contrôle de systèmes domotiques.....	44
4.1.3	Commande par les gestes d'un robot chargé des tâches domestiques	44
4.1.4	Contrôle d'un fauteuil roulant.....	45
4.1.5	Réhabilitation de personne.....	46
4.1.6	Contrôle d'images en chirurgie.....	47
4.1.7	Contrôle aérien	48
4.1.8	Contrôle par les gestes et le regard	48
4.1.9	Langage des signes.....	49
4.1.10	Réalité virtuelle	51
4.1.11	Applications musicales	52
4.1.12	Applications industrielles.....	54
4.1.13	Applications éducatives.....	56
4.1.14	Authentification biométrique	57
4.2	Catégorisation des applications du contrôleur Leap Motion	58
4.3	Etat de l'art des applications du Leap Motion : discussion	63

4.3.1	Types de gestes employés dans les différentes études	63
4.3.2	Utilisation du Leap Motion dans la vie courante	63
4.3.3	Utilisation du Leap Motion avec d'autres capteurs	64
4.3.4	Interaction en environnement tridimensionnel	66
4.3.5	Interactions multiples	66
4.3.6	Utilisation du Leap Motion pour une application musicale.....	67
4.3.7	Utilisation du Leap Motion pour la réhabilitation de personnes.....	68
4.3.8	Reconnaissance de gestes.....	68
4.3.9	Utilisation du Leap Motion en chirurgie ouverte.....	69
4.3.10	Utilisation du Leap Motion avec les jeux	70
4.3.11	Applications particulières.....	70
4.3.12	Utilisation du contrôleur Leap Motion pour les applications web.....	71
5.	L'interface de programmation API Java du contrôleur Leap Motion	72
5.1	Présentation de l'interface de programmation API Java du Leap Motion.	72
5.2	Evaluation de l'interface de programmation API Java du Leap Motion ...	76
5.3	Programmes Java Leap Motion.....	79
5.4	Tests des programmes Java Leap Motion	80
6.	Etude de cas (Use case).....	83
6.1	Méthodologie de l'enquête avec utilisateurs.....	84
6.2	Enquête utilisateur magicien d'Oz.....	85
6.3	Conclusion de l'enquête avec utilisateurs.....	94
6.4	Directives de conception	96
6.5	Implémentation de l'interface suite aux résultats de l'enquête.....	97
6.5.1	Résultats des tests.....	102
6.5.2	Conclusion des tests de l'étude de cas	105

7. Conclusions et perspectives	107
7.1 Contenu	107
7.2 Synthèse	108
7.3 Contributions personnelles.....	110
7.4 Perspectives futures.....	111
Bibliographie	112

Introduction

Le *Leap motion* est un périphérique de reconnaissance de gestes par la vision développé à partir de 2013. Il représente la première application industrielle permettant la reconnaissance des gestes de la main.

De ce fait, il autorise un nouveau genre d'interaction avec les mains et les doigts. La reconnaissance des gestes peut se faire en deux dimensions de manière statique (posture) ou en trois dimensions de manière dynamique.

Le constructeur du *Leap Motion* le présente comme étant précis, fiable et sensible. Il est aussi portable, facile d'utilisation et peu onéreux. Il constitue l'objet de ce mémoire.

Le point de départ est de s'interroger sur les défis et perspectives du *Leap Motion*. En effet, la reconnaissance des gestes de la main en deux ou en trois dimensions rend possible de nouvelles interactions homme-machine comme les commandes gestuelles ou comme celles offertes entre un acteur réel et un acteur virtuel.

Les questions de recherche sont :

- le *Leap Motion* répond-il aux attentes des utilisateurs en termes de précision, de fiabilité et de robustesse pour un suivi de gestes statiques ou dynamiques dans un environnement bi ou tridimensionnel de tâches usuelles (pointage, manipulation d'objet,...) ?
- quelles sont les applications les plus prometteuses du *Leap Motion* ?
- quelles sont les bonnes pratiques dans la création d'une application intégrant le *Leap Motion* ?

Méthodologie de recherche :

Les deux premières étapes du travail introduisent la problématique relative au contrôleur *Leap Motion*.

La première étape est l'exploration et la lecture des documents trouvés via le moteur de recherche *Google scholar* sur base de mots clé. Le but de cette étape est de récolter des informations générales dans le domaine de la reconnaissance de gestes : définitions, classifications des gestes, catégorisation des périphériques de reconnaissance.

Les principaux périphériques sont répertoriés suivant leur date d'apparition et leur catégorie. Les étapes du processus de reconnaissance de gestes par la vision sont expliquées ainsi que le principe de reconnaissance de geste par le toucher.

Les sources utilisées dans cette première étape sont des ouvrages généraux traitant des gestes et plus spécifiquement des gestes de la main ainsi que, des références sur les interactions homme machine et sur les gestes en environnement tridimensionnel.

La seconde étape du travail a pour but de se familiariser avec le contrôleur *Leap Motion*. Cette étape permet d'en connaître les éléments suivants : caractéristiques dimensionnelles et techniques, composants, fonctionnement, champ de vision, volume et zones d'interaction, système de coordonnées, entités détectées (mains, doigts, outils, mouvements et gestes), plate-forme ainsi que les possibilités de développement.

Les sources utilisées pour cette seconde étape sont les sites internet du constructeur *Leap Motion* et des exposés s'y rapportant.

La troisième étape du travail est d'évaluer les performances du contrôleur *Leap Motion*. Son but est de fournir la réponse à la première question de recherche : le *Leap Motion* possède-t-il les capacités attendues des utilisateurs en termes de précision, fiabilité et robustesse ?

L'évaluation des performances consiste en un résumé de plusieurs références scientifiques, porte sur les capacités attendues, ainsi que pointage et interaction 3D. Des améliorations de ses performances sont proposées.

Les sources utilisées pour cette troisième étape sont plusieurs références scientifiques d'études comportant des tests portant sur le contrôleur *Leap Motion*.

La quatrième étape du travail est de fournir un état de l'art des applications du contrôleur *Leap Motion* suivant une thématique en se basant sur des références scientifiques. La quatrième étape a pour but de répondre à la deuxième question de recherche : quelles sont les applications les plus prometteuses du *Leap Motion* ?

Les applications sont catégorisées par thème dans un tableau afin d'en dégager les caractéristiques. Sur base de ce tableau, est proposée une discussion personnelle et critique de l'état de l'art des applications du *Leap Motion* ainsi que de leurs possibles améliorations.

Les sources utilisées pour cette quatrième étape sont quarante références scientifiques portant sur le contrôleur *Leap Motion*. Chacune d'entre elle est résumée en reprenant les éléments suivants : le but de l'application, le thème, l'année de parution, le type de geste et les interfaces de programmation employées, les autres capteurs utilisés conjointement, les particularités et les tests réalisés.

Les deux dernières cinquième et sixième étapes ont pour but de répondre à la troisième question de recherche : quelles sont les bonnes pratiques dans la création d'une application intégrant le *Leap Motion* ?

La cinquième étape du travail vise à prendre de connaissance de l'interface de programmation API java du contrôleur *Leap Motion* et a pour but d'en évaluer les capacités de reconnaissance de gestes. Pour ce faire, des programmes java ont été développés.

Cette étape permet d'appréhender les éléments suivants : la librairie dynamique de l'interface de programmation et ses classes, les six types de données et leurs paramètres. L'analyse des capacités de l'interface de programmation repose sur un résumé d'éléments factuels issus d'une quinzaine de références scientifiques.

Les sources utilisées pour la cinquième étape sont les sites internet du constructeur du *Leap Motion* et une quinzaine de références scientifiques.

La sixième étape du travail consiste en une observation et une analyse sous la forme d'étude de cas portant sur une carte. Elle consiste à déplacer la carte dans les quatre directions, à la zoomer et la dézoomer à l'aide de gestes. Afin d'identifier les gestes les plus intuitifs, une enquête utilisateur est réalisée. Enfin, un prototype réagissant aux gestes les plus utilisés d'après l'enquête utilisateur est créé. Celui-ci est confronté aux utilisateurs de l'enquête afin de connaître leur satisfaction et d'identifier des éléments à améliorer en matière d'interaction gestuelle.

La méthodologie commence par une introduction générale, puis elle répond de plus en plus précisément aux questions de recherche posées pour aboutir finalement à un prototype utilisable.

La manière de répondre aux questions de recherche est la suivante :

- Les capacités du contrôleur et celles de son interface de programmation sont résumées et commentées. Il en résulte des différences par rapport à celles annoncées par le constructeur ;
- Les applications sont répertoriées sur base d'un état de l'art. Des applications plus prometteuses et des améliorations envisageables sont énoncées ;
- Les bonnes pratiques dans la création d'une application intégrant le *Leap Motion* sont identifiées : elles consistent à acquérir les connaissances techniques, réaliser une enquête utilisateur pour constituer un catalogue de gestes intuitifs de l'application, créer un prototype réagissant aux gestes les plus utilisés d'après l'enquête puis finalement confronter ce prototype aux utilisateurs afin de connaître leurs impressions et améliorer l'interaction gestuelle.

Il est à noter que l'interface de programme ne reconnaît pas systématiquement les gestes de base car il les confond avec d'autres gestes, il n'appréhende pas le démarrage du geste, les conditions de luminosité sont mauvaises et il y a des problèmes d'occultation ou la vitesse du geste n'est pas optimum.

En conclusion, la problématique est rappelée, les résultats de l'observation et de l'analyse présentés pour mettre en évidence les nouvelles connaissances, poser les questions critiques et les perspectives qui en découlent.

1. Vue d'ensemble de la reconnaissance de gestes

1.1 Définitions

Afin de circonscrire le cadre de la recherche, il est utile de définir l'interaction gestuelle avec les périphériques classiques et d'énoncer les avantages de l'interface gestuelle. Deux définitions du geste, l'une générale, l'autre plus spécifique permettent de compléter ce propos.

Actuellement, la façon la plus usuelle d'interagir avec un ordinateur est réalisée au moyen des périphériques d'entrée classiques tels que le clavier ou la souris.

Avec ces périphériques, l'interaction c'est-à-dire l'action coordonnée entre l'homme et l'ordinateur est limitée au minimum. Elle permet l'entrée et la sortie des données ainsi que l'activation de commandes mais uniquement de manière rationnelle et intelligible.

L'expérience utilisateur peut être enrichie grâce à l'interaction gestuelle avec l'ordinateur car elle est plus naturelle et plus intuitive. De plus, elle peut être source d'émotion et de plaisir.

Les interfaces gestuelles apportent les avantages suivants [Baudel et al, 1993]¹ :

- Les gestes constituent un moyen de communication naturel pour l'homme et sont plus expressifs que le langage uniquement parlé. Ils contiennent des idées et de l'information. Certains d'entre eux sont universels et semblables à une manipulation directe d'objets dans le monde réel ;
- De par l'aspect dynamique du geste, l'homme peut, en une seule action, activer une commande et ses paramètres ;
- Le périphérique d'entrée permet une interaction directe entre l'homme et l'ordinateur.

Une première définition générale du geste est la suivante [<https://fr.wikipedia.org>]² : « Un geste est un signe manuel ou corporel qui permet d'illustrer les mots du langage, de les compléter ou de les appuyer. Le geste peut aussi être utilisé pour remplacer la parole. On distingue différentes catégories de gestes : certains apportent une information sur le message. D'autres apportent plutôt des indices sur les intentions ou émotions du locuteur. Les gestes forment une partie importante de la communication non verbale. »

Une deuxième définition du geste en relation avec la communication est la suivante [Kendon, 2004]³ & [Dumas, 2018]⁴ : « Un geste est une forme de communication non verbale dans laquelle les actions visibles du corps engendrent des messages particuliers, en lieu et place du langage parlé ou en parallèle avec des mots. Les gestes incluent les mouvements de la main, du visage, ou d'autre partie du corps. Ils diffèrent de la communication non verbale car ils ne communiquent pas de message purement expressif, proxémique ou d'attention. »

1.2 Classifications des gestes

Les gestes peuvent être classés suivant différents critères. Les classifications des gestes de la main propres au capteur *Leap Motion* présentées ci-dessous vont de la plus générale à la plus spécifique.

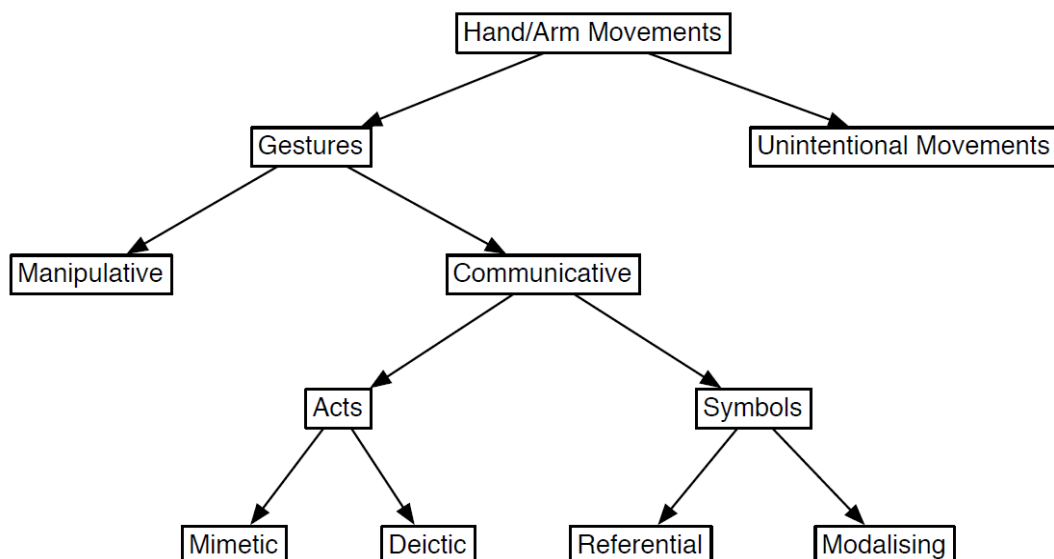
Cadoz [Cadoz, 1994]⁵ propose une classification des gestes suivant leur fonction :

- La fonction épistémique qui consiste à percevoir l'information depuis l'environnement ;
- La fonction ergodique qui concerne la manipulation, la régulation ou la modification de l'environnement physique ;
- La fonction sémiotique qui permet la communication d'information significative vers l'environnement.

La majorité des gestes sont de nature sémiotique car l'homme agit sur l'environnement qui lui effectue un retour sensoriel. Ce retour sensoriel peut influencer le comportement de l'homme et concernerait d'avantage la fonction ergodique. Rime et Schiaratura [Rime et al, 1991]⁶ donnent une classification des gestes sémiotiques dans les interactions homme - machine :

- Symboliques : les gestes qui ont une seule signification comme les gestes du langage des signes. L'ensemble de ces gestes est repris implicitement ou explicitement dans un dictionnaire ;
- Déictiques : les gestes de pointage ;
- Iconiques : les gestes utilisés pour transmettre une information comme la forme, la taille ou l'orientation d'un objet ;
- Pantomimiques : les gestes de l'acteur qui mime un mouvement d'un outil ou d'un objet dans sa main.

Les gestes qui nous intéressent principalement sont ceux de la main. Pavlovic, Sharma et Huang [Pavlovic et al, 1997]⁷ présentent une taxonomie des gestes de la main et du bras pour les interactions homme – machine [Dumas, 2018]⁸ :



Les mouvements de la main et du bras comprennent les gestes et les mouvements non intentionnels. Les mouvements non intentionnels se distinguent des gestes car ils n'apportent pas d'information significative.

Les gestes sont subdivisés en gestes manipulateurs et en gestes communicatifs.

- Les gestes manipulateurs sont ceux utilisés pour agir sur les objets dans un environnement afin d'induire par exemple une rotation ou un mouvement de ces objets.
- Les gestes communicatifs ont pour vocation de communiquer de l'information. Ils sont subdivisés en actes et en symboles.
 - Les symboles sont les gestes qui ont un rôle linguistique. Les symboles peuvent être référentiels ou modalisants.
 - Les symboles référentiels appelés également gestes iconiques représentent une action particulière ou un signe du langage.
 - Les symboles modalisants accompagnent le langage parlé pour le compléter.
 - Les actes sont les gestes liés directement à une interprétation du mouvement lui-même. Ils peuvent imiter des actions (gestes mimétiques) ou signifier un pointage (gestes déictiques).

Une gesticulation c'est-à-dire un ensemble de gestes peut aussi fournir de l'information. La posture concerne un geste statique. Une séquence de postures constitue un geste dynamique. Les gestes sont multidimensionnels soit bidimensionnels soit tridimensionnels.

Les gestes peuvent être caractérisés suivant le type de mouvement utilisé pour les effectuer.

Il existe les mouvements impulsifs qui comportent de fortes variations d'accélération pendant un court laps de temps et les mouvements fluides qui impliquent de faibles variations d'accélération pendant un laps de temps long.

La nature des gestes que l'homme réalise pour interagir avec l'ordinateur a un lien avec le type de contrôle et le retour d'information.

Le contrôle discret est souvent associé à des gestes symboliques et des gestes composés de mouvements impulsifs. Le contrôle continu est souvent associé à des gestes fluides. Il existe une relation directe entre les valeurs renvoyées par le capteur et les valeurs manipulées.

Le retour d'information permet à l'homme de percevoir l'effet de l'action qu'il vient de réaliser. Les retours possibles sont : le retour visuel (lunettes, écran), le retour sonore (écouteurs, haut-parleur) et le retour haptique (dispositifs tactiles, vibreurs). Le retour d'information se fait dans un contexte d'interaction dépendant de trois éléments : l'homme, la plate-forme et l'environnement. Un concepteur doit penser la technique au travers de ce contexte.

1.3 Les périphériques de reconnaissance de gestes

Les principaux périphériques de reconnaissance de gestes sont présentés ci-dessous suivant leur date d'apparition et leur catégorisation.

Il s'ensuit deux classifications graphiques l'une montrant la puissance en fonction de la précision, l'autre montrant l'encombrement en fonction de la précision, du contrôle. Ces graphiques présentent le positionnement du périphérique *Leap Motion* par rapport aux autres dispositifs de reconnaissance.

Il existe trois catégories de périphériques de reconnaissance de gestes :

- les périphériques basés sur des capteurs : les plus anciens utilisent des senseurs magnétiques ou des senseurs d'inertie pour suivre les mouvements, les plus modernes utilisent des accéléromètres qui donnent l'accélération propre dans une direction. Trois accéléromètres fournissent les accélérations dans les trois directions et permettent la reconnaissance dynamique des gestes ;
- les périphériques qui reposent sur la vision par ordinateur : ils utilisent des caméras pour analyser et interpréter les mouvements à partir de séquences enregistrées grâce au traitement d'images ;
- les périphériques qui reposent sur le toucher : ils utilisent un dispositif pouvant agir comme une extension du corps humain et ainsi en capter ses mouvements.

CAPTEURS

Data Glove
Dexterous Hand Master
Cyber Glove
Accele Glove Wiimote

VISION

Videoplace
Media Room
Real Estate Agent
Vision Wand
Microsoft Kinect
Leap Motion
Microsoft
HoloLens

TOUCHER

	Flexible machine Interface		Digital Desk		Fingerworks					iPhone			
	Videoplace				Diamond Touch					Microsoft Surface Computing			
	Multi-Touch Screen				Neonode N1								
	Multi-Touch Tablet				Perceptive Pixel								
A	1	1	1	1	1	1	2	2	2	2	2	2	2
n	9	9	9	9	9	9	0	0	0	0	0	0	0
n	8	8	8	8	8	8	9	9	0	0	0	0	0
é	2	3	4	5	7	9	1	2	3	4	5	7	0
e							8	9	1	2	3	4	5

Les périphériques basés sur des capteurs suivant leur date d'apparition :

- 1987 : *The DataGlove* [Kessler et al, 1995]⁹ (Zimmerman, Lanier and Harvill) : le gant utilise une dizaine de capteurs magnétiques et d'inertie pour capter les mouvements des doigts et de la main. A chaque doigt du gant sont fixées plusieurs fibres optiques. Le gant mesure l'atténuation de la lumière qui se propage dans la fibre optique lorsque le doigt se plie et possède une précision de 5 à 10 degrés. La fréquence d'utilisation est de 30 Hz ;
- 1987 : *Dexterous Hand Master* : gant à exosquelette ou à effet de Hall. Plusieurs segments articulés installés sur la main comportent des capteurs d'angle qui retournent des données relatives à la flexion des doigts. Il comporte une vingtaine de capteurs, possède une précision inférieure à un degré et fonctionne à une fréquence de 100 Hz ;
- 1989 : *The CyberGlove* (Kramer) : gant sans fil à variation de résistance qui mesure un angle de flexion en fonction de la déformation d'une lamelle métallique placée sur un doigt. Il utilise une vingtaine de capteurs afin de mesurer la posture de la main. Sa précision est d'un degré et

fonctionne à une fréquence de 120 Hz ;

- 2002 : *Accele Glove* : gant à accéléromètres. Il possède cinq accéléromètres à l'extrémité de chaque doigt et un sur la face dorsale de la main. En mesurant la direction de la gravité terrestre, ils permettent de déduire l'orientation de la main et des doigts ;
- 2006 : *Wii mote* : périphérique d'entrée de la console de jeu Wii de Nintendo pourvu d'accéléromètres afin de détecter des mouvements et des rotations en 3D. Il permet la reconnaissance dynamique de gestes. La manette permet de pointer des objets sur l'écran. Le pointage est reconnu au travers de la caméra infrarouge située au centre de la *Wii mote*.



DataGlove



CyberGlove



Accele Glove



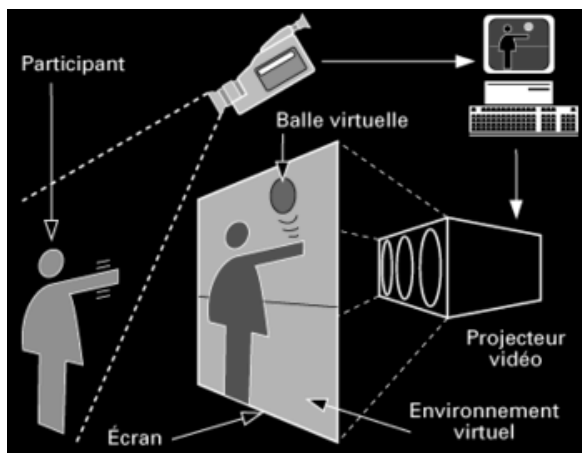
Wii mote

Les périphériques qui reposent sur la vision par ordinateur suivant leur date d'apparition :

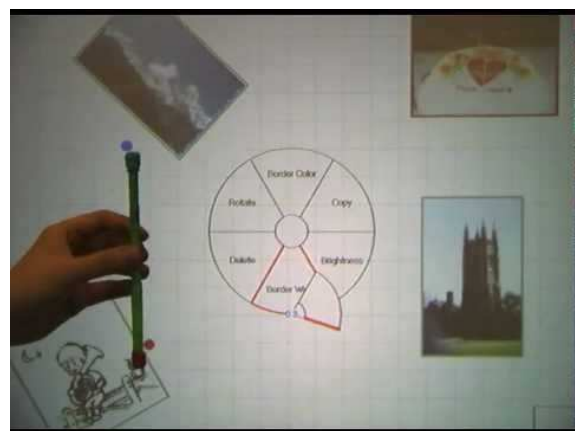
- 1983 : *Videoplace - Wall configuration* (M. Krueger) : procédé d'image en temps réel avec des gestes naturels dynamiques. Installation corporelle sans prothèse avec une stratégie de captation vidéo et une reconnaissance du mouvement de la silhouette du corps dont le contour devient une zone interactive ;
- 1984 : *The Media Room* (Bolt and Negro Ponte's Architecture Machine Group) : grande salle munie d'un panneau de projection sur le mur avec la possibilité de capter les gestes, la parole et le mouvement des yeux. Les gestes de pointage sont mesurés avec un senseur magnétique ;
- 1999 : *REA* (Cassel, MIT Media Laboratory) : *Real Estate Agent* ordinateur humanoïde qui donne des conseils à un acheteur immobilier. Les entrées se font au moyen de caméras et de sons, les sorties au moyen de discours, de gestes et de positions faciales. Le dispositif est orienté tâche et tient compte de l'aspect social. Les gestes en sortie sont exécutés suivant le contexte de communication sur base d'une grammaire et d'un lexique ;
- 2003 : *VisionWand* : une baguette pourvue d'extrémités facilement détectables est utilisée pour l'interaction avec des objets 2D, la sélection de menus et la reconnaissance de gestes. La baguette est filmée à l'aide de deux caméras qui suivent ses extrémités par la technique de vision en 3D. La position de la baguette par rapport à l'écran de projection est calculée par un algorithme. L'échelle des objets 2D est déterminée par la distance de la baguette à l'écran.

Un ensemble de gestes est identifié : pointage, rotation, déplacement parallèle, agrandissement, rétrécissement par apprentissage. A ces gestes dynamiques, peuvent être associés des commandes. D'autres commandes sont associées à des postures de la baguette en position statique ;

- 2010 : *Microsoft Kinect* : périphérique de contrôle de la console de jeu Xbox 360. Il permet la reconnaissance des gestes de tout le corps et de la parole. Un kit de développement logiciel permet de créer des applications. La Kinect possède un système de détection de profondeur 3D muni d'un émetteur et d'une caméra infrarouge. Elle dispose également d'une caméra à codage informatique des couleurs et d'un micro. Le kit de développement permet d'utiliser les données de la caméra de profondeur, de la caméra à codage des couleurs et du squelette de l'utilisateur ;
- 2013 : *Leap Motion* : un périphérique compact qui se branche sur le port USB de l'ordinateur et qui capte les mouvements des mains, des doigts et d'outils portés par les mains. L'appareil est un suiveur optique basé sur la vision stéréoscopique. La scène est illuminée par trois diodes LED. Deux caméras infrarouges en filment des images à deux dimensions qui seront synthétisées en trois dimensions ;
- 2015 : *Microsoft HoloLens* : lunettes intelligentes munies d'un ordinateur. Elles sont capables de détecter le regard, de reconnaître la voix et les gestes. Elles créent un environnement de réalité virtuelle. Le périphérique est autonome et muni de nombreux capteurs et processeurs. Le processeur central est épaulé par un processeur holographique qui traite les signaux des caméras et des capteurs embarqués. Le périphérique dispose de quatre caméras et de quatre diodes LED pour la reconnaissance de l'environnement, d'une double caméra pour le relief et d'une caméra qui enregistre le regard de l'utilisateur. Un algorithme de reconnaissance de formes permet de comprendre en temps réel les gestes de l'utilisateur. Un accéléromètre, un gyroscope et un magnétomètre captent les mouvements de la tête. Un dispositif d'affichage tête haute donne le rendu en réalité virtuelle. Malgré une technologie avancée, la reconnaissance de gestes est anecdotique.



Videoplace



VisionWand



Microsoft Kinect



Leap Motion

Les périphériques qui reposent sur le toucher suivant leur date d'apparition :

- 1982 : *A flexible machine Interface* (N. Meta) : système capable de capter les ombres des doigts et des objets par caméra ;
- 1983 : *Videoplace - Table configuration* (M. Krueger) : peut capter les mains et les doigts de plusieurs personnes. Il y a plusieurs possibilités de gestes. Tant qu'il existe un toucher, les entrées sont présentes et beaucoup d'évènements à gérer. La localisation d'une personne et les interactions collaboratives sont prises en compte ;
- 1984 : *Multi-Touch Screen* (B. Boie, M Hill, Bell Labs) : manipulation d'objets graphiques avec un bon temps de réponse. Le premier écran multi-tactile utilisait un réseau capacitif transparent de capteurs tactiles superposés sur un tube cathodique. Les sorties visuelles de l'écran sont directement sous les capteurs tactiles. Le dispositif reconnaît des points de contact simultanés et multiples puis les interprète grâce à un logiciel. Ceci inclut en général la position de chaque point de contact de façon indépendante, permet d'effectuer des gestes et d'interagir avec plusieurs doigts ;
- 1985 : *Multi-Touch Tablet* (Input Research Group, University Toronto) : permet de capter plusieurs entrées simultanées comme l'emplacement et le degré du toucher. Les entrées tactiles et les sorties visuelles sont séparées les unes des autres ;
- 1991 : *Digital Desk* (P. Wellner, Rank Xerox, University Cambridge) : c'est le premier système de réalité augmentée grâce à un projecteur et une caméra montés au-dessus d'un bureau traditionnel. Il utilise des techniques acoustique et optique. Les gestes de plusieurs doigts peuvent être reconnus ;
- 1998 : *Fingerworks* (Newark, Delaware) : cette firme produit des tablettes qui reconnaissent les gestes de plusieurs doigts. Le clavier *TouchStream LP* comprend une membrane plate et des touches à force nulle. La surface entière du clavier est dotée d'un support multi-tactile qui convertit les mouvements des doigts en événements gérés par la souris et en macros ;
- 2001 : *Diamond Touch* (Mitsubishi Labs) : il est capable de distinguer les mains des doigts. Beaucoup de gestes sont possibles. La table *Diamond Touch* est une interface multi-tactile qui permet à plusieurs personnes d'interagir simultanément tout en identifiant la position du toucher de la personne. La table est utilisée pour faciliter la collaboration en face à face, le brainstorming et la prise de décision ;
- 2004 : *Neonode NI* (Stockholm) : c'est le premier smartphone commandé par les gestes des doigts avec un retour d'information haptique ;
- 2005 : *Perceptive Pixel* (Jefferson Han) : le dispositif est composé d'une plaque de plexiglas translucide sur laquelle un vidéo projecteur projette l'interface utilisateur. Une caméra à infrarouges placée derrière l'écran intercepte les rayons infrarouges déviés par l'utilisateur. Cette technologie a l'avantage d'être adaptable à toutes les tailles d'écrans sans perte de qualité d'image. Le prix de cette technologie est très réduit. L'écran tactile de *Perceptive Pixel* est bien entendu Multi-Touch, et donc par extension multi-utilisateurs. Il est capable d'interpréter des gestes complexes, voire très complexes.
- 2007 : *Apple iPhone* : c'est un téléphone muni d'une interface basée sur le toucher. Il permet plusieurs sortes de gestes. L'interface de l'iPhone s'appuie sur écran tactile capacitif multipoint, dispose d'un appareil photo qui fonctionne également comme une caméra, d'un système de géolocalisation intégré et d'un logiciel de cartographie numérique. Un iPod intégré permet d'écouter et de télécharger de la musique. Il permet la navigation sur le Web, l'envoi et la consultation de son courrier électronique. Il dispose d'applications bureautiques, de fonctions élémentaires telles que les SMS et les MMS et d'autres applications disponibles par téléchargement ;
- 2007 : *Microsoft Surface Computing* : c'est une tablette interactive pour les doigts, les mains, les objets. *Surface computing* est utilisé pour décrire des environnements graphiques spécialisés dans lesquelles les éléments des environnements graphiques traditionnels sont remplacés par des éléments intuitifs. Les interactions de l'utilisateur se font directement sur un

écran tactile et peuvent être rapprochées d'actions familières de la vie de tous les jours comme la manipulation d'objets (photos, vidéos, jeux). Le Surface computing peut avoir un aspect social par exemple la rencontre entre amis autour d'une tablette-surface pour prendre un verre ou partager des photos et des vidéos ;



Diamond Touch



Perspective Pixel

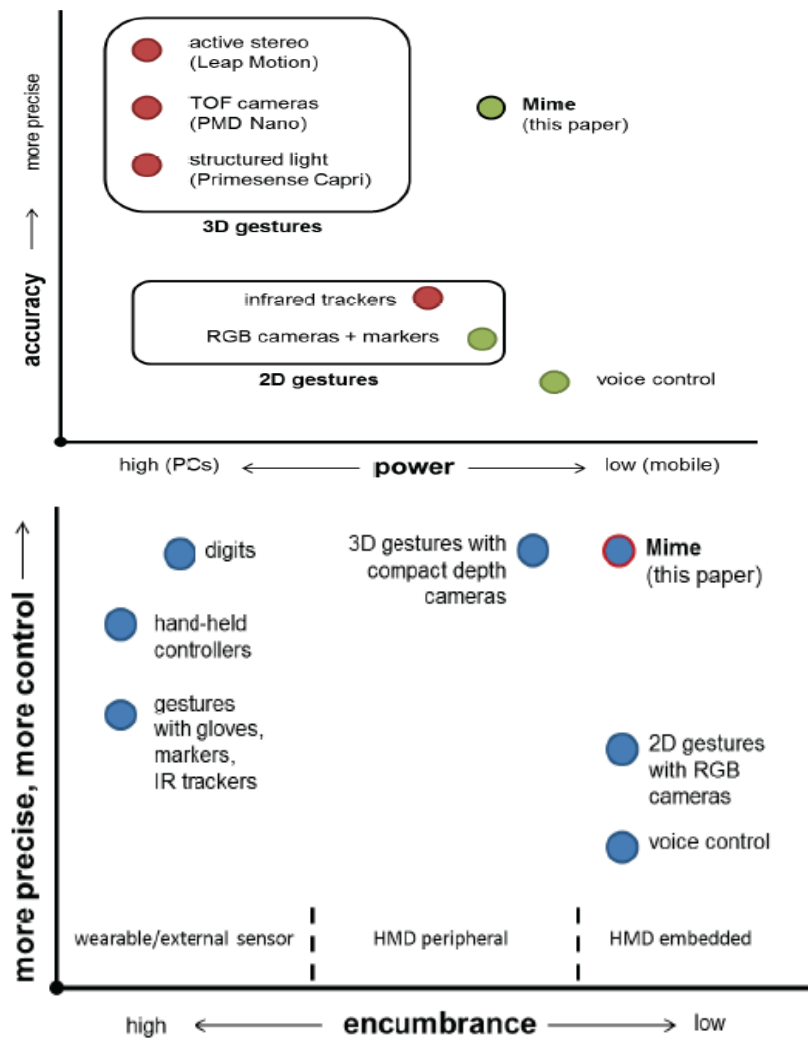


iPhone



Microsoft Surface Computing

Il est possible de classer graphiquement les périphériques : un premier graphique nous montre la puissance en fonction de la précision et un deuxième graphique nous montre l'encombrement en fonction de la précision et du contrôle [Colaço et al, 2013]¹⁰. Ce graphique est issu d'un article sur *Mime* un périphérique compact utilisé pour le contrôle de gestes tridimensionnels. C'est pourquoi les périphériques présentés plus haut n'y sont pas repris.



1.4 La reconnaissance des gestes

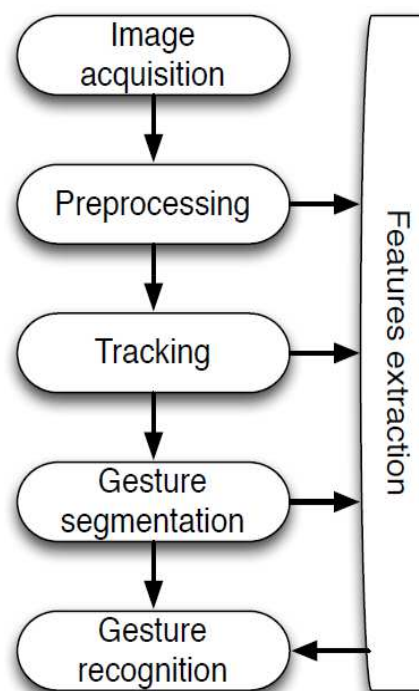
Les périphériques de reconnaissance de gestes utilisent deux manières distinctes de procéder : d'une part, la reconnaissance se fait par la vision au moyen de caméras comme dans le cas du capteur *Leap Motion*, d'autre part, la reconnaissance se fait par le toucher d'une surface tactile comme dans le cas des tablettes ou des smartphones.

1.4.1 La reconnaissance des gestes par la vision

La reconnaissance des gestes par la vision à l'aide de caméras a pour but d'analyser et d'interpréter les gestes de l'homme au moyen d'algorithmes afin de résoudre l'ambiguïté.

Le défi est de capturer les images des caméras et de suivre les mouvements de la main afin de reconnaître dynamiquement un geste.

La reconnaissance de gestes s'établit suivant un processus générique décomposable en plusieurs étapes [Dumas, 2018]¹¹ :



- l'acquisition d'images (Image acquisition) se fait avec une ou plusieurs caméras, en basse ou en haute résolution et en un certain nombre de scènes par seconde ;
- le prétraitement (Preprocessing) effectue la segmentation en soustrayant l'arrière-plan du sujet, en détectant la main et les contours ;
- le suivi (Tracking) : transforme l'image afin de garder des informations utiles puis suivre le sujet d'une scène à l'autre ;
- la segmentation des gestes (*Gesture segmentation*) : initialisation de la posture qui délimite les gestes, décomposition des gestes en sachant que seul le mouvement détient de l'information, approche statistique et règles déclaratives ;

- la reconnaissance des gestes (Gesture recognition) : les données sont analysées à l'aide d'algorithmes ;
- extraction des caractéristiques (Features extraction) : fourni la position, l'accélération, la vitesse, la profondeur spatio-temporelle à partir des étapes de prétraitement, de suivi et de segmentation des gestes du processus.

La reconnaissance statique est basée sur une seule scène. Elle compare les données de la scène avec des informations pré-enregistrées. La reconnaissance dynamique utilise plusieurs scènes et des données temporelles.

1.4.2 La reconnaissance des gestes par le toucher

Un pavé ou une surface tactile reconnaît des points de contact multiples et simultanés qui sont interprétés par un logiciel. Cela inclut en général la position de chaque point de contact de façon indépendante et permet d'effectuer des gestes et d'interagir.

Il existe différentes technologies :

- Les technologies électriques utilisées dans les tablettes, les smartphones ;
- les technologies optiques utilisées dans les grands systèmes (tables) ;
- les technologies à ondes basées sur les ondes acoustiques et les radiofréquences.

2. Le contrôleur Leap Motion

Les informations suivantes sont tirées des sites internet [<https://developer.leapmotion.com>] et [<http://www-igm.univ-mlv.fr/~dr/XPOSE2013/LeapMotion>]¹².

2.1 Intérêt de l'étude du contrôleur Leap Motion

Le contrôleur *Leap Motion* mérite une attention particulière car il est la première déclinaison industrielle qui reconnaît les gestes de la main. Il autorise un nouveau genre d'interaction homme-machine : le pilotage de l'ordinateur par ces gestes.

Il permet une interaction avec les mains, les doigts et les objets détenus par les mains. Il est plus précis qu'une souris, aussi fiable que le clavier et plus sensible qu'une tablette. La reconnaissance des gestes peut se faire en deux dimensions de manière statique (posture) ou en trois dimensions de manière dynamique. Il est portable, facile d'utilisation et peu onéreux.

Les applications possibles sont très variées. Elles incluent des applications avec des interactions 2D comme celles des smartphones et des applications plus innovantes 3D dans un volume d'interaction comme la manipulation d'objets virtuels.

Le contrôleur *Leap Motion* permet d'apporter de nouveaux gestes personnalisés en les programmant. Il suscite la création et une utilisation intuitive.

Ce sont pour toutes ces raisons qu'il focalise l'intérêt de ce mémoire.

2.2 Présentation du contrôleur Leap Motion

Le contrôleur *Leap Motion* est un périphérique d'entrée de données capable de détecter trois sujets :

- le centre de la main caractérisé par la distance au point de référence ;
- la position en trois dimensions de l'extrémité des doigts ;
- l'orientation de la main caractérisée par la distance, l'élévation et les angles par rapport au point de référence.

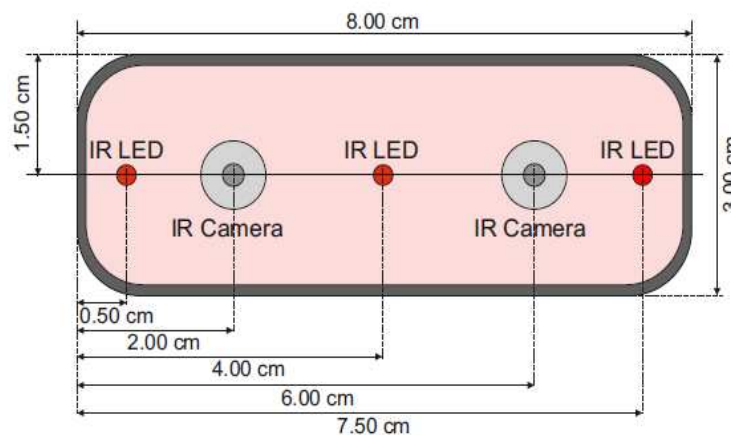
Ces paramètres, s'ils sont disponibles, sont délivrés à chaque cadre (*Frame*) et traités par un algorithme de reconnaissance des gestes.

Le contrôleur *Leap Motion* a été développé dès 2008 par Michael Buckwald et David Holz. Ceux-ci ont créé la société *Leap Motion Inc* en 2010. La première version du contrôleur a été rendue publique en octobre 2012. La seconde version de juillet 2013 était plus réactive et contrôlait mieux les variations de luminosité.

Le périphérique *Leap Motion* est un boîtier de dimensions 80 mm (longueur) x 30 mm (largeur) x 12,7 mm (hauteur) qui se branche à un ordinateur par une connexion USB.



Vue en taille réelle



Vue schématique

Le *Leap Motion* est un suiveur optique basé sur la vision stéréoscopique. Il dispose de LED indiquant son fonctionnement et d'un connecteur USB pour la transmission de données.

Il comprend deux caméras infrarouges monochromatiques 1,3 MP et trois diodes émettrices LED infrarouges. Il synthétise en trois dimensions des images deux dimensions prises par les deux caméras infrarouges.

La scène est illuminée par les diodes émettrices, les caméras captent la lumière réfléchie par l'objet suivi et une mesure « *time of flight* » est prise.

Cette mesure est proportionnelle à la distance de la caméra à l'objet et est effectuée par chaque pixel de la caméra permettant d'obtenir une image complète en 3D de l'objet suivi.

Les positions délivrées sont relatives au point central du *Leap Motion*. Le fonctionnement de l'appareil est perturbé par les sources externes de lumière infrarouges et peut dès lors se placer en mode robuste.

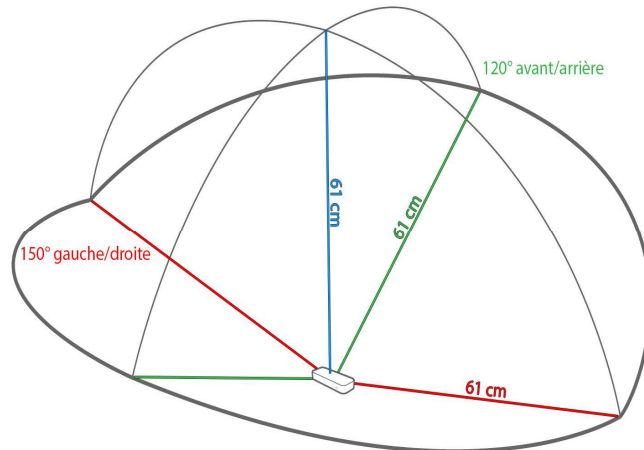
D'après le site internet du constructeur, le *Leap Motion* permet de déterminer la position de l'objet suivi dans un rayon d'un mètre avec une précision spatiale affichée de 0,01 mm à une fréquence de rafraîchissement maximum de 200/300 Hz. Ces valeurs seront vérifiées par la suite.

L'interaction gestuelle est précise. Le logiciel permet de récupérer la position et l'orientation de la main, de chaque doigt ou d'un objet pointeur. Les mouvements des doigts peuvent être capturés de

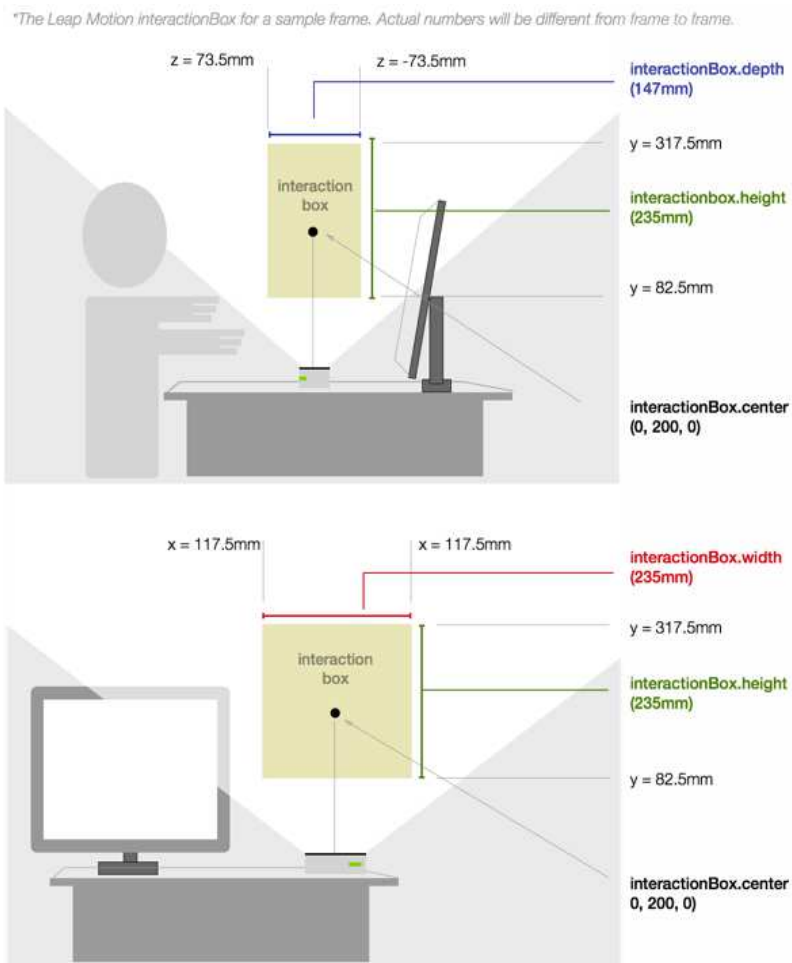
manière indépendante. Il peut suivre jusqu'à quinze doigts et son prix est abordable.

Le logiciel de traitement consomme assez bien de puissance CPU. Il est possible de diminuer cette consommation mais au détriment de la précision. Le *Leap Motion* chauffe beaucoup et rapidement.

Le *Leap Motion* a un champ de vision d'une pyramide inversée d'environ un mètre. Il capte tous les mouvements à moins de 60 cm de son centre, au-delà il perd beaucoup en précision.

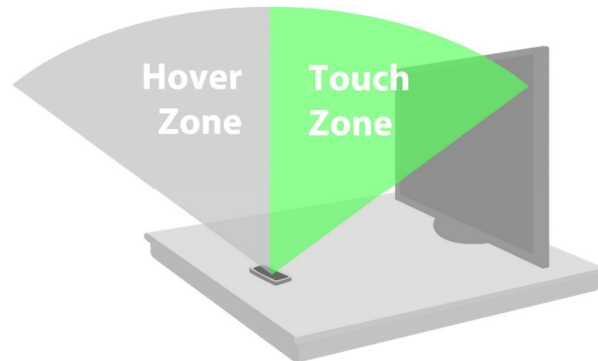


Pratiquement, la boîte d'interaction (*Interaction Box*) se positionne comme ci-dessous. Elle correspond au volume dans lequel doit se situer les objets pour être détectés.



Le *Leap Motion* possède deux zones d'interaction :

- La zone de proximité (*Hover Zone*) dans laquelle l'utilisateur se déplace, positionne les pointeurs ;
- la zone de contact (*Touch Zone*) dans laquelle l'évènement se crée.

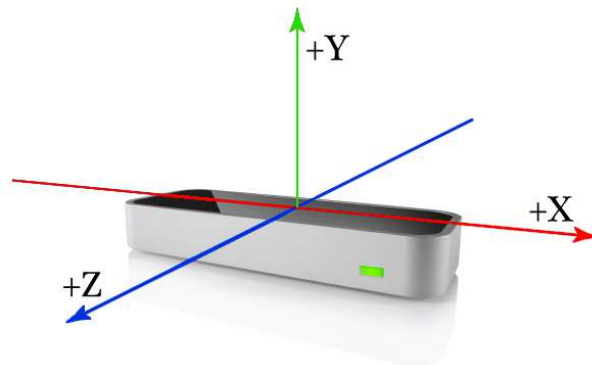


Pour interagir avec le *Leap Motion*, l'utilisateur peut effectuer des pointages précis ou d'amples mouvements. L'utilisateur gère le suivi en fonction de la vitesse et de la précision.

Il dispose de trois modes prédéfinis :

- précision (la précision est privilégiée par rapport à la vitesse) ;
- équilibré (la précision et la vitesse sont équitables) ;
- vitesse élevée (la vitesse est privilégiée par rapport à la précision).

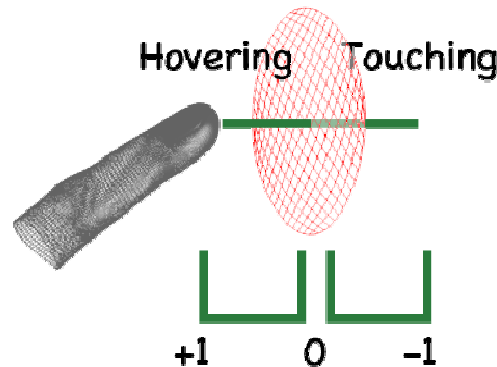
Le *Leap Motion* dispose de son propre système cartésien droitier de coordonnées. Le système a pour origine le centre du contrôleur. Les axes x et z se trouvent dans le plan horizontal, l'axe des x étant parallèle au grand côté de l'appareil. L'axe y est vertical, avec des valeurs positives augmentant vers le haut (contrairement à l'orientation vers le bas de la plupart des systèmes de coordonnées graphiques). L'axe z a des valeurs positives croissant vers l'utilisateur.



Le *Leap Motion* utilise une surface virtuelle pour simuler le toucher. Cette simulation utilise trois zones :

- La proximité (*Hovering*) ;
- Le toucher (*Touching*) ;
- Rien (*Null*).

La Zone *Null* permet d'atténuer le passage brusque d'une zone à l'autre. Le Leap Motion se déplace de 1 à -1. L'*Hovering* s'étend de 1 à 0 et le *Touching* de 0 à -1.



Les caméras infrarouges sont sensibles à la luminosité, il existe un mode robuste prenant en compte les conditions environnementales. De ce fait, il privilégiera la fiabilité des données de suivi au détriment de la fluidité. Ce mode peut s'activer automatiquement si la luminosité est trop faible.

Il y a peu de latence entre ce qui se passe en réalité et ce qui se voit à l'écran, c'est la raison pour laquelle le *Leap Motion* est utilisé également dans des applications basées sur la réalité virtuelle tridimensionnelle. Il est alors placé sur un casque de réalité virtuelle et permet l'interaction gestuelle. Il est intégré dans certains ordinateurs portables (*HP ENVY LEAP SE* ou *ASUS G750*) comme périphérique d'entrée.

Ces différentes possibilités induisent des applications dont voici quelques exemples classifiés par domaines :

- Education : écrans interactifs dans les classes, interaction avec différentes plates-formes, interfaces utilisateurs dédiés pour l'industrie ;
- Santé : accès aux données médicales et aux images en environnement stérile, mesures de régression et de progrès après ou en cours de traitement, contrôle de la chambre d'un patient, interaction de personnes ne pouvant se servir d'un clavier ou d'une souris ;
- Visualisation de données : données de navigation complexes autour des interfaces utilisateurs, identification (biométrie), décisions efficaces, données partagées et collaboration ;
- Industrie manufacturière : visualisation durant le processus de conception, application interactive avec l'environnement, gestion de la productivité, visualisation du procédé de fabrication, contrôle de robot ;
- Commerce de détail : accès d'informations dans les lieux publics (écrans ou kiosques pour obtenir un produit ou un service), expériences interactives dans les magasins dans le respect de la vie privée ;
- Divers : authentification biométrique non intrusive (scan de la main) utilisée pour le contrôle et la commande de l'application, applications CAD, transactions, systèmes d'accès, activation d'ordinateur portable, contrôle de système domotique, navigation au travers d'applications avec divers écrans, extractions d'informations, de vidéos ;
- Applications musicales ;
- Langage des signes.

2.3 La plate-forme du contrôleur Leap Motion

Les étapes de l'installation du *Leap Motion* sur l'ordinateur sont les suivantes : branchement du *Leap Motion* à un port USB, installation du logiciel *Leap Motion* et réalisation des calibrages nécessaires en créant des gestes personnalisés.

La plate-forme du *Leap Motion* est composée de trois éléments :

- Les périphériques : le contrôleur *Leap Motion* et la connexion USB à l'ordinateur ;
- Les drivers et le kit de développement SDK: le logiciel *Leap Motion* et les algorithmes ;
- Les applications métier et consommateurs, les relations développeurs (Airspace (*App Store*) et les applications tierces).

Il y a deux possibilités de développement :

- Le développement indigène sous mac, pc et linux écrit en C++ et compatible avec les langages suivants : C++, C#, objective C, Python, Java à partir du Software Development Kit (*SDK*) disponible sous developer.leapmotion.com. La compatibilité est assurée grâce à Swig un outil Open source générant les liaisons inter-langages ;
- Le développement web supporté par la plupart des moteurs web modernes disponible en JavaScript et sous js.leapmotion.com. La librairie JavaScript se compose de deux éléments : un Serveur Web Socket et la bibliothèque JavaScript du coté client.

Sur le portail du développeur, il y a le *Software Development Kit* (*SDK*) natif, la documentation et des exemples. La documentation est répartie par langage et par différentes interfaces de programmation API (*Application Programming Interface*). Il existe des forums de discussion. Le *Software Development Kit* comprend différentes classes pour concevoir une application.

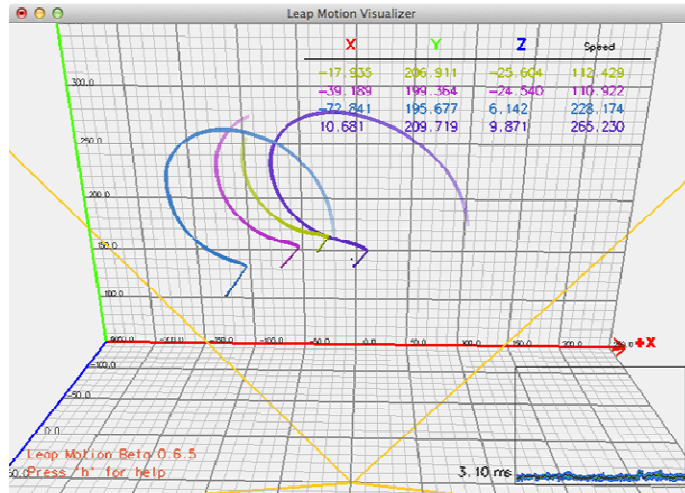
Sous l'API JavaScript, Il y a beaucoup d'exemples et de tutoriels. JavaScript permet de réaliser rapidement un prototype et de visualiser le code source. La réalisation rapide d'un prototype permet de vérifier si l'utilisateur peut interagir avec l'interface et s'il a une bonne idée spatio-temporelle de ce que le logiciel lui demande.

L'API JavaScript est intéressante pour débiter et affiner le prototype avant de se lancer dans un développement sous java par exemple. Les applications disponibles sont catégorisées : productivité, jeux, éducation, outils créatifs, science, applications d'origine, plugin, application web, musique.

Dans la façon de concevoir une application avec *le Leap Motion*, les interfaces classiques bidimensionnelles (souris et claviers) ne doivent pas nécessairement disparaître. L'important est d'améliorer l'interaction par une approche tridimensionnelle par exemple : comment améliorer la navigation, comment procurer plus de plaisir, comment améliorer la réalité virtuelle, faire quelque chose que l'utilisateur ne faisait pas auparavant avec les interfaces classiques et rendre l'interaction plus naturelle.

L'utilisateur doit se familiariser avec la technologie, il existe une période d'apprentissage. Les données que le système enregistre peuvent nous servir.

Il est possible grâce au visionneur (*Visualizer*) de regarder les actions afin de se rendre compte des interactions et de s'assurer que l'utilisateur soit conscient de l'espace 3D. Un objet est caractérisé par sa vitesse et sa position à un moment donné.



Dans un cadre donné (*Frame*) le Leap Motion voit à l'aide du visionneur (*Leap Motion Diagnostic Visualizer*) :

- Les mains (*Hands*) ;
- Les doigts (*Fingers*) ;
- Les outils (*Tools*).

Les mouvements repris dans l'API sont :

- La translation (*Translate*) ;
- La rotation (*Rotate*) ;
- La mise à l'échelle (*Scale*).

Au niveau des gestes, l'API est censé reconnaître des gestes simples comme :

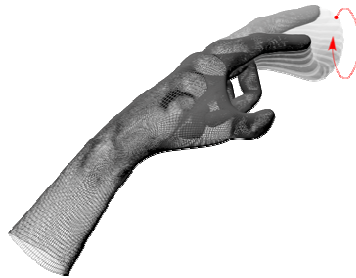
- des cercles (*CircleGesture*) ;
- un balayage avec les mains (*SwipeGesture*) ;
- un pointage avec un doigt (*KeyTapGesture* ou *ScreenTapGesture*).

La reconnaissance du mouvement commence au moment où l'utilisateur effectue un mouvement continu dans un sens et il se termine si l'utilisateur ralentit sa vitesse de déplacement ou s'il change de direction. Les données sont mises à jour à partir du moment où le mouvement a commencé.

Le *Leap Motion* peut reconnaître un mouvement de balayage à l'aide de la classe *SwipeGesture*.



Le *Leap Motion* peut reconnaître un mouvement circulaire de l'index à l'aide de la classe *CircleGesture*.

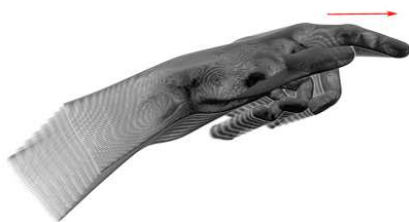


Le *Leap Motion* identifie deux types de toucher :

- le *Key Tap* un mouvement vers le bas similaire à une pression sur un clavier d'ordinateur géré par la classe *KeyTapGesture* ;



- le *Screen Tap* un mouvement vers l'écran similaire à une pression tactile sur celui-ci géré par la classe *ScreenTapGesture*.



Il y a plusieurs API correspondant soit à la zone de contact (*Touch Zone*) soit à la boîte d'interaction (*Interaction Box*).

3. Evaluation des performances du Leap Motion

Le but de ce chapitre est de présenter suivant différents aspects, une évaluation des performances du contrôleur *Leap Motion*. Cette évaluation est une version résumée basée sur plusieurs références scientifiques dont le but est de tester le contrôleur *Leap Motion* dans un domaine particulier.

3.1 Précision, fiabilité et robustesse du contrôleur Leap Motion

La précision, la fiabilité et la robustesse du contrôleur *Leap Motion* sont présentés et commentés au travers de trois références. La première traite de la position des doigts, la seconde de l'aptitude pour un suivi statique et dynamique et la troisième utilise un robot comme appareil référent pour ses tests.

3.1.1 Précision et fiabilité du Leap Motion pour la mesure de position des doigts

“Evaluation of a portable markerless finger position capture device: accuracy of the Leap Motion controller in healthy adults” [Tung et al, 2015]¹³

Le but de cette étude est d'évaluer le *Leap Motion* pour mesurer la position d'un doigt de patients dans le cas de besoins neuromoteurs cliniques comme atteindre et pointer.

L'évaluation examine les conditions qui influencent la fiabilité du *Leap Motion* pour le positionnement du doigt en particulier le fait de toucher un objet extérieur et valide les résultats grâce à un système optique de capture des mouvements.

Participants à l'expérience : neuf patients adultes entre 18 et 30 ans (six femmes, trois hommes) sans problèmes neurologiques.

Appareillage : un système optique de capture de mouvement *Optotrak 3020* placé à l'extrémité de l'index, un contrôleur *Leap Motion* version 1.2.0, un ordinateur portable *Lenovo G550*, le logiciel SDK en C++. Le capteur *Leap Motion* se trouve à 7,5 cm de l'écran, à 26 cm du bord de la table et ses axes de coordonnées sont alignés sur l'*Optotrak 3020*.

Condition de test : en touchant de la cible (écran) et à quatre centimètres de la cible (sans toucher). Les données du *Leap Motion* sont comparées au marqueur optique *Optotrak* attaché à l'index à l'aide du logiciel *Matlab R13*.

Sur l'ensemble des mesures, la moyenne quadratique (*RMS*) de l'erreur du *Leap Motion* est de 17,30 +/- 9,56 mm échantillonné à une fréquence de 65,47 +/- 21,53 Hz. Les mesures acceptables sont significativement moindres dans les conditions de toucher (44 % contre 64 % sans toucher). Le *Leap Motion* est moins fiable lorsque l'utilisateur touche le stimulus visuel présent sur l'écran.

Il est inconsistant à proximité des limites de son champ visuel. Le *Leap Motion* est suffisamment fiable pour mesurer une performance motrice d'une tâche de pointage qui ne demande pas une haute précision de position. Il peut, par exemple, être utilisé pour vérifier la coordination des mains ou pour évaluer les fonctions exécutives (Test de *trail-making*). Un des grands avantages du *Leap Motion* est de pouvoir générer en continu des données sur les mains en mouvement dans l'espace.

3.1.2. Analyse de la précision, de la fiabilité du capteur Leap Motion et son aptitude pour le suivi statique, dynamique

“An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking” [Guna et al, 2014]¹⁴

Le but de cette recherche est d'analyser la précision, la fiabilité du *Leap Motion* en conditions statique, dynamique et de déterminer son aptitude comme capteur économique de suivi des doigts, de la main, d'un objet. L'évaluation est réalisée à l'aide d'un système de capture optique de grande vitesse et de grande précision.

Appareillage : le contrôleur *Leap Motion* est placé sur une table de 60 cm par 60 cm à une hauteur de 73 cm et fermement attaché à celle-ci. Le *Leap Motion* transmet les données sur les objets à un ordinateur de bureau (Intel Core i7). Une série de scripts écrits en Python utilisant l'API du *Leap Motion* gèrent les données d'acquisition en temps réel ainsi que l'enregistrement. Le visionneur du *Leap Motion* contrôle le test en temps réel. Un système optique est calibré pour des mesures portant sur un volume d'un mètre cube avec une résolution de 1,3 million de pixels et une fréquence d'échantillonnage de 500 cadres par secondes. Les caméras encerclent le *Leap Motion*. Une série de marqueurs de diamètre 12,5 mm sont utilisés pour les mesures.

Les mesures en condition statique consistent à acquérir un nombre limité de points dans l'espace et en condition dynamique à suivre le mouvement d'objets ayant une inter-distance constante dans l'espace calibré. Il est utilisé pour les mesures statiques, un bras en plastique simulant les mouvements du bras humain et pour les mesures dynamiques un outil en forme de V simulant deux doigts humains.

Dans le scénario statique, la déviation standard fut inférieure à 0,5 mm pour tous les temps et dans les meilleures conditions, inférieure à 0,01 mm. La déviation standard est d'autant plus importante que l'on s'éloigne du *Leap Motion*. Le *Leap Motion* est un appareil précis et fiable pour le suivi de la position de points statiques. Le *Leap Motion* ne peut suivre les points statiques que quelques secondes puis stoppe par manque d'environnement stable. Les meilleures conditions de mesures sont celles pour lesquelles les mains sont entièrement visibles dans l'espace de sensibilité du *Leap Motion*.

Le scénario dynamique révèle une faiblesse de performance et une diminution importante de précision au-delà d'une distance supérieure à 250 mm du *Leap Motion*. Une limitation de performance importante est l'inconsistance de fréquence d'échantillonnage. La valeur moyenne de cette fréquence est inférieure à 40 Hz donc relativement basse. Elle varie significativement en conditions statique et dynamique.

En conclusion, le *Leap Motion* de par son espace de sensibilisation limité et de par l'inconsistance de sa fréquence d'échantillonnage est inutilisable comme système professionnel de suivi des mouvements.

3.1.3 Analyse de la précision et de la robustesse du contrôleur Leap Motion

“Analysis of the accuracy and robustness of the leap motion controller” [Weicher et al, 2013]¹⁵

Le but de l'expérience est d'analyser les capacités du *Leap Motion* en précision et en fiabilité, de spécifier un test objectif pour les capteurs 3D en utilisant un robot industriel et de définir un critère de qualité considérant des spécifications de nature industrielle.

Appareillage : le contrôleur *Leap Motion* est placé sur une table dans l'espace de contrôle d'un robot industriel *Kuka* muni d'un stylo de référence. Les systèmes de coordonnées du *Leap Motion* et du robot coïncident. Un ordinateur portable muni du logiciel d'évaluation sert d'appareil de contrôle entre le *Leap Motion* et le robot. Une région d'intérêt correspondant à un volume où les mesures statiques et dynamiques seront prises est définie. Dans le but de tester l'influence dimensionnelle des mesures, le diamètre du stylo de référence variera.

La précision est la capacité du capteur 3D de déterminer une position donnée dans l'espace 3D. La fiabilité du capteur 3D est de localiser la même position à chaque répétition de mesure.

Le scénario statique permet la mesure d'une déviation maximum d'une position connue au cours du temps. Le scénario dynamique est utilisé pour tester la capacité du capteur 3D d'acquérir une position précise d'un objet en mouvement.

En sachant que la précision de position de la main humaine se situe aux alentours de 0,4 mm, la précision expérimentale doit se trouver dans le même ordre de grandeur car le *Leap Motion* doit reconnaître des gestes de la main. Le robot de référence choisi possède un stylo de référence d'une précision de 0,2 mm (en-dessous de la précision due à un mouvement involontaire de la main humaine).

En condition statique, une déviation entre la position désirée et la position mesurée inférieure à 0,2 mm a été obtenue. Cette constatation se vérifie indépendamment de l'axe sur lequel se fait la mesure. Il n'y a pas d'influence du diamètre du stylo sur la précision de la mesure. En condition dynamique, indépendamment du plan de mesure, une précision inférieure à 2,5 mm est obtenue (moyenne 1,2 mm).

La fiabilité a une moyenne inférieure à 0,17 mm. Lors d'un déplacement sur des positions discrètes, la déviation standard est de 0,7 mm en moyenne.

Il est possible d'améliorer le champ des interactions homme-machine par le développement d'applications pour le *Leap Motion*.

3.2 Evaluation du contrôleur Leap Motion comme appareil de pointage

Le but de cette évaluation est de comparer les capacités de pointage sans contact du *Leap Motion* par rapport à celle d'une souris standard.

3.2.1 Evaluation du contrôleur Leap Motion comme nouvel appareil de pointage sans contact

“Evaluation of the leap motion controller as a new contact-free pointing device” [Bachmann et al, 2014]¹⁶

L'évaluation consiste en une analyse des performances utilisateur d'une sélection de tâches avec le *Leap Motion* comparée à celle réalisée avec une souris standard.

Participants : six hommes d'âge moyen 25,25 ans participent à une séance d'une heure trente.

Appareillage : un logiciel écrit en C++ est utilisé pour présenter les tâches et enregistrer les trajectoires des participants, une souris standard *Logitech* et le contrôleur *Leap Motion*. Le mouvement d'accélération de la souris correspond à l'ordre de grandeur de celui du *Leap Motion* soit un mouvement horizontal de 300 mm. L'écran se trouve à 70 cm du participant et est disposé verticalement.

Procédure : les cibles sélectionnées sont en vert. Après la phase d'apprentissage, les participants les pointent à l'aide des deux appareils. Le geste *Key Tab* du *Leap Motion* a été choisi comme geste de pointage équivalent à un clic de souris. Ce geste est reconnu si l'extrémité du doigt fait une rotation vers le bas et revient ensuite à sa position initiale à une vitesse d'au moins 40 mm/sec. La norme ISO 9241-9 est respectée.

Résultats : évaluation des performances utilisateur soient le taux d'erreur, le temps de mouvement, le débit, le niveau d'effort mental, le mouvement de trajectoire.

Le taux d'erreur du *Leap Motion* est très supérieur (7,8%) à celui d'une souris classique (2,8 %). Plus de degré de liberté doivent être contrôlés dans le cas du *Leap Motion* par rapport à ceux de la souris. Il est possible que la performance puisse souffrir d'une charge additionnelle. Plus de pratique peut atténuer le taux d'erreur entre le *Leap Motion* et la souris.

Le *Leap Motion* comme appareil de pointage journalier a des performances limitées par rapport à une souris classique ce qui n'enlève rien à ses possibilités de reconnaissance de gestes.

3.2.2 Evaluation de tâche de pointage par le Leap Motion en environnement virtuel 3D

“Pointing task evaluation of leap motion controller in 3d virtual environment” [Coelho et al, 2014]¹⁷

Dans l'ensemble des périphériques pouvant interagir en environnement virtuel 3D, le *Leap Motion* a été choisi grâce à sa portabilité, sa précision, sa facilité d'utilisation et sa possible popularité pour la manipulation d'objets virtuels.

L'étude est basée principalement sur la translation et la déformation d'objet. Six aspects de convivialité seront analysés : la vitesse, la facilité d'apprentissage, la fatigue après utilisation, la coordination et la persistance.

Expérience : deux types de tâche seront réalisées avec la souris et avec le *Leap Motion*. Les différents aspects repris ci-dessus seront pris en compte. La première tâche est une entrée 3D pour un pointage initial. La deuxième tâche est un accès à l'aide du bouton de déroulement de la souris en 2D et un accès 3D avec le *Leap Motion*.

La précision est toujours importante. Dans un environnement virtuel dynamique, le temps de réalisation peut compromettre fortement la performance, c'est le cas pour des tâches multiples. Une tâche simple est facile à exécuter par les testeurs.

Le *Leap Motion* est plus performant que la souris pour réaliser des tâches de pointage en environnement 3D. La souris comme périphérique d'entrée est valable uniquement pour des tâches simples.

3.3 Interaction 3D en utilisant le contrôleur Leap Motion

L'évaluation d'une interaction gestuelle 3D permet d'estimer les capacités du *Leap Motion* quand il réagit avec des objets 3D virtuels.

3.3.1 Exploration de l'interaction 3D en utilisant le contrôleur Leap Motion

“Exploring direct 3d interaction for full horizontal parallax light field displays using leap motion controller” [Adhikarla, et al 2015]¹⁸

Le but de l'étude est la conception et l'évaluation d'une interaction gestuelle 3D avec un champ de lumière horizontal. Un champ de vision de lumière définit une scène visuelle en utilisant des faisceaux de lumière émis de multiples sources depuis la scène. Chaque point de la scène est rendu différemment des autres, il en résulte une vision plus réaliste. Le *Leap Motion* a été choisi pour explorer les avantages de l'affichage du champ lumineux pour son interaction sans les mains et pour l'expérience utilisateur procurée. L'ensemble procure une interaction haptique directe avec les objets 3D virtuels rendus par le champ de lumière.

Expérience : un ordinateur de contrôle fourni une sortie de l'image en 2D sur un écran LCD et une sortie modifiée affichant le champ de lumière. Les données d'interaction de l'utilisateur sont transmises à cet ordinateur par le *Leap Motion*.

Evaluation : le champ d'affichage lumineux et le concept d'interaction sont appréciés suivant le temps de complétude de la tâche (temps entre le toucher et le moment d'interaction), la charge cognitive à l'aide du questionnaire NASA TLX (*Task Load Index*) et la perception de l'expérience utilisateur à l'aide du questionnaire UEQ (*User Experience Questionnaire*).

Le champ d'affichage autorise une interaction 3D plus précise comparée aux autres technologies de rendu. L'interaction est plus intuitive, plus naturelle.

Une procédure de calibration est nécessaire. Le *Leap Motion* produit parfois des lectures anormales comme la position d'un doigt qui aurait bougé entraînant l'impossibilité de reconnaissance du geste.

3.4 Comparaison entre le Leap Motion et le Microsoft Kinect

Les capteurs *Leap Motion* et *Microsoft Kinect* peuvent être utilisés simultanément pour augmenter la précision de reconnaissance. Ils sont, de ce fait, complémentaires.

3.4.1 Evaluation comparative du Leap Motion et du Microsoft Kinect

La latence de bout en bout du *Leap Motion* est bien plus faible (85 ms) que celle du *Microsoft Kinect* (170 ms) [Brown et al, 2014]¹⁹. Cette latence correspond au temps mis entre une action de l'utilisateur et le retour à l'écran correspondant.

3.4.2 Reconnaissance des gestes de la main avec le Leap Motion et le Kinect

“Hand gesture recognition with leap motion and kinect devices” [Marin et al, 2014]²⁰

Le but de cette étude est de comparer les deux appareils pour la reconnaissance des gestes de l'alphabet. La reconnaissance des gestes se fait à l'aide d'un algorithme à classeurs multiples (*Support Vector Machine*) qui extrait les caractéristiques. Pendant une phase d'entraînement, les données récoltées par l'algorithme identifient une série de gestes.

Le *Leap Motion* contrôle trois sujets : le centre, l'orientation de la main et la position de l'extrémité des doigts. Les paramètres de distance, d'élévation et d'angulation en sont déduits et traités par l'algorithme.

A partir de données de couleur et de profondeur le *Kinect* extrait les éléments de la main : les doigts, la paume. Les paramètres traités par l'algorithme sont la courbure et la corrélation. Le *Leap Motion* produit un nombre limité d'informations par rapport au *Kinect* équipé d'une caméra de profondeur. Sa zone d'interaction est plus petite mais les données extraites sont plus précises que celles du *Kinect*.

Résultats : les gestes sont acquis en même temps par le *Leap Motion* et le *Kinect*. La précision d'ensemble pour le *Leap Motion* suivant les paramètres de distance, d'élévation et d'angulation est de 80,86 %. La précision d'ensemble pour le *Kinect* suivant les paramètres de courbure et de corrélation est de 89,71%. En regroupant les deux appareils, une précision plus importante de 91,28 % que chacun d'eux pris isolément est obtenue. La bonne précision obtenue en combinant les deux appareils autorise l'analyse des gestes dynamiquement. Certains doigts ne sont cependant pas détectés à l'aide du *Leap Motion* ce qui diminue sa fiabilité.

3.5 Amélioration des facultés de reconnaissance de gestes du Leap Motion

Il est possible d'améliorer les capacités de reconnaissance du contrôleur *Leap Motion* par :

- une interpolation de trajectoire d'un mouvement 3D sur base de données d'une trajectoire initiale ;
- la modélisation du squelette de la main qui fournit un suivi plus précis ;
- un algorithme qui approxime un contour d'image en une série de segments ;
- une intégration d'autres capteurs qui permettent de percevoir d'autres types de signaux.

3.5.1 Le Leap Motion pour la capture de mouvement 3D par spline

« Leap Motion pour la capture de mouvement 3D par spline L1 » [Hernoux et al, 2013]²¹

Le *Leap Motion* est utilisé pour l'apprentissage par un système robotisé UR10 de la gestuelle d'un opérateur permettant la reproduction de tâches complexes en 3D.

Le SDK (*Software Development Kit*) du *Leap Motion* permet de récupérer les positions et l'orientation des mains, des doigts. Ces paramètres permettent de composer un vecteur de six valeurs qui sera envoyé au robot en TCP/IP pour le commander. Cependant, le protocole d'acquisition de ces données ne permet pas d'assurer un échantillonnage temporel constant. Même à vitesse presque constante des zones sont dépourvues de mesures.

Afin de permettre la reproduction du mouvement par le robot, il est nécessaire d'interpoler les données mesurées en conservant la trajectoire initialement décrite. Pour effectuer l'interpolation, il est proposé d'utiliser des *splines* minimisant la norme L1 de leur dérivée seconde qui ont comme propriété de préserver la trajectoire issue des données.

3.5.2 Modélisation du squelette de la main

“Markerless Hand Gesture Interface Based on LEAP Motion Controller” [Avola et al, 2014]²²

Il est utile de développer une interface de reconnaissance des gestes de la main pour interagir avec de nombreuses applications. L'API du contrôleur *Leap Motion* produit un modèle du squelette de la main qui fournit des données de suivi précises. Ces données seront analysées pour reconnaître les gestes par un algorithme qui interprète les mouvements du centre de la main et des doigts. Les étapes de la reconnaissance sont les suivantes :

- prétraitement : projections 2D suivant les plans (x, y), (x, z) et (y, z) de la courbe 3D de l'objet suivi temporellement ;
- extraction des caractéristiques et reconnaissance du geste ;
- stockage du geste dans une librairie.

3.5.3 Utilisation d'un algorithme non paramétrique

“A Non-parametric RDP Algorithm Based on Leap Motion” [Zhan et al, 2016]²³

Le *Leap Motion* peut fournir des données à un algorithme de reconnaissance de gestes mais n'est pas

adapté pour les gestes complexes. L'utilisation d'un algorithme non paramétrique RDP permet d'augmenter la robustesse, de réduire le bruit du *Leap Motion* et de reconnaître plus de gestes. L'algorithme procède en réduisant le nombre de points d'un contour d'image en l'approximant à une série de segments.

3.5.4 Intégration d'autres capteurs pour percevoir d'autres types de signaux

“ConductHome: Gesture Interface Control of Home Automation Boxes” [Branstett et al, 2015]²⁴
Le logiciel *ConductHome* interagit avec les applications domotiques et fournit des informations utiles à l'utilisateur. Ces applications peuvent concerner la sécurité, l'efficacité énergétique, la communication et ont des interfaces différentes. L'interaction gestuelle se fera au travers d'une interface unique et conviviale.

Des gestes de base comme le balayage (*Hand Swipe*) ou le cercle (*Finger Circle*) facilement reconnaissables par le *Leap Motion* seront utilisés pour interagir avec les systèmes domotiques.

3.6 Synthèse

Les performances du *Leap Motion* sont évaluées au travers de trois qualités :

- la précision qui est la capacité du capteur de déterminer une position donnée dans l'espace ;
- la fiabilité qui est la capacité du capteur de localiser la même position à chaque répétition de mesure ;
- la robustesse qui est la capacité du capteur à conserver ses performances.

Précision : Les mesures en toucher de cible sont moins précises que sans toucher [Tung et al, 2015]²⁵. Le *Leap Motion* est un appareil précis pour le suivi de points statiques (déviations entre la position désirée et obtenue inférieure à 0,2 mm [Guna et al, 2014]²⁶ et [Weichert et al, 2013]²⁷). Pour le suivi de points dynamiques, la précision est moindre (déviations obtenues inférieures à 2,5 mm [Weichert et al, 2013]²⁸). La déviation standard tant pour le suivi statique que dynamique est plus importante au fur et à mesure que l'utilisateur s'éloigne du *Leap Motion* [Guna et al, 2014]²⁹. En suivi dynamique, une diminution importante de précision est constatée au-delà de 250 mm du *Leap Motion* [Guna et al, 2014]³⁰.

Fiabilité : Le *Leap Motion* est moins fiable en toucher de cible que sans toucher. Il est suffisamment fiable pour mesurer une tâche de pointage qui ne nécessite pas une haute précision de position [Tung et al, 2015]³¹. Le *Leap Motion* est un appareil fiable pour le suivi de points statiques [Guna et al, 2014]³². Certains doigts ne sont cependant pas détectés à l'aide du *Leap Motion* ce qui diminue sa fiabilité.

Robustesse : Les meilleures conditions de mesures sont celles pour lesquelles les mains sont entièrement visibles dans l'espace de sensibilité. Le *Leap Motion* ne peut suivre les points statiques que quelques secondes puis stoppe par manque d'environnement stable. Une limitation de performance importante est l'inconsistance de fréquence d'échantillonnage. Elle est relativement basse (inférieure à 40 Hz) et varie significativement en conditions statique et dynamique [Guna et al, 2014]³³. Il est possible que la performance puisse souffrir d'une charge additionnelle (plus de CPU utilisé).

Le *Leap Motion* de par son espace de sensibilisation limité et de par l'inconsistance de sa fréquence d'échantillonnage est inutilisable comme système professionnel de suivi des mouvements [Guna et al, 2014]³⁴.

Le *Leap Motion* est comparé à la souris comme appareil de pointage. Il en résulte qu'en environnement bidimensionnel le taux d'erreur pour des tâches de pointage avec le *Leap Motion* est très supérieur (7,8%) à celui d'une souris (2,8 %). Dans ces conditions, le *Leap Motion* est donc moins performant que la souris [Bachmann et al, 2014]³⁵. Par contre, pour des tâches de pointage en environnement tridimensionnel, le *Leap Motion* se révèle plus performant que la souris [Coelho, 2014].

Il est possible d'améliorer la précision de l'interaction en environnement tridimensionnel par l'utilisation d'un champ de lumière horizontal [Adhikarla et al, 2015]³⁶.

Le *Leap Motion* peut être utilisé avec le *Microsoft Kinect*. En regroupant les deux appareils, une précision plus importante de 91,28 % que chacun d'eux pris isolément est obtenue. La bonne précision obtenue en combinant les deux appareils autorise l'analyse des gestes dynamiquement [Marin et al, 2014]³⁷.

Il est possible d'améliorer les capacités de reconnaissance du contrôleur *Leap Motion* par :

- une interpolation de trajectoire d'un mouvement 3D sur base de données d'une trajectoire initiale [Hernoux et al, 2013]³⁸ ;
- la modélisation du squelette de la main qui fournit un suivi plus précis [Avola et al, 2014]³⁹ ;
- un algorithme qui approxime un contour d'image en une série de segments [Zhan et al, 2016]⁴⁰ ;
- une intégration d'autres capteurs qui permettent de percevoir d'autres types de signaux [Branstett et al, 2015]⁴¹.

4. Etat de l'art des applications du Leap Motion

4.1 Compte-rendu des applications possibles avec le Leap Motion

Le but de ce chapitre est de fournir un état de l'art des applications du contrôleur *Leap Motion* à partir d'une quarantaine de références. Les applications sont répertoriées par thème.

La méthodologie de recherche des références est la suivante :

Les articles concernant le *Leap Motion* sont principalement écrits en anglais ce qui implique l'emploi de mots clé de recherche en anglais. La recherche s'effectue en grande partie par le moteur *Google scholar* qui permet de trouver des articles scientifiques. La fréquence de citation élevée sur *Google scholar* constitue un bon critère de choix d'une référence de même que la date de parution récente.

Quelques domaines d'application sont renseignés sur le site du constructeur, la recherche commence par ces mots clé. Ensuite, au moment où un article est trouvé dans un domaine, un second est sélectionné avec une fréquence un peu inférieure tout en gardant une date de parution récente.

Il est possible de découvrir des articles à l'aide de mots clés spécifiques : *Leap Motion nime* pour le domaine musical, *Leap Motion uist* pour les technologies et interactions, *Leap Motion ICMI* pour le domaine multimodal, *Leap Motion CSCW* pour les systèmes collaboratifs.

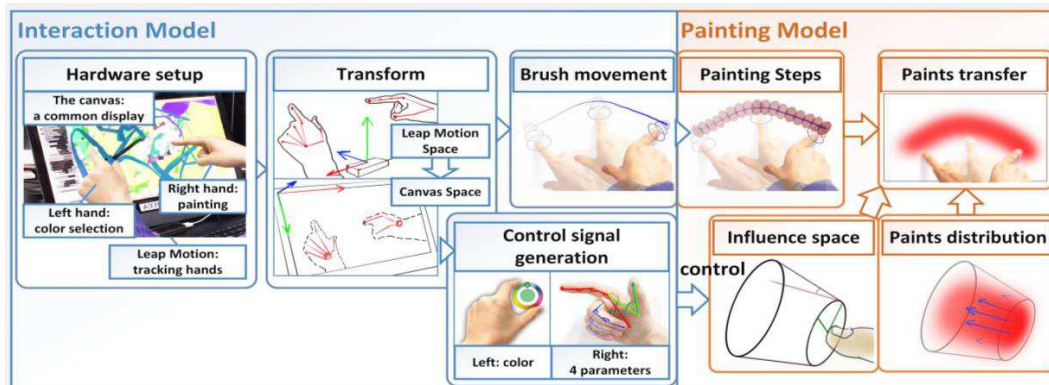
Les principaux thèmes identifiés sont les suivants : peinture, contrôle de systèmes domotiques, contrôle d'un robot chargé des tâches domestiques, réhabilitation de personnes, imagerie d'opérations chirurgicales, contrôle par le regard et les gestes, langages des signes, réalité virtuelle, applications musicales, applications industrielles, éducation et authentification biométrique.

Pour chaque thème et pour chaque référence, l'application est décrite ainsi que le logiciel de reconnaissance et le(s) logiciel(s) complémentaire(s). Dans le cas où elles sont disponibles, les informations suivantes sont résumées : les résultats des principaux tests et la manière de les mener, le type de geste concerné, l'utilisation d'autres capteurs.

4.1.1 Peinture

“A flexible finger-mounted airbrush model for immersive freehand painting” [Lyu et al, 2017]⁴²

L'utilisateur crée différents effets avec des techniques d'interaction flexibles et adaptables.



Peinture à l'aide d'un aérographe virtuel

Le modèle conceptuel inclut deux sous-modèles :

- un modèle d'interaction comprenant quatre parties : réglage du *Leap Motion* et de l'écran, transformation des gestes de la main du repère de coordonnées du *Leap Motion* vers celui de l'écran, contrôle de mouvement de la brosse, contrôle de la couleur et des paramètres de peinture ;
- Un modèle peinture.

Lorsque l'aérographe virtuel intercepte l'écran, il se place en mode peinture jusqu'au moment où il sort de l'espace d'interaction.

Jeremy Sutton réalise de la peinture dans l'air à l'aide du *Leap Motion* et du logiciel (*Corel painter freestyle*) [Sutton, 2013]⁴³. Le mouvement des mains dans l'espace introduit une nouvelle sensation de peinture plus proche de la sculpture. Il est possible d'utiliser un contrôle de pression disponible sur certaines tablettes pour le contraste des couleurs.

4.1.2 Contrôle de systèmes domotiques

“ConductHome: Gesture Interface Control of Home Automation Boxes” [Branstett et al, 2015]⁴⁴

Le logiciel *ConductHome* interagit avec les applications domotiques et fournit des informations utiles à l'utilisateur. Ces applications peuvent concerner la sécurité, l'efficacité énergétique, la communication et possèdent des interfaces différentes.

Des gestes de base comme le balayage (*Hand Swipe*) ou le cercle (*Finger Circle*) facilement reconnaissables par le *Leap Motion* seront utilisés pour interagir avec les systèmes domotiques. L'interaction gestuelle se fait au travers d'une interface unique et conviviale qui peut venir en aide aux personnes âgées ou malades.

4.1.3 Commande par les gestes d'un robot chargé des tâches domestiques

Le *Leap Motion* peut être utilisé pour commander par les gestes un robot chargé des tâches domestiques.

Dans un premier cas [Georgoulas et al, 2014]⁴⁵, un bras robotisé à six degrés de liberté commandé par les gestes est placé à différents endroits de la maison : salle de bain, chambre, chaise roulante afin d'aider les personnes âgées ou malades dans leurs activités journalières. Les données du *Leap Motion* sont acheminées via le port USB d'un mini-PC situé sur un dispositif pouvant se déplacer par commande gestuelle vers les différents endroits, sont traitées puis commandent par wifi le bras robotisé.

Dans un deuxième cas [Yu et al, 2015]⁴⁶, un robot humanoïde *NAO* commandé par les gestes reconnus par le *Leap Motion* est utilisé pour exécuter divers services dans la maison. L'ordinateur de contrôle reçoit les données des gestes de l'utilisateur par la connexion USB du *Leap Motion* et envoie les commandes au robot par wifi. L'ordinateur est muni du logiciel SDK en Python, d'un algorithme de cartographie des données et d'un module de commande. Si la main droite se trouve dans le champ de vision du *Leap Motion*, des gestes prédéfinis sont reconnus et enregistrés. Si la main gauche se retrouve dans la zone opérationnelle du *Leap Motion*, les positions cartésiennes et les orientations sont enregistrées.

4.1.4 Contrôle d'un fauteuil roulant

“Hand posture control of a robotic wheelchair using a leap motion sensor and block sparse representation based classification” [Bojali, 2014]⁴⁷

Les postures de la main gauche sont capturées par le *Leap Motion* en collectant la rotation et la direction de la paume et des angles d'inclinaison puis évaluées par l'algorithme de classification BS-SRC. Cet algorithme convient bien pour cette application en temps réel.

Il y a cinq gestes de la main gauche à reconnaître : aller de l'avant, tourner à droite, tourner à gauche, arrêter, faire marche arrière. L'approche proposée est facile à implémenter, très précise et n'apporte pas de limitation du mouvement de la main suivie. Elle est robuste aux conditions d'éclairage et n'entraîne pas d'erreur de classification.

Le système possède plusieurs ordinateurs portables. Le premier ordinateur portable traite de la reconnaissance des gestes. Les commandes reconnues sont envoyées par câble réseau au deuxième ordinateur portable principal. Cet ordinateur principal s'occupe d'éviter les obstacles et contient les modules de déplacement autonomes. Enfin, l'ordinateur principal envoie un signal à l'ordinateur de la chaise qui commande les déplacements par une liaison RS232.

Le *Leap Motion* est situé sur la partie droite de la chaise et suit les objets à une fréquence de 80 Hz. Le volume de détection effectif est une pyramide inversée de hauteur 47 cm dont le sommet est le centre du *Leap Motion*. L'algorithme de classification BS-SRC est écrit en MATLAB et une interface spécifique est utilisée pour analyser les données du *Leap Motion* dans ce langage.



Fauteuil roulant contrôlé par un capteur *Leap Motion*

4.1.5 Réhabilitation de personne

La réhabilitation de personnes ayant perdu l'usage de leur(s) main(s) les aide à le recouvrir. La façon la plus intuitive et naturelle est d'exécuter une interaction sans les mains. Elle peut être réalisée par le jeu à l'aide du *Leap Motion* et conduit à une réhabilitation rapide.

“Free-hand interaction with leap motion controller for stroke rehabilitation” [Khademi et al, 2014]⁴⁸

Après avoir subi un accident vasculaire cérébral, les personnes peuvent avoir perdu partiellement ou complètement l'usage d'une ou de deux mains. Les aider à retrouver cet usage peut améliorer significativement leur qualité de vie. La façon la plus intuitive et la plus naturelle de réhabiliter ces personnes est de pratiquer une interaction sans les mains. La réhabilitation la plus rapide et la plus efficace se fait par le jeu car le patient ne se sent pas contraint et peut réaliser un progrès.

Le système de réhabilitation : il combine le jeu *Fruit Ninja* avec le contrôleur *Leap Motion* en utilisant une version en JavaScript de ce jeu. Le *Leap Motion* remplace la souris pour interagir dans le jeu à l'aide de la main.

L'étude porte sur quatorze patients victimes d'un accident vasculaire cérébral. Le Jeu *Fruit Ninja* a été adapté pour que ces patients d'handicap différent puissent jouer une minute. Pour les patients qui ont des difficultés à se servir d'un doigt, un outil porté par la main le remplace. L'expérience montre une forte corrélation entre de bons scores au jeu et de bons résultats au test clinique. Un modèle cinématique de l'entièreté de la main permet de disposer d'un système de suivi plus précis et robuste que celui du *Leap Motion* qui ne reprend que la position et la direction des doigts.

“Application for hand rehabilitation using leap motion sensor based on a gamification approach” [Karashanov et al, 2016]⁴⁹

L'idée de base est de développer le tonus musculaire en augmentant la précision des gestes en rendant le processus de réhabilitation plaisant et stimulant.

Le *Leap Motion* identifie les doigts et les paumes des mains avec une grande précision et autorise les patients à être assis ou en position couchée. Le *Leap Motion* est facilement transportable et positionnable de sorte qu'il peut servir pour différents patients et en différents endroits. Il est peu coûteux et peut être utilisé avec tous les systèmes d'exploitation. Il ne nécessite pas de synchronisation des gestes et donc peut servir pour des patients de tout âge et de tout genre.

Expérience : les patients sont confrontés à plusieurs jeux. Ils devront prendre différents objets pour les placer dans une poubelle, rassembler des objets pour confectionner une pyramide et se déplacer dans un labyrinthe.

Le retour d'expérience est encourageant dans le sens où le système peut être utilisé à domicile à des fins de réhabilitation.

“A review on technical and clinical impact of Microsoft Kinect on physical therapy and rehabilitation” [Khademi et al, 2014]⁵⁰

Pour le suivi de la main avec individualisation des doigts et pour le suivi en espace rapproché, le *Leap Motion* convient mieux que le *Kinect* de Microsoft. Par contre, quand le suivi de l'ensemble du corps est requis, le *Kinect* ou l'*Asus Xtion* sont plus appropriés.

“3D serious games for Parkinson's disease management” [Blažica et al, 2016]⁵¹

Le système non intrusif vise à entraîner les membres supérieurs d'un patient atteint de la maladie de Parkinson à l'aide du *Kinect* et la motricité des doigts par de petits gestes à l'aide du *Leap Motion*.

4.1.6 Contrôle d'images en chirurgie

“System for touchless interaction with medical images in surgery using Leap Motion” [Manolova, 2014]⁵²

Le but du système est de pouvoir maintenir un environnement stérile pendant l'opération tout en ayant un contrôle direct sur les images pré et intra-opératoires en les manipulant et en les sélectionnant.

Le système proposé contient un contrôleur de gestes *Leap Motion* et une interface *Open Source* de visualisation des données médicales. Une librairie de reconnaissance de gestes *Open Source* est ensuite constituée.

Le *Leap Motion* reconnaît les éléments suivants : les mains, les doigts et des outils mais aussi les gestes fait par ces éléments. Le *Leap Motion* dispose d'une zone d'interaction de toucher où les utilisateurs pourront manipuler, sélectionner et ajuster les paramètres des images.

Par rapport au *Microsoft Kinect* qui a besoin d'un espace d'interaction volumineux (18 m³), le *Leap Motion* peut agir dans un espace plus réduit (0,23 m³).

“Leap motion gesture control with osirix in the operating room to control imaging first experiences during live surgery” [Bizzotto et al, 2014]⁵³

Osirix est un logiciel d'archivage d'images et un système de communication. L'association du *Leap Motion* et d'*Osirix* pendant l'opération pour contrôler les images par les gestes s'avère intéressante afin d'éviter les risques d'infection et est de coût relativement réduit.



“Use of a gesture user interface as a touchless image navigation system in dental surgery: Case series report” [Rosa et al, 2014]⁵⁴

L'interaction gestuelle par le *Leap Motion* se fait au niveau du fauteuil dentaire. L'ordinateur principal se trouve en zone non stérile. Il y a deux écrans un en zone stérile et un autre près de l'ordinateur principal. Le système permet la navigation dans les images et garantit les conditions de stérilité.

4.1.7 Contrôle aérien

“Interactions gestuelles pour StripTIC, un environnement tangible pour les contrôleurs aériens” [Gauthier et al, 2014]⁵⁵

Les contrôleurs aériens utilisent des bandes de papier correspondant à un avion avec ses informations de vol.

Ces bandes de papier sont soumises à plusieurs scénarios de manipulation gestuelle :

- sortie de la bande papier par le contrôleur de sa colonne d'attente avant atterrissage pour être pris en charge par un autre contrôleur ;
- particularisation d'un avion en utilisant l'indentation c'est-à-dire un décalage horizontal de la bande de papier par rapport aux autres ;
- sortie de la bande de papier dans un conteneur ; ce qui signifie que l'avion n'est plus sous la responsabilité du contrôleur ;
- transfert d'un avion au secteur suivant en transmettant la bande de papier à un autre contrôleur et en indiquant verbalement les caractéristiques de vol.

Ces bandes de papier sont remplacées par des objets virtuels qui seront ensuite manipulés comme des objets réels. Les interactions prises en compte permettent d'enrichir les interactions des objets virtuels par des gestes, de les dupliquer et d'interagir avec le système.

Implémentation : le Microsoft Kinect couplé au logiciel de reconnaissance de gestes 3-Gear et le Leap Motion pour la détection des gestes en bordure du système. Le capteur Kinect est placé au-dessus des bras et des mains de l'utilisateur et couvre l'ensemble de la surface du tableau récapitulatif des vols.

4.1.8 Contrôle par les gestes et le regard

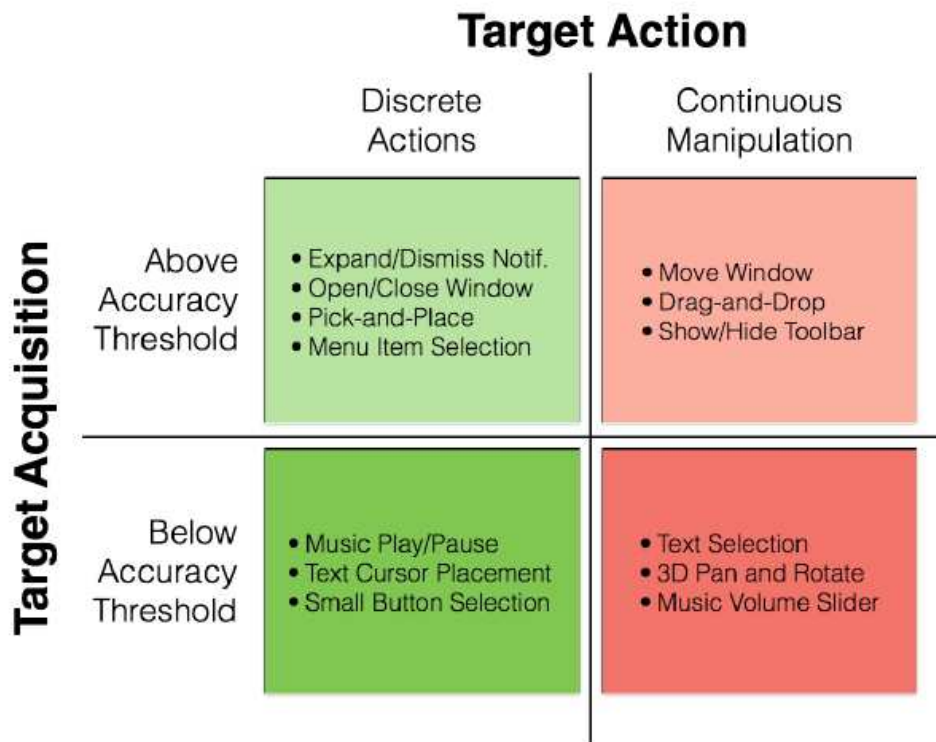
“Gaze + gesture: Expressive, precise and targeted free-space interactions” [Chatterjee et al, 2015]⁵⁶

Le regard est bien adapté pour être rapide et précis pour le pointage d'objets mais souffre d'un manque d'expressivité. Les gestes sont lents et imprécis pour le pointage mais peuvent exprimer une variété de fonctions interactives. Ces deux modes sont donc complémentaires.

Le système utilisé est peu coûteux et peu invasif. Il est composé d'un dispositif de suivi du regard positionné sur un trépied à 55 cm de l'utilisateur du *Leap Motion* pour assurer le suivi des gestes de la main droite et d'un ordinateur de contrôle muni du logiciel de reconnaissance de gestes et du regard.

Afin de caractériser les techniques d'interaction, nous distinguons une phase d'acquisition et une phase d'action sur la cible.

Comme nous le montre le schéma ci-dessous, la phase d'acquisition peut demander une précision grossière ou une précision fine et la phase d'action peut comporter des actions discrètes ou une manipulation continue. Ces subdivisions entraînent une taxonomie des interactions regard avec gestes suivant le cadran dans lequel on se situe.



Au titre d'exemples : ouvrir et fermer une fenêtre correspond au premier cadran avec une précision grossière et des actions discrètes, bouger la fenêtre au deuxième cadran avec une précision grossière et une manipulation continue, placer un curseur de texte correspond au troisième cadran avec une précision fine et des actions discrètes, sélectionner du texte correspond au quatrième cadran avec une précision fine et une manipulation continue.

Actuellement, les systèmes qui utilisent les gestes et le regard pour interférer sont rares mais peuvent être amenés à se développer afin de réaliser des interactions sans le toucher plus efficaces.

4.1.9 Langage des signes

“Investigating the dexterity of multi-finger input for mid-air text entry” [Sridhar et al, 2015]⁵⁷

Le Leap Motion est utilisé pour reconnaître les mouvements des doigts. Un signe des doigts est identifié pour correspondre à une lettre. L'ensemble des lettres reconnues constitue un mot.

Les résultats de l'étude fournissent des informations sur la reconnaissance de chaque doigt et sur la coactivité avec les autres doigts. L'apprentissage des gestes est un obstacle pragmatique pour les entrées de mouvement de plusieurs doigts. Si le geste fait pour signifier une lettre demande trop d'apprentissage, il sera difficile à adopter.

Le *Leap Motion* n'a pas été fiable pour reconnaître certaines articulations des doigts. Les utilisateurs sont limités par leur vitesse d'entrée des signes : une vitesse trop élevée entraîne des erreurs de positionnement.

“A sign language to text converter using Leap motion” [Khan et al, 2016]⁵⁸

L'étude porte sur un prototype de conversion en texte de signes grâce au contrôleur *Leap Motion*. La conversion en texte de signes est utilisée par les personnes malentendantes ou sourdes afin de communiquer.

Trois techniques sont utilisées pour reconnaître vingt-six gestes statiques de la main correspondant aux vingt-six lettres de l'alphabet.

Ces techniques sont :

- la correspondance de modèle géométrique qui analyse et trouve de petites parties d'un modèle d'image ;
- les réseaux neuronaux artificiels (*ANN*) ;
la corrélation croisée : un modèle mathématique qui mesure la similitude de deux séries en fonction du décalage de l'une par rapport à l'autre.

Le prototype utilise l'API *WebSocket* du *Leap Motion* pour reconnaître les gestes. Le serveur *WebSocket* fournit des données de suivi sous forme de message *JSON* à l'application web. Ces données sont présentées comme un objet *JavaScript*. L'algorithme détecte si le *Leap Motion* est actif ou en attente.

Les gestes statiques correspondant aux vingt-six lettres de l'alphabet doivent être enregistrés puis stockés dans une librairie, ensuite, ils seront comparés aux gestes suivis par le *Leap Motion*. L'entrée des gestes statiques par le *Leap Motion* est plus aisée qu'un codage individuel des lettres.

Au point de vue des résultats, la technique de la correspondance de modèle géométrique est la plus précise avec un pourcentage de 52,56 % pour la reconnaissance de l'alphabet.

Ensuite, viennent la technique des réseaux neuronaux artificiels (*ANN*) avec 44,87 % et la technique de la corrélation croisée avec 35,90 %.

“American sign language recognition using leap motion sensor” [Chuan et al, 2014]⁵⁹

Comparé aux systèmes de reconnaissance des signes existants basés sur le *cyberglove* ou le *Microsoft Kinect*, celui utilisant le *Leap Motion* est plus abordable.

L'API du *Leap Motion* fournit les caractéristiques de position et de vitesse des doigts et de la paume de main.

Dans cette étude, deux algorithmes de reconnaissance ont été utilisés et comparés :

- la méthode des k plus proches voisins (*KNN*) qui retient la classe la plus représentée parmi les k sorties associées aux k entrées les plus proches de la nouvelle entrée x ;
- la méthode des machines à vecteur de support (*SVM*) qui détermine à quelle classe appartient un échantillon.

Les résultats nous montrent que la meilleure moyenne obtenue par la méthode *KNN* est trouvée pour k =7 et est de 72,78%. Quant à la méthode *SVM*, la meilleure moyenne obtenue est de 79,83 % avec un noyau gaussien.

Pour les deux méthodes, une série de lettre est mal reconnue.

“Arabic sign language recognition using leap motion sensor” [Elons et al, 2014]⁶⁰

Deux techniques de reconnaissance sont utilisées : une technique basée sur la classification naïve bayésienne (*Naive Bayes*) et l’autre sur un réseau de neurone artificiel (*Multilayer Perceptron MPL*).

Les deux techniques présentent des précisions de reconnaissance élevées : 98 % (*Naive Bayes*) et 99 % (*Multilayer Perceptron MPL*).

“The leap motion controller: a view on sign language” [Potter et al, 2013]⁶¹

L’utilisation du contrôleur *Leap Motion* présente les avantages suivants :

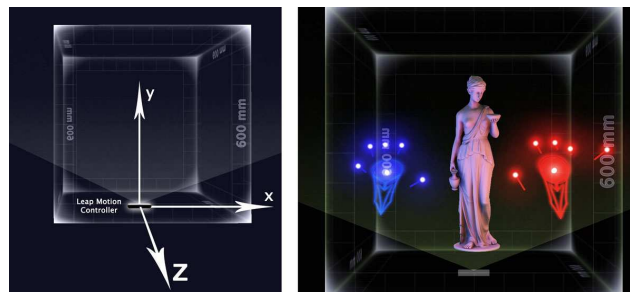
- l’API du *Leap Motion* réduit les erreurs et augmente la précision ;
- la granulométrie du *Leap Motion* est bien indiquée en reconnaissance de gestes ;
- le *Leap Motion* est capable de détecter de petits mouvements.

Par contre, il est difficile de maintenir la précision et la fiabilité quand les mains ne sont pas dans la ligne de mire du capteur et de détecter deux doigts qui soient l’un contre l’autre.

Le *Leap Motion* convient bien pour détecter des signes basiques mais pas pour des signes complexes en contact avec le visage ou le corps.

4.1.10 Réalité virtuelle

“A 3D interface to explore and manipulate multi-scale virtual scenes using the leap motion controller” [Fanini, 2014]⁶²



Une interface 3D est une interface qui permet l’interaction 3D. Une série de tests expérimentaux ont été réalisés afin d’en évaluer la convivialité. Il en ressort que le contrôleur *Leap Motion* permet, pour un prix abordable, une navigation et une manipulation d’un objet dans des environnements virtuels d’échelles différentes. Un procédé incluant des arrêts (*Stop*) et des glissements (*Drag*) permet d’augmenter la précision, de faciliter les transitions et de diminuer la fatigue. Un mode édition autorise de changer d’échelle par une manipulation des deux mains.

“A leap-supported, hybrid AR interface approach” [Regenbrecht et al, 2013]⁶³

La solution présentée est une nouvelle approche d’interface qui combine une vidéo 2D de réalité augmentée avec un modèle converti en pixel 3D, permettant des interactions plus convaincantes avec les objets 3D et leurs environnements.

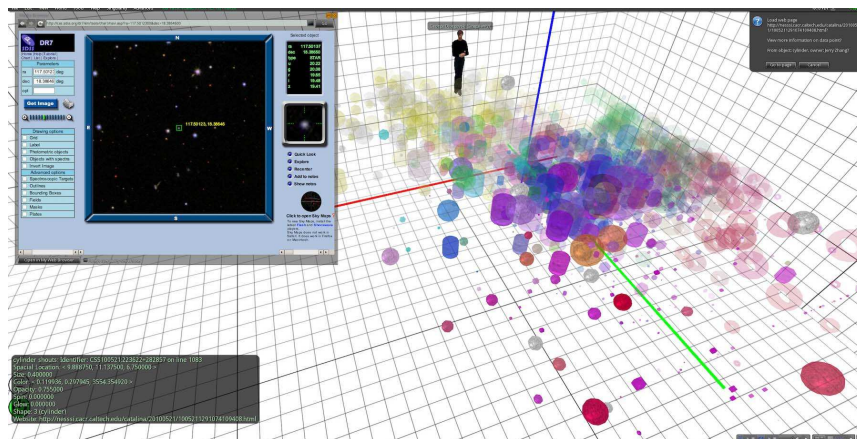
Le *Leap Motion* permet de résoudre le problème d'occlusion entre les doigts et le contenu virtuel. Il facilite les interactions d'objets virtuels.

“Immersive and collaborative data visualization using virtual reality platforms” [Donalek et al, 2014]⁶⁴

Le but de l'étude est de maximiser le modèle de reconnaissance humaine à travers un usage en réalité virtuelle.

L'outil de visualisation de données multidimensionnel apporte une solution simple, naturelle pour explorer et voir des données partagées dans un même espace. Cette solution permet aussi de mieux comprendre les données et de percevoir leurs relations.

L'application utilise le *Leap Motion* car il est portable, flexible et peu coûteux. Le logiciel développé est en *Unity 3D*.



4.1.11 Applications musicales

“Leap motion as expressive gestural interface” [Ritter et al, 2014]⁶⁵

Les systèmes informatiques performants de gestion de la musique utilisent fréquemment la reconnaissance de gestes.

Le *Leap Motion* permet la capture de petits mouvements spécifiques. Son utilisateur peut contrôler des paramètres visuels et musicaux. Comme instrument musical, le *Leap Motion* est polyvalent et expressif.

“Lessons Learned in Exploring the Leap Motion™ Sensor for Gesture-based Instrument Design” [Han et al, 2014]⁶⁶

On distingue les instruments musicaux augmentés et les quasi-instruments musicaux.

Les instruments musicaux augmentés sont des instruments complétés par des capteurs qui ajoutent la possibilité de contrôler des effets sonores. Un piano peut par exemple être contrôlé par le capteur *Leap Motion* pour combiner des sons acoustiques et électroniques.

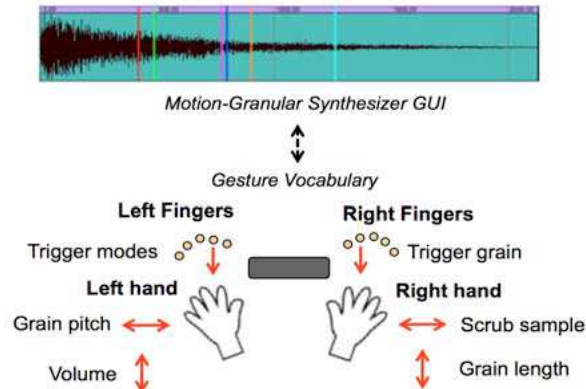
Les quasi-instruments musicaux sont des instruments virtuels contrôlés par les gestes. Un capteur *Leap Motion* peut contrôler un quasi-piano (*Air piano*).

Le sujet décrit deux instruments musicaux augmentés (les *Air Key* et les *Air Pad*) et analyse comment ils peuvent compléter un clavier traditionnel. Les *Air Key* sont utilisés pour recréer un clavier virtuel avec différentes notes paramétrables. Les *Air Pad* donnent différents effets sonores par création de boutons correspondant à des zones d'interaction.

“Implementations of the Leap Motion in sound synthesis, effects modulation and assistive performance tools” [Hantrakul et al, 2014]⁶⁷

Le but de l'étude est d'analyser les applications musicales du capteur *Leap Motion*.

L'application la plus connue du *Leap Motion* avec un logiciel de musique est *Geco*. L'utilisateur est capable grâce à une interface intuitive de transformer les gestes de la main en messages *MIDI*. Les gestes reconnus sont les mains ouvertes ou fermées sans contrôle individuel des doigts.



Grâce à l'API *beta*, il est possible de suivre distinctement la main gauche, la main droite, les différents doigts de chaque main. Il est donc possible de créer avec son aide un synthétiseur granulaire où des fragments de son peuvent être extraits et séquencés afin de créer de nouveaux effets sonores. Les commandes du synthétiseur utilisent les mains et les différents doigts.

Le *Leap Motion* peut également commander un objet hémisphérique composé de plusieurs haut-parleurs. Une interaction par le geste de la main peut agir individuellement sur chaque nœud de l'objet qui correspond à un haut-parleur spécifique pour ajuster son volume.

“The Composing Hand : Musical Creation with Leap Motion and the BigBang Rubette.” [Tormoen et al, 2014]⁶⁸

Le *Leap Motion* peut réinventer la composition musicale car il a une faible latence et une haute précision. Il procure beaucoup de paramètres qui répondent aux attentes des musiciens.

L'audience peut entendre en direct l'interaction entre les mouvements de l'interprète et les sons produits. Le *Leap Motion* permet la conception de sons par sélection directe des touches d'un clavier virtuel.

Le logiciel *BiGBang Rubette* permet de voir la représentation de la composition musicale comprenant tous les objets musicaux.

“Infrared vs. Ultrasonic Finger Detection on a Virtual Piano Keyboard.” [De Pra et al, 2014]⁶⁹

La détection infrarouge des doigts utilise le *Leap Motion*. Elle est continue et nécessite beaucoup de puissance CPU pour suivre chaque doigt. Les doigts ne sont pas toujours reconnus par le système spécialement quand les mouvements sont rapides.

“A preliminary evaluation of the leap motion sensor as controller of new digital musical instruments” [Silva et al, 2013]⁷⁰

L'étude montre que le *Leap Motion* peut apporter une bonne simulation tant que le rythme reste lent. A un rythme normal, il présente des problèmes de défaut de suivi, d'occlusion et une latence inacceptable avec le logiciel de développement actuel.

Le piano requiert une haute précision de mouvement des mains et n'est pas l'instrument idéal pour le suivi en profondeur par le capteur *Leap Motion*. Pour l'expérience utilisateur, un retour tactile est conseillé.

“Probabilistic models for designing motion and sound relationships” [Françoise et al, 2014]⁷¹

Les modèles sont classés suivant la multi modularité et la temporalité.

Les modèles multimodaux se distinguent des modèles de mouvement. Les modèles de mouvement synthétisent le son à partir d'exemples de mouvements. Les modèles multimodaux synthétisent le son à partir d'une génération de paramètres issus d'exemples de mouvements et de sons.

Les modèles temporels se distinguent des modèles instantanés. Les modèles instantanés travaillent avec des mouvements statiques, au contraire, les modèles temporels tiennent compte du changement du mouvement au cours du temps.

Le *Leap Motion* peut être utilisé dans un modèle de mouvement temporel (ensemble de gestes) ou dans un modèle multimodal instantané (modèle physique).

“Hand Gestures in Music Production” [Berndt et al, 2016]⁷²

L'utilisateur peut effectuer un ensemble de gestes codifiés pour reprendre les caractéristiques du son. Nous distinguons les gestes qui contrôlent :

- Le tempo (vitesse) ;
- La dynamique (volume) ;
- Le phrasé (lié ou piqué) ;
- Le timbre de l'instrument.

4.1.12 Applications industrielles

“Toward welding robot with human knowledge: A remotely-controlled approach” [Liu et al, 2015]⁷³

Un robot de soudure doit pouvoir moduler sa tâche en fonction de changements de production. Souvent, leur adaptabilité est limitée par rapport à celle de l'homme. L'étude présente un système qui permet au robot de soudure de suivre les gestes de l'homme pour moduler sa tâche.

Le mouvement de soudure de l'opérateur est suivi précisément par le contrôleur *Leap Motion* qui commande un robot de soudure à six degrés de liberté. Ce robot est pourvu de capteurs qui observent la pièce travaillée et reconstruisent la surface soudée en 3D.

“Using Integrated Vision Systems: Three Gears and Leap Motion, to Control a 3-finger Dexterous Gripper” [Zubrycki et al, 2014]⁷⁴

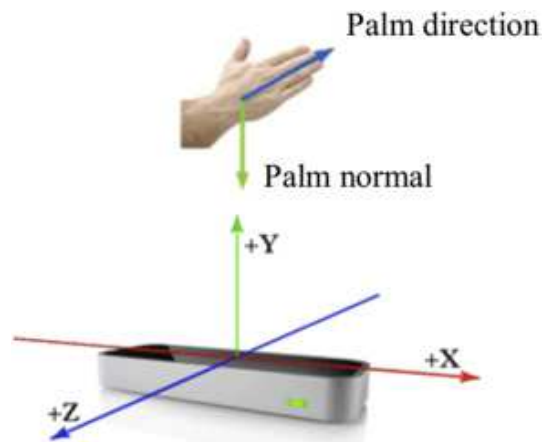
L'utilisation d'un *Leap Motion* contrôlant un grappin muni de trois doigts vient du besoin d'obtenir une commande plus intuitive. Par rapport à une commande à l'aide d'un gant sensitif, le *Leap Motion* n'a pas besoin d'un calibrage poussé et la main de l'utilisateur n'est pas mobilisée en permanence. La limitation principale est que l'utilisateur doit connaître toutes les fonctions dédiées à tous les gestes préalablement enregistrés.

“Multi-LeapMotion sensor based demonstration for robotic refine tabletop object manipulation task” [Jin et al, 2016]⁷⁵

Comparé aux dispositifs mécaniques, les systèmes de suivi de gestes sont moins chers et peuvent être installés en différents endroits. Le *Leap Motion* est un système de suivi qui possède, à priori, d'une grande précision et dispose de logiciels reconnaissant les postures et les gestes.

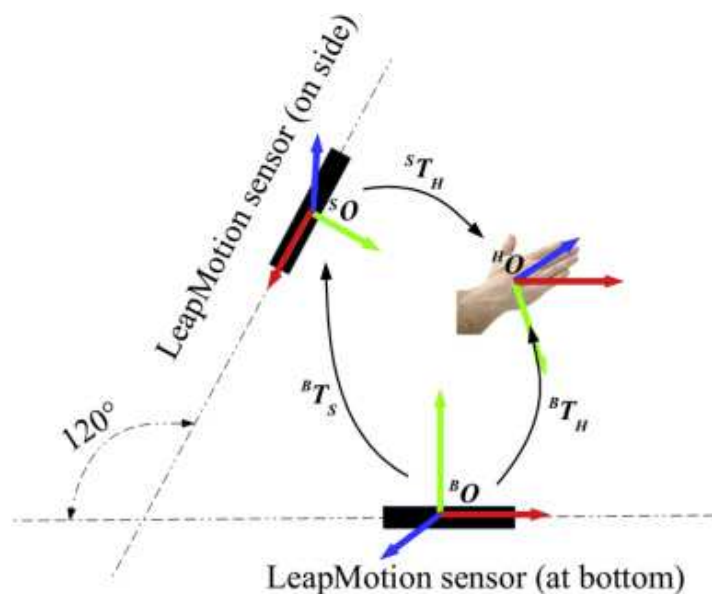
Un seul capteur identifie correctement les gestes quand la paume de la main se trouve dans une orientation idéale. L'utilisation de deux capteurs améliore la stabilité du suivi de la main et la reconnaissance de gestes.

Pour obtenir une grande précision de reconnaissance de geste avec un seul capteur *Leap Motion*, il faut que la paume de la main tourne de moins de 60° .



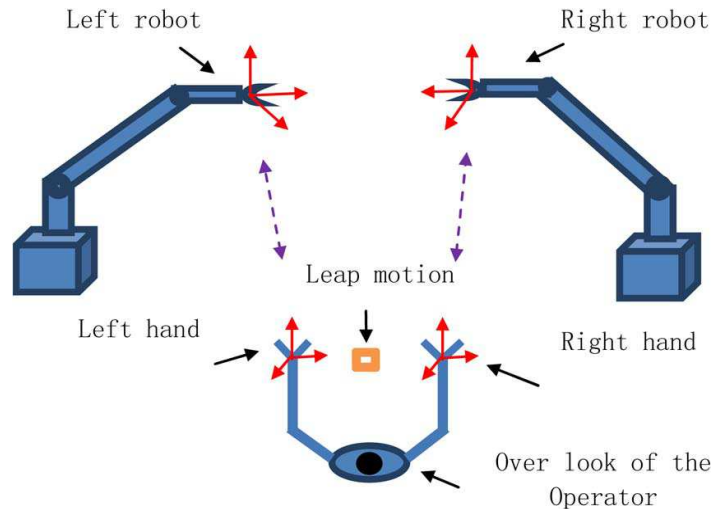
Des problèmes d'occlusion peuvent apparaître quand l'angle de rotation de la paume est proche de 90° (des doigts peuvent être occultés par d'autres doigts (*finger-occlusion*)) ou proche de 180° (des doigts sont occultés par la paume (*palm-occlusion*)).

Dans le cas de deux capteurs, la meilleure précision s'obtient quand les axes X des deux capteurs forment un angle de 120° . Dans cette configuration, les problèmes d'occlusion n'existent plus.



“A markerless human–robot interface using particle filter and Kalman filter for dual robots” [Du et al, 2015]⁷⁶

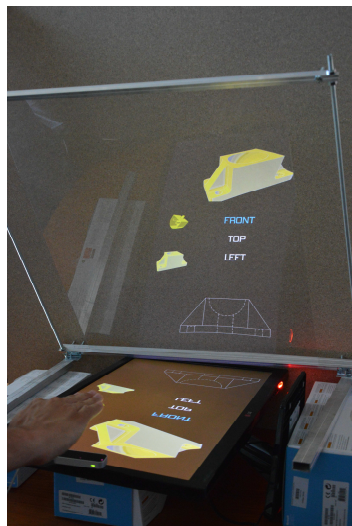
Deux robots sont commandés par interaction des mains. Le robot de gauche est commandé par la main gauche de l'opérateur et celui de droite par sa main droite. Chaque main est suivie par un capteur Leap Motion. Un filtre Kalman et un filtre partiel améliorent la précision en diminuant le bruit ainsi que les erreurs de suivi.



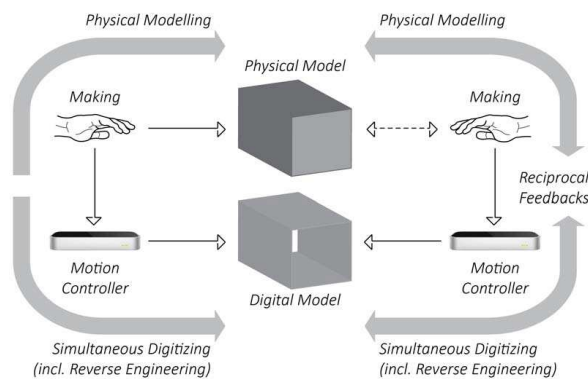
4.1.13 Applications éducatives

“Learning technical drawing with augmented reality and holograms” [Figueiredo et al, 2014]⁷⁷

Le dessin technique est important pour les ingénieurs mécaniciens. Il est souvent difficile pour les étudiants de déduire les vues trois dimensions à partir de dessins en deux dimensions. L'étude montre comment aider les étudiants à visualiser les modèles en trois dimensions. Une façon de faire est d'utiliser les hologrammes interactifs. Le contrôleur *Leap Motion* est utilisé pour interagir avec les hologrammes avec une série de six gestes (déplacements à droite, à gauche, en avant, en arrière, agrandissement et rétrécissement).



“Enhanced Human-Computer Interaction for Digital Conception and Manufacturing in Architectural Education” [Aşut, 2014]⁷⁸



Les phases de conception et de fabrication sont intégrées. Il est possible de revenir constamment à une phase antérieure afin d’améliorer le modèle.

Le mode stylo permet de contrôler les mouvements de la souris pour tracer la géométrie du modèle digital.

Le mode geste permet de contrôler, avec le *Leap Motion*, les formes à dessiner et les outils (déplacement, rotation, copier/coller).

4.1.14 Authentification biométrique

“Leap motion controller for authentication via hand geometry and gestures” [Chan et al, 2015]⁷⁹

L’étude montre comment le *Leap Motion* peut être utilisé comme appareil d’authentification dans un premier cas lors de l’identification et dans un second cas en continu.

Dans le cas de l’identification lors du contrôle d’accès, un scénario statique est utilisé (sans prise en compte du temps) et il est demandé aux participants de dessiner un cercle avec un doigt.

Le scénario comprend la classification des données statique par un algorithme. Cet algorithme est un classificateur basé sur la forêt d’arbre décisionnel (*Random Forest Classifier*). Une sélection d’attributs géométriques de la main identifie ceux qui contribuent à une même classification. L’algorithme teste chaque attribut et retourne une valeur qui indique à quel point il correspond à la classification.

Le résultat des tests indique que l’authentification statique donne une précision de 99,97 %.

Dans le cas de l’authentification continue pendant l’usage d’une application (le site Wikipédia pour les tests), un scénario dynamique est utilisé (avec prise en compte du temps) et il est demandé aux participants de bouger leur index droit. Le contrôleur *Leap Motion* reconnaît leurs gestes.

L’algorithme utilisé est le *Template Matching*. Cet algorithme explore la possibilité de reconnaître un petit nombre d’attributs d’identification de l’utilisateur.

Le résultat des tests indique que l'authentification continue donne une précision de 98%.

En termes d'authentification, seule la géométrie de la main est prise en compte pour le cas statique. Pour le cas dynamique quand il y a un grand nombre d'attributs de gestes, la classification basée sur les propriétés physiques est suffisante.

Le contrôleur *Leap Motion* peut être employé pour le contrôle d'accès et pour l'authentification continue des utilisateurs. Il constitue une alternative non coûteuse aux mots de passe clavier.

4.2 Catégorisation des applications du contrôleur Leap Motion

La catégorisation des applications du contrôleur *Leap Motion* consiste à fournir sous forme d'un tableau par thème et pour chaque référence, l'année où elle fut écrite, son titre, le type de gestes utilisé, l'interface ou les interfaces de programmation employé(s), la combinaison ou non du Leap Motion avec d'autres capteurs en les mentionnant, les remarques éventuelles.

Cette catégorisation donne une vue synthétique de l'état de l'art des applications du contrôleur *Leap Motion*.

Catégorisation des applications du contrôleur Leap Motion

Année	Catégorie d'application	Titre	Type de gestes				API interface de programmation	Combinaison avec d'autres capteurs	Remarques
			Epistémique	Symbolique	Déictique	Manipulation			
2013	Peinture	Peinture dans l'air avec <i>Corel Painter</i>		Symbole référentiel à interpréter	Sélection des brosses, des textures, des commandes, des teintes, de la saturation	Brosses et outils	<i>Corel Painter Freestyle</i>	Non	Permet une nouvelle sensation créative
2017	Peinture	Peinture sans les mains sur un écran à l'aide d'un aérographe virtuel		Symbole référentiel à interpréter	Réglage du <i>Leap Motion</i> et de l'écran. Contrôles. Transformation de repères.	Aérographe virtuel	Un modèle d'interaction : calibrage, transformation de repères, contrôles et génération de signaux. Un modèle peinture : espace d'influence, distribution de peinture.	Non	Les techniques d'interaction sont adaptables et flexibles
2015	Contrôle de systèmes domotiques	Protocole gestuel de contrôle de systèmes domotiques			Choix de l'application domotique, navigation dans le menu		Interface gestuelle unique gérée par le logiciel <i>ConductHome</i>	Non	L'interface gestuelle est conviviale et peut venir en aide aux personnes âgées ou malades
2014	Commande par les gestes d'un robot chargé des tâches domestiques	Interaction dans l'environnement habitable avec des robots de service et le contrôleur <i>Leap Motion</i>		Symbole moralisateur en plus d'une commande vocale		Bras robotisé et interface mobile avec PC, <i>Kinect</i> commandés par les gestes	Un robot interface se déplace dans le logement par la voix et les gestes. Des robots de service muni d'un bras articulé sont commandés par les gestes avec le <i>Leap Motion</i> par l'intermédiaire du robot interface en wifi.	Un capteur <i>Microsoft Kinect</i> se trouve sur l'interface mobile afin de disposer d'une vue 3D de l'intérieur du logement	L'ensemble constitue une interface conviviale pour exécuter des tâches domestiques dans différents endroits du logement
2015	Commande par les gestes d'un robot chargé des tâches domestiques	Télémanipulation par les gestes d'un robot humanoïde pour les tâches domestiques				Robot humanoïde NAO	SDK du <i>Leap Motion</i> en Python. Algorithme de cartographie. Module de commande du robot <i>NAO</i> .	Non	Aide aux personnes malades ou âgées dans leurs tâches domestiques
2014	Contrôle d'un fauteuil roulant	Contrôle d'un fauteuil roulant par le capteur <i>Leap Motion</i> à l'aide d'un algorithme de classification				Fauteuil roulant	Algorithme de classification <i>BS-SRC</i> . Interface d'analyse de données du <i>Leap Motion</i> en <i>MATLAB</i> .	Gyroscope. GPS assistant de navigation personnel.	Un ordinateur pour la reconnaissance des gestes, un ordinateur d'évitement d'obstacles et un ordinateur de commande des moteurs du fauteuil.
2014	Réhabilitation de personne	Interaction sans les mains avec le contrôleur <i>Leap Motion</i> pour la réhabilitation après un accident vasculaire cérébral			Découpage des fruits dans le jeu <i>Fruit Ninja</i>	Fruits à déplacer dans le jeu <i>Fruit Ninja</i>	Utilisation d'une interface de programmation en JAVA écrite au-dessus du SDK du <i>Leap Motion</i> . Version JavaScript du jeu <i>Fruit Ninja</i> .	Non	Le modèle comprend la cinématique de l'entière de la main et permet un suivi plus précis et plus robuste que celui du SDK du <i>Leap Motion</i> .
2016	Réhabilitation de personne	Réhabilitation de la main par le jeu à l'aide du <i>Leap Motion</i>			Pointage d'objets	Manipulation d'objets. Déplacements. Dessins.	Utilisation du SDK standard du <i>Leap Motion</i>	Non	Le système peut être utilisé à domicile en vue d'une réhabilitation.
2014	Réhabilitation de personne	Revue de l'impact clinique et technique de la <i>Microsoft Kinect</i> en réhabilitation			Pointage d'objets	Manipulation d'objets			Suivi en espace rapproché : <i>Leap Motion</i> . Suivi de tout le corps : <i>Microsoft Kinect</i> .
2016	Réhabilitation de personne	Jeux en trois dimensions pour la gestion de la maladie de Parkinson			Pointage d'objets	Manipulation d'objets	Le SDK du <i>Leap Motion</i> nécessite des ajustements de son volume d'interaction afin de permettre d'utiliser la <i>Microsoft Kinect</i> pour l'entraînement des membres supérieurs.	Entraînement des membres supérieurs à l'aide de la <i>Microsoft Kinect</i> et de la motricité des doigts à l'aide du <i>Leap Motion</i> .	Le système est non intrusif et peu coûteux.
2014	Imagerie d'opérations chirurgicales	Système d'interaction sans toucher de gestion d'images médicales en chirurgie avec le <i>Leap Motion</i>			Sélection d'images médicales	Manipulation d'images médicales	Interface Open Source de visualisation des données médicales. Le SDK du <i>Leap Motion</i> est utilisé pour la reconnaissance de gestes.	Non	Le système permet de conserver un environnement stérile pendant l'opération tout en ayant un contrôle direct sur les images pré et intra-opératoires.
2015	Imagerie d'opérations chirurgicales	Contrôle de gestes par le <i>Leap Motion</i> et par <i>Osiris</i> en salle d'opération			Sélection d'images médicales	Manipulation d'images médicales	Application de contrôle des gestes <i>Game Wave</i> . Système de communication et d'archivage <i>Osiris</i> . Gestion d'images médicales <i>OS X Apple</i> .	Non	Limitation des risques d'infection à un coût réduit.

Catégorisation des applications du contrôleur Leap Motion

Année	Catégorie d'application	Titre	Type de gestes				API interface de programmation	Combinaison avec d'autres capteurs	Remarques
			Epistémique	Symbolique	Déictique	Manipulation			
2014	Imagerie d'opérations chirurgicales	Utilisation d'une interface gestuelle comme système de navigation d'images en chirurgie dentaire			Sélection d'images médicales	Manipulation d'images médicales	Logiciel d'acquisition d'image en 2D <i>Kodak dental imaging software</i> . Logiciel d'acquisition d'image en 3D <i>Carestream CS 3D imaging software</i> . Capteur digital de radiographie.	Non	L'interaction gestuelle se fait au niveau du fauteuil dentaire. C'est la première fois qu'une interface utilisateur naturelle est utilisée en chirurgie dentaire.
2014	Contrôle aérien	Interactions gestuelles pour la représentation d'un avion en environnement tangible à destination des contrôleurs aériens	Geste épistémique d'action potentielle	Geste symbolique d'intention de communiquer	Pointage et doubles pointage	Manipulation des bandes de papiers	Reconnaissance de gestes <i>3-Gear</i> API codée en Java et en C# pour l'interface avec le système existant <i>Strip TIC</i>	Le capteur <i>Microsoft Kinect</i> est utilisé pour implémenter les interactions. Il est couplé au logiciel <i>3-Gear</i> . Le <i>Leap Motion</i> est utilisé pour détecter les gestes en bordure du système.	Le système permet plusieurs interactions gestuelles impliquant deux capteurs et un outil de développement de reconnaissance de gestes <i>3-Gear</i> .
2015	Contrôle par le regard et les gestes	Le geste et le regard pour des interactions expressives, précises et ciblées sans le toucher			Pointage précis ou grossier	Manipulation d'objet précise ou grossière	Utilisation du SDK du <i>Leap Motion</i> pour reconnaître les gestes prédéfinis. Application développée en C++. Bibliothèque <i>OpenFrameworks Graphics</i> .	Dispositif de suivi du regard pour le pointage d'objet. Utilisation du <i>Leap Motion</i> pour le suivi des gestes de la main droite qui expriment des fonctions interactives.	Exemples d'applications : contrôle de plages musicales, traitement de texte (sélection par le regard et commandes par les gestes).
2014	Contrôle par les doigts d'appareils mobiles	Etude spécifique de toucher avec les doigts pour des appareils mobiles			Pointage, pression longue, double pointage		Développement d'un algorithme de reconnaissance des doigts au-dessus du SDK du <i>Leap Motion</i> en Python. Smartphone avec annuaire. Ensemble d'applications commandées par les doigts.	Non	Identification de l'usabilité et de la performance de chaque doigt pour les différentes applications.
2016	Contrôle de plusieurs modes d'interaction et de périphériques	Un patron pour des interfaces utilisateur multimodales et multi dispositifs	Type de gestes non spécifié				Utilisation du <i>Model View Controller</i> en JavaScript	<i>Microsoft Kinect</i> , <i>Leap Motion</i> , visiocasque et téléphone mobile monté à l'intérieur du casque.	Le modèle comprend les données des différentes vidéos. La vue gère la visualisation de l'information. Le contrôleur interprète les entrées, utilise le modèle et sélectionne la vue en réaction aux événements.
2016	Calibrage de périphérique	Etalonnage d'un appareil optique de réalité augmentée par le <i>Leap Motion</i>	Type de gestes non spécifié				Utilisation de la méthode d'alignement du simple point actif <i>Tuceryan et Navab</i>	Appareil optique à calibrer	
2016	Langage des signes	Du langage des signes à la conversion de texte en utilisant le <i>Leap Motion</i>		Symbole référentiel signe du langage			Utilisation du SDK JavaScript du <i>Leap Motion</i> pour la reconnaissance de gestes statiques repris en librairie. Trois techniques de reconnaissance des signes sont comparées : correspondance du modèles géométrique, réseaux artificiels neuronaux et corrélation croisée.	Non	La conversion en texte de signes est utilisée par les personnes sourdes ou malentendantes afin de communiquer. La technique de correspondance de modèle géométrique est la plus précise. Elle analyse et trouve de petites parties d'un modèle d'images.
2014	Langage des signes	Reconnaissance du langage des signes américain en utilisant le capteur <i>Leap Motion</i>		Symbole référentiel signe du langage			Le SDK du <i>Leap Motion</i> fournit les caractéristiques de positions, de vitesses des doigts et de la paume de la main. Certaines caractéristiques ne sont pas exploitables en langage des signes comme la position de la paume. Le signe peut être placé n'importe où. Pour reconnaître les signes, deux algorithmes de reconnaissance ont été comparés et utilisés : la méthode des k les plus proches et la méthode des machines à vecteur de support.	Non	La meilleure moyenne obtenue est celle de la méthode des machines à vecteur de support SVM.
2014	Langage des signes	Reconnaissance du langage des signes arabes en utilisant le contrôleur <i>Leap Motion</i>		Symbole référentiel signe du langage			Deux techniques de reconnaissance des signes arabes sont utilisées : l'une basée sur la classification naïve bayésienne et l'autre sur un réseau de neurone artificiel.	Non	Les deux techniques présentent des précisions de reconnaissance élevées (plus de 98 %).

Catégorisation des applications du contrôleur Leap Motion

Année	Catégorie d'application	Titre	Type de gestes				API interface de programmation	Combinaison avec d'autres capteurs	Remarques
			Epistémique	Symbolique	Déictique	Manipulation			
2013	Langage des signes	Le contrôleur <i>Leap Motion</i> : une vue en langage des signes		Symbole référentiel signe du langage australien			Le SDK du <i>Leap Motion</i> réduit les erreurs et augmente la précision. La granulométrie est bien indiquée pour la reconnaissance de gestes et le contrôleur permet de détecter de petits mouvements.	Non	Le <i>Leap Motion</i> convient pour détecter des signes basiques mais pas des signes complexes en contact avec la face ou le corps.
2014	Réalité virtuelle	Une interface 3D pour explorer et manipuler des scènes virtuelles en utilisant le contrôleur <i>Leap Motion</i>		Navigation dans l'espace virtuel, changement d'échelle	Choix des objets	Manipulation d'objets virtuels	Le SDK du <i>Leap Motion</i> est utilisé pour la reconnaissance de gestes. Une interface de programmation graphique libre <i>OpenSceneGraph</i> est utilisée conjointement pour une visualisation en grand, la gestion et le contrôle des données 3D complexes.	Non	Un procédé incluant des arrêts et des glissements permet d'augmenter la précision, de faciliter les transitions et de diminuer la fatigue.
2013	Réalité virtuelle	Une interface hybride de réalité augmentée supportée par le contrôleur Leap Motion		Changement de l'environnement	Pointage d'objets	Prise et déplacement d'objets	Solution graphique hybride : les doigts sont rendus en 2D, l'estimation de profondeur est donnée par le <i>Leap Motion</i> à l'aide d'un modèle converti en pixel 3D.	Caméra vidéo 2D de réalité augmentée	Les interactions sont plus convaincantes entre les objets et leur environnement. La correction des problèmes d'occlusion entre les doigts et l'environnement est nécessaire.
2014	Réalité virtuelle	Visualisation de données immersives, collaboratives avec des plateformes de réalité virtuelle		Visualisation des données	pointage des données	Manipulation des données	Logiciel de visualisation des données développé en Unity 3D	Casque de réalité virtuelle <i>Oculus Rift</i> Souris 3D et <i>Microsoft Kinect</i>	Le but de l'étude est de maximiser le modèle de reconnaissance humaine à travers un usage en réalité virtuelle.
2014	Application musicale	Le <i>Leap Motion</i> employé comme une interface gestuelle expressive		Contrôle des paramètres visuels et musicaux	Choix des cadres Pointages des mains et des doigts		Le SDK du <i>Leap Motion</i> est utilisé pour la reconnaissance des gestes de base (cercle, balayage, pointage d'écran et de touche). Utilisation du <i>MRLeap</i> qui fournit une solution d'interfaçage entre les utilisateurs et le <i>Leap Motion</i> pour une maximum de polyvalence et de contrôle avec <i>MaxMSP</i> .	Non	Le Leap Motion convient pour détecter de petits mouvements spécifiques. Comme instrument musical, le <i>Leap Motion</i> est polyvalent et expressif.
2014	Application musicale	Le capteur <i>Leap Motion</i> conçu comme un instrument contrôlé par les gestes		Contrôle des zones d'interaction et des effets sonores	Pointage des touches Pression et relâchement des touches		Le SDK du <i>Leap Motion</i> est utilisé pour la reconnaissance de gestes de pointage pour déclencher des messages MIDI	Pas de combinaison avec un autre capteur mais le <i>Leap Motion</i> complète un clavier pour en faire un instrument musical augmenté.	Les <i>Air Key</i> créent un clavier virtuel avec des notes paramétrables. Les <i>Air Pad</i> fournissent des effets sonores par création de boutons suivant des zones d'interaction.
2014	Application musicale	Implémentations du <i>Leap Motion</i> dans les sons de synthèse, les effets modulants et les outils d'assistance performants		Contrôle par les mains ouvertes ou fermées			Le logiciel Geco permet de transformer les mouvements de la main en messages MIDI.	Non	Pas de contrôle possible par les doigts.
				Contrôle des paramètres visuels et musicaux	Choix des cadres Pointages des mains et des doigts		Utilisation du logiciel <i>MaxMSP</i> avec l'API du <i>Leap Motion</i>	Non	Ne permet pas de distinguer la main droite de la main gauche ni quel doigt interagit.
				Contrôle d'effets, de paramètres musicaux	Sélection des notes, des échelles, des octaves		Utilisation du nouvel API <i>beta</i> qui permet de suivre distinctement les mains et les doigts de chaque main.	Non	
				Contrôle des effets sonores			Utilisation de <i>A square</i> et de <i>Ableton Live</i>	Non	Modulation et mélange de plusieurs effets sonores par un contrôle des mains.
				Contrôle de volume de longueur des fragments	Déclenchement des modes et des fragments		Utilisation du logiciel <i>MaxMSP</i> pour un synthétiseur de courts fragments. Suivi des mains et des doigts avec l'API <i>beta</i> .	Non	Permet de nouvelles façons de synthétiser les sons.
				Contrôle du volume des paramètres sonores	Sélection du haut-parleur		Utilisation du logiciel <i>MaxMSP</i> pour un commander un objet composé de haut-parleurs. Suivi des mains et des doigts avec l'API <i>beta</i> .	Non	
				Contrôle des paramètres musicaux	Pointages des touches		Suivi des mains et des doigts avec l'API <i>beta</i>	Non	Apprentissage de la musique à l'aide du <i>Leap Motion</i> .

Catégorisation des applications du contrôleur Leap Motion

Année	Catégorie d'application	Titre	Type de gestes				API interface de programmation	Combinaison avec d'autres capteurs	Remarques
			Epistémique	Symbolique	Déictique	Manipulation			
2014	Application musicale	Création de musique avec le <i>Leap Motion</i> et le logiciel <i>BigBang Rubette</i>		Contrôle des paramètres musicaux	Pointages des touches du clavier virtuel		Suivi des mains et des doigts avec l'API <i>beta</i> . Utilisation du logiciel <i>BibBang Rubette</i> qui paramétrise les objets musicaux : notes, oscillateurs, modulateurs.	Non	Le logiciel permet de voir la représentation de la composition musicale. Le système augmente l'interaction entre les mouvements de l'interprète et les sons produits.
2014	Application musicale	Détection des doigts sur un clavier de piano virtuel			Pointages des touches du clavier virtuel		Utilisation du SDK pour reconnaître une pression sur une touche envoyée comme une donnée MIDI. Logiciel développé pour envoyer les positions des doigts à une application Android fonctionnant sur tablette.	Non	Les doigts ne sont pas toujours reconnus surtout si le mouvement est rapide.
2013	Application musicale	Evaluation du <i>Leap Motion</i> comme nouvel instrument de musique digital			Pointages des touches du clavier virtuel		Utilisation du SDK du <i>Leap Motion</i> pour la reconnaissance de gestes	Non	Le <i>Leap Motion</i> apporte une bonne simulation pour un rythme lent.
2016	Application musicale	Gestes des mains en production musicale		Gestes de contrôle du tempo, du volume, du phrasé, du timbre			Utilisation du SDK du <i>Leap Motion</i> pour la reconnaissance de gestes	Non	Les gestes ont été codifiés pour reprendre les caractéristiques du son.
2014	Application industrielle	Un robot de soudure commandé par le contrôleur <i>Leap Motion</i>		Suivi des gestes de l'homme			Utilisation d'un algorithme de contrôle prédictif	Non	Le <i>Leap Motion</i> renvoie les données de suivi à un PC qui commande un robot de soudure à six degrés de liberté.
2014	Application industrielle	Utilisation du <i>Leap Motion</i> pour commander un grappin à trois doigts				Manipulation d'objets	Utilisation du SDK du <i>Leap Motion</i> pour la reconnaissance de gestes. Utilisation d'un API complémentaire pour suivre les objets longs et droits et calculer le rayon d'une balle dans la paume de la main.	Non	Le <i>Leap Motion</i> est comparé à un système muni de caméras infrarouges et de caméras couleurs pour la même application. Les gestes et le logiciel sont adaptés en fonction des capteurs.
2015	Application industrielle	Commande de deux robots avec deux capteurs <i>Leap Motion</i> et des filtres				Manipulation d'objets	Utilisation du SDK du <i>Leap Motion</i> pour la reconnaissance de gestes des mains. Un filtre Kalman et un filtre partiel améliorent la précision en diminuant le bruit et les erreurs.	Non	
2014	Education dessin technique	Etude du dessin technique avec la réalité augmentée et les hologrammes			Pointage des hologrammes	Manipulation des hologrammes	Utilisation d'une application en <i>Unity 3D</i> pour visualiser les modèles 3D. Utilisation de l'interface de programmation du <i>Leap Motion</i> pour la reconnaissance de gestes.	Non	Le système permet aux étudiants de déduire les vues en trois dimensions à partir de dessins en deux dimensions.
2014	Education architecture	Interaction homme-machine améliorée pour la conception et la fabrication numérique dans l'enseignement de l'architecture		Traçage des modèles à l'aide du mode stylo et de la souris		Manipulation des modèles déplacement rotation copier/coller	Développement en <i>SketchupMake 3D</i> qui supporte une interface de programmation en <i>Ruby</i> et utilisation du logiciel <i>Autocad 3D</i> Application de reconnaissance de gestes en JavaScript qui utilise le SDK du <i>Leap Motion</i> .	Souris	Les phases de conception et de fabrication sont intégrées. Il est possible de revenir à une phase antérieure pour l'améliorer.
2015	Authentification biométrique	Le contrôleur <i>Leap Motion</i> pour l'authentification avec la géométrie de la main et les gestes		Identification de contrôle d'accès statique Authentification continue dynamique			Utilisation du SDK du <i>Leap Motion</i> pour la reconnaissance de geste en statique (dessin d'un cercle) et en dynamique utilisation de l'index droit avec <i>Wikipédia</i> . Algorithme de classification de données statique et algorithme de classification basé sur les propriétés physiques en dynamique.	Non	L'utilisation du contrôleur <i>Leap Motion</i> constitue une alternative non coûteuse aux mots de passe clavier.

4.3 Etat de l'art des applications du Leap Motion : discussion

Au vu du tableau synthétique de catégorisation et de la lecture d'autres références, il s'ensuit une réflexion personnelle de l'état de l'art des applications du *Leap Motion*.

4.3.1 Types de gestes employés dans les différentes études

Les gestes épistémiques concernent uniquement l'application de contrôle aérien.

Les gestes symboliques sont employés dans les domaines suivants : la peinture, le contrôle aérien, le langage des signes, la réalité virtuelle, la musique, l'industrie et la biométrie.

Les gestes déictiques sont utilisés dans toutes les applications sauf le langage des signes, l'industrie et la biométrie.

Les gestes de manipulation d'objets dépendent du domaine d'application. Ils ne sont pas présents dans les applications musicales ni dans celles du langage des signes.

4.3.2 Utilisation du Leap Motion dans la vie courante

Au vu des différents articles, le Leap Motion ne convient pas pour une utilisation scientifique qui demande une grande fiabilité. Par contre, le *Leap Motion* convient à la vie courante pour des applications dont les utilisateurs connaissent déjà la finalité. En raison de son coût relativement faible, il est aisé de réaliser une enquête portant sur plusieurs utilisateurs. Dans ce contexte, l'utilisation du *Leap Motion* en interaction avec un téléviseur [Vatuvu et al, 2014]⁸⁰ est intéressante pour remplacer la télécommande. L'emploi de la télécommande est bien connu de tous de sorte que l'utilisation du contrôleur *Leap Motion* pour son remplacement est intuitive et non déroutante. L'interaction avec le téléviseur a considérablement évolué ces dernières années au point de vue contenu, technologie (effets sonores, visuels), accessibilité et échange social mais elle reste toujours basée sur l'emploi d'une télécommande.

La reconnaissance de gestes statiques (postures) est plus fiable avec le Leap Motion [Marin et al, 2015]⁸¹. Cependant, toutes les commandes ne peuvent être uniquement réalisées avec des postures. Dans l'ensemble des gestes dynamiques possibles, il y a lieu de choisir les plus intuitifs pour l'utilisateur et les plus fiables en termes de reconnaissance. Il est important de diminuer les faux positifs (détection positive d'un geste alors qu'il n'existe pas) ainsi que la détection de nouveaux gestes alors qu'il s'agit d'un même geste. Pour certains types de gestes comme le cercle, le *Leap Motion* éprouve des difficultés à identifier le début et la fin du geste. Une piste possible serait de réaliser des gestes statiques fiables pour reconnaître le début et la fin d'un geste dynamique. Marin [Marin et al, 2015]⁸² a pu établir une nomenclature de fonctions basée sur les positions et les orientations des doigts et des mains. Ces fonctions sont envoyées vers un classificateur afin d'établir une segmentation de reconnaissance de gestes statiques. De même, dans le cas du téléviseur, une nomenclature des gestes de contrôle du téléviseur est à établir.

Le *Leap Motion* peut être utilisé comme dispositif en entrée d'écriture. Le caractère réalisé dans l'air par un doigt est identifié. Le problème est que le début et la fin d'entrée de données (correspondant au

mode stylo) est difficilement identifiable. Des postures clairement identifiables permettent de les distinguer mais cela rend l'interface peu intuitive. Le *Leap Motion* permet le suivi par des gestes intuitifs d'une présentation. Cela donne à l'utilisateur plus de contrôle de son contenu et des informations présentées. Il peut déplacer des objets, les agrandir, dessiner à l'écran, démarrer et arrêter des vidéos dynamiquement à l'aide de gestes. Il est possible d'utiliser des images et des photos à retravailler, de créer des éditions de photos, de mélanger des photos et des peintures. Le *Leap Motion* peut être utilisé pour réaliser des réunions à distance. Chaque participant dispose du volume d'interaction de son *Leap Motion* pour collaborer avec les autres intervenants. Le *Leap Motion* permet la Commande par les gestes d'un robot chargé des tâches domestiques. Plutôt que d'utiliser un vrai robot qui est relativement cher, il est possible de programmer un robot virtuel qui réagit par les gestes. Une enquête utilisateur dans l'environnement de réalité virtuelle détermine si les gestes de commande sont intuitifs.

La commande d'un fauteuil roulant par *Leap Motion* peut être améliorée afin de convenir à plusieurs types d'handicap. Les uns utilisent une commande par joystick, les autres une commande par les gestes. L'ordinateur embarqué nécessitant beaucoup est remplacé par un micro-ordinateur. L'interface homme-machine devient plus conviviale grâce à un petit écran avec des commandes claires et des icônes représentatives des gestes. L'autonomie du fauteuil est meilleure par l'emploi de batteries plus compactes et de moteurs électriques moins puissants.

4.3.3 Utilisation du Leap Motion avec d'autres capteurs

Le tableau indique que peu d'études ont été réalisées pour une utilisation conjointe du contrôleur *Leap Motion* avec d'autres capteurs.

Le capteur avec lequel il est le plus souvent combiné est le capteur *Microsoft Kinect*. Les applications qui les combinent sont les suivantes : la commande par les gestes d'un robot domestique [Gergoulas et al, 2014]⁸³ pour laquelle le capteur *Microsoft Kinect* permet d'obtenir une vue 3D de tout le logement et le capteur *Leap Motion* de commander par les gestes des robots munis d'un bras de service et le robot interface ; la réhabilitation de personnes [Mousavi et al, 2014]⁸⁴ pour laquelle le capteur *Microsoft Kinect* aide à l'entraînement des membres supérieurs (suivi de l'ensemble du corps) et le capteur *Leap Motion* aide à la motricité des doigts (suivi en espace rapproché) ; le contrôle aérien [Gauthier et al, 2014]⁸⁵ pour lequel le capteur *Microsoft Kinect* implémente les interactions et le capteur *Leap Motion* détecte les gestes en bordure du système ; la visualisation de données immersives en réalité virtuelle [Donalek et al, 2014]⁸⁶ pour laquelle le capteur *Leap Motion* pointe et manipule des données tandis que le capteur *Microsoft Kinect* autorise la visualisation de leur ensemble.

D'autres applications spécifiques combinent le contrôleur *Leap Motion* avec un autre appareillage : le contrôle d'un fauteuil roulant [Boyalı et al, 2014]⁸⁷ utilise en plus du capteur *Leap Motion*, un gyroscope et un assistant personnel de navigation ; le contrôle par le regard [Chatterjee et al, 2015]⁸⁸ se fait par dispositif particulier de suivi et les contrôle des gestes par le *Leap Motion* ; les applications de réalité virtuelle utilisent soit une caméra vidéo 2D de réalité virtuelle ou un casque de réalité virtuelle ; le *Leap Motion* est employé en complément de la souris pour une application éducative architecturale [Aşut,2014]⁸⁹.

Pour la première fois, le contrôleur *Leap Motion* permet d'interagir manuellement avec des objets en trois dimensions d'un environnement virtuel grâce à un suivi détaillé et précis des gestes de la main.

Cette interaction permet de concevoir des scénarii plus réalistes en réalité virtuelle. Une partie importante de la communication homme – machine est donnée par le retour tactile ou haptique (concernant le sens du toucher) qui demande une réponse de l'utilisateur avant la poursuite d'une étape. Le *Leap Motion* ne permet pas une perception claire de profondeur car l'utilisateur ne sait pas exactement si ses mains sont bien détectées. Cela engendre des erreurs d'interprétation et une mauvaise réponse de l'utilisateur. Plusieurs possibilités existent pour enrichir l'interaction : les retours tactilo-kinesthésiques ou haptiques, visuels ou auditifs.

Dans le cadre de la réalité virtuelle, le retour tactile ou haptique est naturel et donne l'impression d'une expérience utilisateur réelle. C'est pourquoi l'utilisation conjointe d'un contrôleur portable (Hydra) [<https://fr.wikipedia.org>, 2017]⁹⁰ permettant un retour tactile sous forme de vibration ou pression, donne à l'utilisateur une meilleure perception de l'espace. L'interaction des deux mains est aussi plus réaliste. L'utilisateur peut avoir une relation tangible avec l'objet de la scène. Le *Leap Motion* peut être utilisé pour effectuer des mouvements et des rotations d'objets 3D.

La perception de profondeur (la distance entre l'objet et la scène autour de celui-ci) est également difficile à estimer dans un environnement réel. Seul un entraînement plus poussé permet à l'utilisateur d'améliorer la reconnaissance de ses mains et de ses doigts. Cela concerne par exemple le toucher d'écran ou du clavier et le clic en trois dimensions. De plus, il peut avoir des problèmes de vitesse et de position. L'emploi d'un retour auditif, visuel ou tactile peut aider l'utilisateur. Comme retour visuel de la position où il faut cliquer, le bouton peut par exemple devenir plus grand et être accompagné d'un son comme retour auditif. Le retour tactile nécessite quant à lui l'emploi conjoint d'un autre capteur comme décrit plus haut.

Le contrôle d'une tâche de préhension d'un objet virtuel doit être le plus simple possible. Il en résulte que le test portera sur une unique variable booléenne : la main a-t-elle tous ses doigts visibles alors elle peut prendre l'objet sinon il y a détection du poing et elle ne peut prendre l'objet.

Le graphique moléculaire [Sabir et al, 2013]⁹¹ est utilisé pour la recherche et pour la communication. Il permet de manipuler des macromolécules. Cette manipulation est difficile avec la souris et le clavier. Elle peut être facilitée par les gestes de la main et la voix. Les gestes plus précis de la main et des doigts seront suivis à l'aide du *Leap Motion*. Les gestes du corps et la voix seront suivis par le *Microsoft Kinect*.

Une autre utilisation du *Leap Motion* avec un capteur différent est celle de l'utilisation conjointe avec le casque de réalité virtuelle *Oculus Rift*.

Il est intéressant d'étudier le comportement conjoint de ces deux capteurs car ils sont financièrement accessibles. Un article s'y rapportant résume une étude d'une plate-forme de visualisation et d'analyse. Peu de logiciels ont été réalisés jusqu'à présent pour utiliser conjointement ces capteurs. Des gestes particuliers ont été choisis pour une reconnaissance par les deux capteurs. Il s'ensuit que le *Leap Motion* n'est pas assez précis et fiable pour l'utilisation du prototype repris dans l'étude. Le nombre de cadres à la seconde est différent pour le *Leap Motion* et l'*Oculus Rift* et réduit la performance. Il s'ensuit que le *Leap Motion* ne convient pas très bien pour être monté sur ce casque de réalité virtuelle.

L'utilisation conjointe du *Leap Motion* avec une table d'interaction peut créer un système de particules de différentes couleurs à l'aide de gestes des mains au-dessus du *Leap Motion*. Le système de

particules suit ensuite les mouvements de la main en réalisant des effets visuels. Le *Leap Motion* peut contrôler des LED lumineuses par les gestes en ouvrant et fermant les LED et créer des effets lumineux.

4.3.4 Interaction en environnement tridimensionnel

Une utilisation intéressante du *Leap Motion* concerne le contrôle et la visualisation de données. Les technologies de visualisation 3D, regroupant les visionneuses 3D et les éditeurs en trois dimensions sont peu adaptés aux interfaces utilisateurs disponibles pour les ordinateurs personnels (le clavier et la souris) conçu pour une interaction en 2D. Il en va de même pour les interfaces tactiles dont les actions sont limitées à un plan 2D. L'apprentissage du contrôle et de la visualisation des données par le seul *Leap Motion* peut être long car les zones de détection ne sont pas clairement délimitées. Une interface plus intuitive peut être utilisée pour suivre la main de l'utilisateur à l'aide du *Leap Motion* dans l'espace 3D. Elle utilise une carte matériellement libre (*open source*) équipée d'un microcontrôleur (*Arduino*) [<https://fr.wikipedia.org>, 2018]⁹² programmé pour produire des signaux électriques convertis en signaux lumineux par des LED. Le signal lumineux avertit que la main se trouve dans la zone de détection souhaitée.

Le son joue un rôle important pour une immersion véritable dans un monde virtuel. Il renforce l'interaction et conduit à plus d'émotion et de profondeur. Pour ce faire, il est couplé à la reconnaissance des mains et au retour visuel assurant l'illusion d'une sensation tactile. Les effets sonores choisis lors de la manipulation des objets 3D et de leurs déplacements dans l'environnement doivent créer une illusion naturelle.

La peinture dans l'air avec le *Leap Motion* peut s'employer dans le but de disposer dynamiquement et rapidement d'une représentation visuelle d'un agencement de couleurs d'un modèle en 3D à partir d'un logiciel de dessin. Ces modèles sont ensuite utilisés dans les domaines de la mode, du marketing, du design, de l'éducation.

4.3.5 Interactions multiples

Contrôle de système domotique

Les contrôles domotiques peuvent être multiples et de différents types : contrôle du chauffage en fonction de la météo, des conditions extérieures, du système d'intrusion, du système de détection incendie, de la musique dans une ou l'autre pièce, contrôle de la lumière, visualisation de la météo, contrôle vocal, par les gestes, par le toucher. Les retours d'interaction peuvent être mélangés par exemple sonore et visuel. Certains contrôles non accessibles aux sourds le sont par les gestes.

Contrôle par le regard et les gestes

Le regard permet une sélection dans un menu, dans un ensemble d'images et les gestes permettent une interaction plus précise pour réaliser des commandes.

Exemples :

- sélectionner dans une bibliothèque de livres, d'articles traitant d'un sujet puis visualiser, tourner les pages, zoomer, conserver des données ;
- sélectionner des articles d'un journal puis les visualiser, les conserver et les comparer ;
- voyager à l'aide de *Google Earth*, choisir l'endroit par le regard, l'affiner par les gestes et se déplacer par les gestes.

Contrôle aérien

Le contrôle aérien nécessite des interactions particulières et spécifiques. Il existe tout un langage propre aux contrôleurs. Il peut être intéressant d'améliorer leur travail grâce à une maquette assez proche de ce qu'elle sera en réalité. Les interactions sont multiples et peuvent être de différents types : gestuelles, vocales, sensorielles. Il est impératif de faire une enquête utilisateur poussée pour connaître le vocabulaire employé et les techniques déjà en place.

Langage des signes

Le *Leap Motion* a été testé pour la reconnaissance de différents langages des signes avec plus ou moins de succès. Il est intéressant de permettre à deux personnes une sourde ou une muette et une valide de dialoguer à distance grâce à la reconnaissance des gestes par le *Leap Motion*, de traduire visuellement ou vocalement. La reconnaissance vocale convertit la voix en langage des signes compréhensibles par la personne sourde ou muette.

4.3.6 Utilisation du Leap Motion pour une application musicale

Dans une application musicale, afin de visualiser les instruments d'un orchestre, il est possible de les représenter par un ensemble de particules colorées [Fonteles et al, 2015]⁹³. Ensuite, l'utilisateur évalue l'arrangement musical de l'orchestre à l'aide de ce retour visuel.

Les gestes du chef d'orchestre sont suivis en temps réel par le *Leap Motion* et pilotent des entrées MIDI (*Musical Instrument Digital Interface*) contenant les informations de l'arrangement pour chaque instrument rendues visibles à l'écran.

Un contrôleur virtuel MIDI émet des changements d'état en réponse à l'interaction par les gestes de l'utilisateur reconnus par le *Leap Motion*. La station de travail digitale gère les effets audio grâce à ces changements d'état. Des éléments musicaux sont enregistrés séparément pour être restitués selon un certain ordre postérieurement. Le système doit contrôler simultanément plusieurs paramètres afin de créer des performances complexes. L'interface naturelle doit rendre un jeu identique à celui de plusieurs instruments réels.

La définition du vocabulaire musical permet d'identifier de nouvelles commandes par les gestes. L'enregistrement et l'identification des gestes du chef d'orchestre pour telle ou telle partition permet d'aider les musiciens dans leur interprétation. Une commande par les gestes peut compléter un instrument classique.

4.3.7 Utilisation du Leap Motion pour la réhabilitation de personnes

Utilisation du Leap Motion pour la reconnaissance en temps réel des gestes statiques (postures) de réhabilitation des extrémités supérieures

Le *Leap Motion* reconnaît plus aisément des postures. La reconnaissance utilise un arbre de reconnaissance qui classe les postures de la plus courante à la moins courante (des extensions de chacun des doigts aux positions de la paume). La classification peut être améliorée par un algorithme spécifique. Les mesures sur la posture reconnue permettent ensuite de suivre la réhabilitation d'un patient en fonction de critères médicaux.

Utilisation du Leap Motion pour le déplacement et l'interaction dans un environnement virtuel 3D pour la réhabilitation de personnes ayant subi un accident vasculaire cérébral

Les gestes sont utilisés pour la réhabilitation de personnes victime d'AVC (accident vasculaire cérébral) car ils font partie du langage naturel. L'environnement virtuel 3D n'est autre qu'une habitation. Les patients se déplacent dans cet environnement en effectuant des actions dans un contexte donné. Ils sont aidés par des menus d'action et de navigation qui apparaissent en premier plan de l'environnement 3D. Les gestes sont choisis pour être les plus intuitifs et adaptés à la navigation et à l'interaction voulue.

Dans le domaine de la Réhabilitation de personnes, il est important de faire le lien entre les résultats de recherche médicaux et l'application développée. Celle-ci sera modifiable en fonction de la personne d'une part et de l'amélioration de sa condition d'autre part. L'utilisation de différents capteurs comme le *Leap Motion* et la *Microsoft Kinect* est intéressante afin de tester différentes possibilités ou différents membres voire de l'entièreté du corps.

4.3.8 Reconnaissance de gestes

La reconnaissance de gestes peut être statique ou dynamique. [Shao et al, 2016]⁹⁴

La reconnaissance de gestes statiques (postures) est principalement basée sur la distance relative de la paume et des doigts. Deux distances sont calculées : la distance de l'extrémité d'un doigt au centre de la paume et la distance entre deux extrémités de doigts adjacents. La main ouverte avec tous les doigts déployés et le poing sont des postures dont les distances des extrémités servent de paramètres pour les différentes postures. Les autres postures sont construites à partir des distances entre les doigts et la paume.

La reconnaissance dynamique de gestes se distingue de la reconnaissance statique par une vitesse de déplacement entre les doigts et la paume supérieure à certain seuil. A partir d'un mouvement global, les mouvements des parties constituantes en sont déduits. D'abord les mouvements des mains puis ceux des doigts.

Le kit de développement logiciel du *Leap Motion* est utilisé pour la reconnaissance des gestes de base (cercle, pointage, balayage et sélection de touche) et pour le suivi des mains et des doigts en espace rapproché. Un logiciel complémentaire est utilisé pour les reconnaissances de gestes spécifiques.

D'autres logiciels propres au domaine étudié complètent l'interface de programmation de l'application.

Amélioration de l'interaction homme-machine

Dans le meilleur des cas, l'interface de programmation du *Leap Motion* reconnaît un geste après la réalisation complète de celui-ci. Cette façon de procéder réduit la souplesse de l'interaction car elle induit des temps d'attente. L'interaction serait fortement améliorée avec une reconnaissance rapide d'un geste dynamique de la main avant que ce geste ne soit complètement réalisé.

Cela est réalisable avec un algorithme de classification selon les étapes suivantes :

- La trajectoire suivie par le *Leap Motion* est filtrée ;
- les séquences de positions sont quantifiées ;
- le filtre est réduit par remplissage de zéros ;
- une base de données est nourrie de gestes partiellement exécutés ;
- le modèle d'algorithme spécifique (*SVM* [<https://fr.wikipedia.org>, 2018]⁹⁵) est entraîné à la reconnaissance des éléments de la base de données ;
- la reconnaissance rapide de gestes incomplets est possible.

Utilisation d'une librairie de gestes

L'ajout de gestes reconnus à une librairie augmente le nombre de fonctionnalités distinguables par les utilisateurs à l'aide de gestes. Ces gestes peuvent être statiques ou dynamiques. L'ajout de gestes dynamiques implique une amélioration des algorithmes de reconnaissance.

4.3.9 Utilisation du Leap Motion en chirurgie ouverte

Pour cette application, le *Leap Motion* est intéressant car il est non invasif. Les gestes humains sont très complexes ce qui entraîne l'utilisation d'un algorithme de reconnaissance complexe. L'algorithme de reconnaissance simplifié du *Leap Motion* ne permet pas de tout percevoir.

Lors d'une opération chirurgicale du cerveau, il est important pour le chirurgien d'avoir une vue en trois dimensions afin de localiser précisément une tumeur. Un scanner IRM en deux dimensions permet d'acquérir une grande série de coupes du cerveau et un algorithme le reconstitue en trois dimensions.

Le *Leap Motion* permet au chirurgien de manipuler ce modèle 3D (zoom, dé zoom, rotation, vues). L'interaction par les gestes est précise et assez intuitive pour les chirurgiens. Elle permet de leur faire gagner du temps et de conserver un environnement stérile. Le potentiel de cette application à l'aide du *Leap Motion* est élevé.

Il est important de définir un vocabulaire gestuel spécifique pour une seule main (la deuxième se chargeant de l'opération) afin de se concentrer plus sur l'effet du geste.

La saisie et la manipulation d'objets 3D par les gestes permettent de faire une répétition à l'avance de l'opération chirurgicale pour vérifier par simulation l'impact de tel ou tel geste sur l'objet 3D (le cerveau ou tout organe du patient).

4.3.10 Utilisation du Leap Motion avec les jeux

Le contenu du jeu est primordial pour absorber l'attention du joueur. Les jeux qui emploient la reconnaissance de gestes capturent l'attention du joueur pendant toute la progression du jeu.

Le *Leap Motion* permet un nouveau type d'interaction avec une interface utilisateur naturelle *NUI*. Cette interface a la particularité d'être invisible et de le rester au fur et à mesure que l'utilisateur interagit avec le système. L'emploi d'une interface naturelle pour laquelle les interactions viennent spontanément sans apprentissage accroît l'immersion dans le contenu du jeu. Le *Leap Motion* suit les gestes de l'utilisateur sans aucun contrôle et en temps réel pendant qu'il réagit avec le contenu du jeu. L'utilisation d'une interface utilisateur naturelle nécessite que les conditions par défaut correspondent aux objectifs de l'utilisateur. Pour le *Leap Motion* cela implique que les gestes répondant à tel évènement soient identiques à ceux pensés par l'utilisateur. L'idéal est que l'interface de programmation du *Leap Motion* inclut les gestes usuels des joueurs. Le kit de développement du *Leap Motion* inclut plusieurs API dont celui des mains et celui des doigts. Un lexique qui explique un mouvement à l'aide des mains et des doigts aidera les utilisateurs face aux attentes du jeu dans une situation particulière. Certains jeux impliquant beaucoup de joueurs en même temps (par exemple les jeux de sport) ne sont pas réalisables avec le *Leap Motion*.

4.3.11 Applications particulières

Utilisation du Leap Motion pour la conduite de drone

Le contrôle du drone se fait à l'aide de gestes naturels : aller en avant, en arrière, à gauche, à droite par des commandes de position de la main ; le roulis, le tangage et le lacet par des commandes d'orientation de la main. Un microcontrôleur (*Arduino*) se charge de convertir les mouvements reconnus de la main par le *Leap Motion* en signaux électriques puis en signaux infrarouges de commande du drone.

Utilisation du Leap Motion pour le contrôle par les gestes d'une source de puissance d'un équipement de laboratoire alimentant un circuit électronique expérimental

L'intérêt du contrôle par les gestes est que l'utilisateur ne touche pas physiquement le circuit électronique expérimental. Le contrôle par les gestes de la source de puissance est une alternative au contrôle électronique par un ordinateur. L'interface naturelle utilisateur (*NUI*) reprend les fonctionnalités de l'instrument de test. Le système à développer identifie les gestes qui déclencheront ces fonctionnalités. Il est important d'analyser l'utilisabilité du dispositif au travers d'une enquête de satisfaction ainsi que l'entraînement nécessaire pour son usager.

Biométrie

Le *Leap Motion* permet la reconnaissance et l'identification de l'individu à l'aide de paramètres propres : angles des doigts, position de la main, longueur des doigts à des fins d'authentification.

Application industrielle

La généralisation de la reconnaissance de geste d'un humain facilite l'apprentissage d'une machine et améliore la flexibilité du travail.

4.3.12 Utilisation du contrôleur Leap Motion pour les applications web

Le service *Leap Motion* [<https://developer.leapmotion.com>] ⁹⁶ est fourni par un serveur utilisant un protocole *websocket*. Ce protocole est activé ou désactivé par l'utilisateur afin de reconnaître les gestes ou pas. Le serveur envoie les données de suivi sous forme de *JSON* et l'application configure les messages à renvoyer en retour.

Avec l'application web, le nombre de gestes reconnus est limité car le capteur *Leap Motion* prévient la confusion des gestes afin de permettre une navigation plus aisée. La possibilité de reconnaissance de gestes est donc plus faible que celle disponible pour une application locale.

Le choix des gestes implique toujours de disposer de gestes intuitifs et naturels. Cependant, la précision de la plupart des gestes est limitée pour une expérience utilisateur correcte. Si plusieurs applications web travaillent simultanément, les volumes d'interaction de celles-ci ne correspondent pas nécessairement. En outre, l'interaction dépend entièrement de l'état de la connexion internet.

5. L'interface de programmation API Java du contrôleur Leap Motion

5.1 Présentation de l'interface de programmation API Java du Leap Motion

Ce chapitre présente l'interface de programmation API java du contrôleur *Leap Motion*. L'interface consiste en une librairie dynamique constituée de différentes classes utilisables pour le développement logiciel. Les données sont transférées depuis le *Leap Motion* grâce au contrôleur. Un système d'écouteur prend en charge les différents évènements.

Les données sont réparties en six types : les données de la main, des doigts, des outils, les gestes de base, les mouvements spécifiques et les images brutes. Chaque type a des paramètres propres.

L'interface de programmation du *Leap Motion* contient des fonctions définies qui sont stockées dans les bibliothèques. L'interface native est une librairie dynamique qui peut être utilisée pour créer de nouvelles applications basées sur le contrôleur *Leap Motion*. Les programmeurs peuvent se servir des fonctions incluent dans les bibliothèques en les ajoutant à leur logiciel. Le *Leap Motion* dispose d'une interface de programmation qui permet la reconnaissance et le suivi des mains, des doigts et des outils.

L'interface de programmation est contenue dans le kit de développement SDK du *Leap Motion*. Ce kit de développement contient les API ainsi que les documentations pour différents langages de programmation, le lien de téléchargement, un forum de discussion et des outils de développement. L'interface de programmation API java fonctionne de la même façon que celle des autres langages de programmation et se base sur l'API écrit en C++. Les fonctions natives en C++ sont appelées par java à l'aide d'un outil open source SWIG (*Simplified Wrapper and Interface Generator*).

Par rapport au premier kit de développement, le nouveau kit incorpore de nouvelles fonctionnalités telles que :

- la main droite et la main gauche sont définies et différenciées ;
- il existe un indicateur de préhension de la main ;
- les cinq doigts de la main sont présents ;
- chaque doigt peut être identifié et possède un type ;
- il existe un indicateur de pincement entre le pouce et l'index ;
- chaque os des mains est défini ainsi que la position des jointures des os ;
- la détection de la main est plus aisée.

Les mesures physiques de l'interface de programmation sont données dans les unités suivantes : la distance en millimètres, le temps en microsecondes, la vitesse en millimètres par seconde, l'angle en radians.

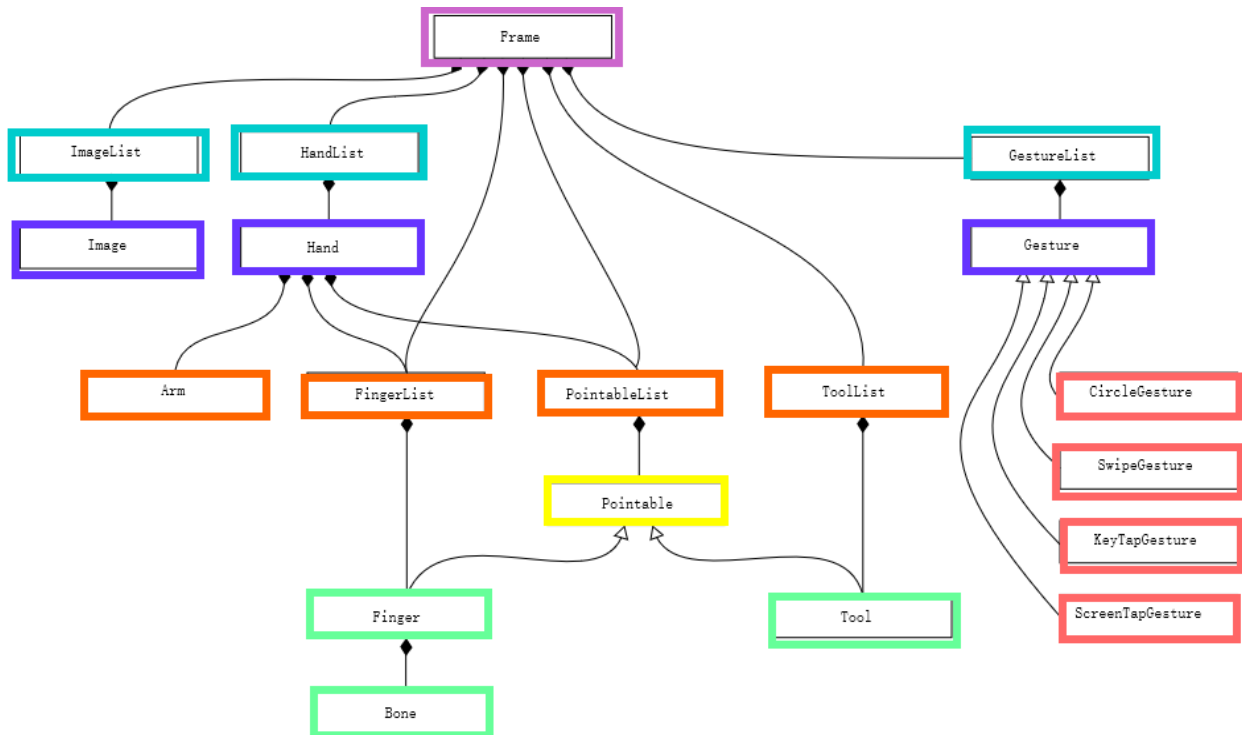
Le Contrôleur *Leap Motion* se charge de transférer des données du capteur au logiciel *Leap Motion*. Le contrôleur *controller* sert à configurer la détection de gestes, à retrouver les cadres avec les données détectées par le capteur. Dès sa création, l'application est automatiquement connectée au capteur *Leap Motion* et les données sont disponibles.

Un système d'écouteur prend en charge différents évènements dont les principaux sont :

- *onConnect (Controller)* : cette méthode est invoquée au moment où l'interface de programmation est connectée au capteur *Leap Motion*. Elle permet principalement de spécifier les différents gestes auxquels l'écouteur sera confronté ;
- *onFrame (Controller)* : cette méthode est invoquée à chaque mesure de capteur et va nous permettre d'analyser les informations à raison de trente images par seconde.

L'interface de programmation permet aux développeurs de disposer des données du contrôleur reprises dans des classes définies. La classe principale est la classe *Frame* et est le point de départ de la reconnaissance des données de la main, des doigts, des os, des outils, des gestes et des images. Chaque cadre *Frame* contient une liste des entités reconnues.

La figure ci-dessous représente les classes de la librairie du Leap Motion [<https://developer.leapmotion.com/documentation>]⁹⁷



Les six types de données de base obtenues avec le contrôleur *Leap Motion* sont les suivantes :

1. les données des mains

- le type de main : droite ou gauche (*String*) ;
- la position de la paume : le centre de la paume (*mm* (millimètres) (*float x, float y, float z*)) ;
- le vecteur normal : celui normal au plan fait par les doigts ;
- le vecteur de direction : celui parallèle à la direction des quatre doigts ;
- les facteurs de mouvement : les échelles relatives, les facteurs de rotation et de translation entre deux mouvements ;
- la vitesse de la paume en *mm/sec* ;
- la force de préhension.

Le modèle de reconnaissance de la main est optimal quand ses cinq doigts sont visibles. Autrement, il travaille par prédiction.

2. les données des doigts

L'objet Doigt *Finger* appartient à la classe *Pointable Object* et partage des caractéristiques similaires à l'objet Outil *Tool* :

- l'identifiant ID du doigt ;
- la position du doigt (*mm* (millimètres) (*float x, float y, float z*));
- la direction du doigt donnée par un vecteur ;
- le type de doigt : pouce, index, majeur, annuaire, auriculaire.

A partir de l'objet doigt *Finger*, il est possible de déduire les données des os de l'utilisateur.

Les os *Bone* sont automatiquement classés en quatre classes :

- *Distal Phalanges* : phalanges distales ;
- *Intermediate Phalanges* : phalanges intermédiaires ;
- *Proximal Phalanges* : phalanges proximales ;
- *Metacarpals* : métacarpes.

3. Les données des outils

Les éléments préhensibles comme un stylo, un crayon sont considérés comme des outils par l'API du *Leap Motion*. Les outils et les doigts sont des objets *pointable*. Les propriétés qui rendent les outils distincts des doigts sont :

- la longueur ;
- la finesse ;
- la droiture.

4. Les gestes reconnus par l'interface de programmation API

Le cercle *CircleGesture* : un mouvement circulaire de la main au-dessus du *Leap Motion*.

Les paramètres du cercle sont :

- le sens : le sens des aiguilles d'une montre ou le sens opposé ;
- le rayon du cercle tracé ;
- le vecteur normalisé ;
- le nombre de tours ;
- la position du centre ;
- le temps mis pour le réaliser.

Le balayage *SwipeGesture* : un mouvement horizontal ou vertical de balayage de la main au-dessus du *Leap Motion*.

Les paramètres du balayage sont :

- la position du milieu du balayage ;
- la direction ;
- la vitesse en *mm/sec* ;
- le temps mis pour le réaliser.

Le toucher d'écran *ScreenTapGesture* : un mouvement court de pointage avec un doigt vers l'écran comme le toucher d'un écran tactile au-dessus du *Leap Motion*.

Les paramètres du toucher d'écran sont :

- la position ;
- la direction ;
- le temps mis pour le réaliser.

Le toucher de clavier *KeyTapGesture* : un mouvement vers le bas, comme en tapant sur un clavier au-dessus du *Leap Motion*.

Les paramètres du toucher de clavier sont :

- la position ;
- la direction ;
- le temps mis pour le réaliser.

5. La reconnaissance de mouvements spécifiques

Les mouvements reconnaissables par l'interface de programmation API du *Leap Motion* sont de deux types : les mouvements d'objets dans tout un cadre et les mouvements d'une seule main.

Pour ces deux types de mouvements l'interface de programmation API peut détecter :

- une translation : un mouvement linéaire dans les trois dimensions ;
- l'échelle : un mouvement d'expansion ou de contraction relative ;
- la rotation : un mouvement de changement angulaire dans les trois dimensions.

Une translation dans un cadre reflète le changement moyen de position de tous les objets dans la scène. Une translation d'une main reflète le changement de sa position quand elle bouge.

L'échelle d'un cadre reflète le mouvement des objets de la scène qui se rapprochent ou s'éloignent les uns des autres. L'échelle d'une main reflète le changement de l'écart entre les doigts.

La rotation d'un cadre reflète le mouvement différentiel entre les objets de la scène. La rotation d'une main reflète le changement de l'orientation d'une unique main.

6. Obtenir des images brutes des capteurs des caméras du *Leap Motion*

Il est possible d'obtenir deux images brutes des caméras infrarouges à partir de l'interface de programmation du *Leap Motion*. L'image de gauche est indexée à 0 et celle de droite à 1.

Paramétrage

L'origine du système de coordonnées x,y,z se situe au centre du contrôleur *Leap Motion*.

Les axes utilisés sont les axes x,y,z ou seulement les axes x,y (pas de profondeur z).

Les unités du système de coordonnées sont en pixel ou en mm.

Il est nécessaire d'établir l'échelle correspondante à son application par exemple combien de pixels au mm comporte l'application 2D.

5.2 Evaluation de l'interface de programmation API Java du Leap Motion

Le but de cette section est de présenter une analyse personnelle des capacités de l'interface de programmation applicative (API) du capteur *Leap Motion*. Cette analyse consiste en une version résumée et individuelle d'éléments factuels présentés dans une quinzaine de références afin d'offrir une vision globale et synthétique des facultés des interfaces de programmation applicatives (API) du capteur *Leap Motion*.

La précision du *Leap Motion* est de l'ordre de 0.01 mm[<https://developer.leapmotion.com>]⁹⁸ (mesure constructeur). Le *Leap Motion* est sujet à des problèmes d'occlusion [Shao, 2016]⁹⁹. Il n'y a pas de garantie que tous les doigts soient détectés quand ils se trouvent proches du capteur. Si les doigts sont cachés par la main ou un doigt d'une autre main, une gigue est créée et la latence (le délai de transmission) varie. La référence de position est perdue et conduit à un comportement erratique. Ce comportement est aussi présent quand tous les doigts sont près les uns des autres et que la caméra ne peut identifier clairement le début et la fin de chaque doigt.

Le logiciel s'est amélioré pour optimiser les performances et quelques limitations [<https://developer.leapmotion.com/documentation>]¹⁰⁰. Cette nouvelle version est dotée d'une reconnaissance du squelette de la main et des avant-bras. Dans cette version, la main est toujours affichée avec cinq doigts même s'ils ne sont pas détectés, la rotation de 180 ° d'une main est correctement interprétée et le contrôleur peut identifier une main gauche d'une main droite.

En général, pour être efficace, le *Leap Motion* nécessite un calibrage [Guna et al, 2014]¹⁰¹. Ce calibrage est réalisé par un logiciel au moyen d'un capteur de plus grande précision. La latence optimale du *Leap Motion* est de 12 millisecondes [blog.leapmotion.com]¹⁰². Le transfert des données entre l'application et l'écran peut constituer un point d'étranglement. Les sources externes de lumière infrarouge peuvent contrecarrer le bon fonctionnement du système et le placer dans un mode dégradé où toutes les fonctionnalités ne sont pas permises. Il y a des occurrences de faux positifs (reconnaissance de geste là où il n'y en a pas), de faux négatifs (non-reconnaissance de geste là où il existe) et des déviations de position. Les occurrences de faux négatifs dépendent fortement du type de geste pratiqué. Il en résulte de mauvaises orientations des doigts et des mains en trois dimensions et des positions en deux dimensions. Une superposition des doigts ne permet pas une identification de ceux-ci. Un mouvement de pincement des doigts est difficile à reconnaître et nécessite un apprentissage de l'utilisateur. Le *Leap Motion* peut quand même travailler dans de faibles conditions d'éclairage mais le suivi et la précision peuvent en dépendre. De manière générale, la reconnaissance de gestes statiques est plus fiable : la mesure de la précision en statique est de l'ordre de 0,2 mm [Weichert et al, 2013]¹⁰³. Elle nécessite cependant l'emploi d'un modèle éprouvé et entraîné. Le suivi d'objet par cheminement dynamique présente une précision de l'ordre de 1,2 mm [Weichert et al, 2013]¹⁰⁴(moyenne).

La cohérence des données et la performance du contrôleur dépendent de l'endroit où les données ont été enregistrées [Shao, 2016]¹⁰⁵ tant dans le cas d'une reconnaissance statique que d'une reconnaissance dynamique de gestes. Si les doigts se déplacent en dehors du champ de vision du *Leap Motion*, ils sont assignés à un nouvel identifiant. Cela arrive également quand un doigt en cache un autre. L'occlusion des doigts de la main crée une fluctuation et une perte du signal. Des données inexactes ont été décelées à une distance de plus ou moins 250 mm du capteur *Leap Motion*. Les

données de suivi des mains deviennent instables quand les mains se trouvent aux frontières du volume de détection.

Une limitation importante du contrôleur est l'incohérence de sa fréquence d'échantillonnage [Guna et al, 2014]¹⁰⁶. Sa valeur moyenne (relativement basse) est de l'ordre de 40 Hz mais varie fortement tant en reconnaissance statique que dynamique. La difficulté est de synchroniser le contrôleur avec d'autres systèmes en temps réel.

Le contrôleur *Leap Motion* n'est pas assez fiable pour être utilisé dans un système prototype [Tung et al, 2015]¹⁰⁷. Le *Leap Motion* est conçu au départ pour être utilisé sur un bureau et reconnaître les gestes dans son volume d'interaction. Le taux de performance du cadre (*frame*) peut être une limitation. Elle est due à l'aberration chromatique et à la distorsion. Les objets transparents (par exemple un stylo entre les doigts) ne sont pas correctement détectés par le capteur *Leap Motion*.

L'orientation du capteur *Leap Motion* par rapport aux mains, aux doigts et aux outils est importante [Han et al, 2014]¹⁰⁸. Lorsque les mains se trouvent au-dessus du capteur *Leap Motion*, la détection fonctionne en général correctement. Si les doigts sont trop proches les uns des autres, ils ne sont pas reconnus. Une distance de 5 mm entre les doigts est nécessaire pour garantir une reconnaissance correcte. Lorsque le capteur *Leap Motion* a tourné de 90 degrés par rapport aux mains, les doigts ne sont pas reconnus. La détection décroît à trois cadres par seconde pour une main statique. Les paumes sont reconnues mais il apparaît un fréquent changement de direction. Il disparaît à une distance de 40 cm du capteur. Les outils (stylo) sont reconnus s'ils sont complètement perpendiculaires au capteur. Si le capteur est placé sur son flanc devant les mains, les doigts et les mains ne sont pas reconnus sauf s'il y a un objet solide au-dessus du capteur ou si la lumière de la pièce est très faible. Si le capteur est placé sur son flanc et que les doigts sont placés sur la table, ils sont reconnus. Ils ne le sont pas s'ils sont légèrement plus hauts que le capteur. Si le capteur forme un angle avec la table, il fonctionne correctement mais il peut y avoir de mauvaises détections.

La position des doigts dans les trois dimensions est précise et correspond aux attentes mais il y a toujours une latence induite entre le mouvement et son contrôle. La position de la paume de la main est déduite de la position des doigts et est donc logiquement moins fiable. La détection de la paume est incomplète si les doigts ne sont pas complètement détectés ou si les conditions d'éclairage sont moins bonnes.

La détection du mouvement de la main d'avant en arrière est bonne de même que les déplacements de haut en bas et de gauche à droite. L'absence de la main est aussi assez bien détectée. Le lacet de la main (déplacement à plat de gauche à droite) est très mal détecté et entraîne une gigue importante. Le rayon de la main n'est pas accessible rapidement. La stabilisation des doigts est utile pour une expérience utilisateur moins répondante mais plus uniforme.

La performance du *Leap Motion* pour les tâches de pointage [Bachmann et al, 2014]¹⁰⁹ est moins bonne que celle d'une souris. Pour les tâches de pointage, le taux d'erreur est de 7,8 % tandis que celui de la souris est de 2,8%.

Des paramètres dans un logiciel peuvent être associés aux dimensions réelles de la main. S'ils ne correspondent pas cela peut causer une défaillance.

Il est possible d'augmenter le volume d'interaction en utilisant plusieurs capteurs *Leap Motion* [Jin et al, 2016]¹¹⁰. Cela procure plus de précision et un suivi continu des données des mains. Les problèmes

d'occlusion sont limités car s'ils apparaissent pour un capteur le deuxième prend le relais. L'étude a montré que la disposition optimum pour le suivi des mains est celle dont l'angle formé par les capteurs est de 120°.

Les données fournies directement par l'API sont des données de positions : extrémités des doigts ; centre de la paume (déduit des positions des doigts), des orientations sous la forme de vecteurs. Il n'y a donc pas beaucoup de données disponibles pour la reconnaissance de gestes.

Evaluation de l'interface de programmation applicative (API) du *Leap Motion* pour les applications en local [<https://developer.leapmotion.com/documentation>]¹¹¹

Forces de l'API :

- les détails fournis par l'API sont précis ;
- l'API donne un accès direct aux données des doigts et des mains ;
- Il n'y a pas besoin d'interpréter un format brut des données.

Faiblesse de l'API :

- la précision et la fidélité de détection de la main ne sont pas maintenues si la main effectue une rotation (de parallèle à perpendiculaire) au-dessus de la surface du contrôleur ;
- impossibilité de détecter une main perpendiculaire à la surface du contrôleur ;
- impossibilité de détecter des doigts trop proches l'un de l'autre ;
- limitation importante si la détection se fait aux bordures du volume d'interaction.

Evaluation de l'interface de programmation applicative (API) du *Leap Motion* pour les applications web [<https://developer.leapmotion.com/javascript>]¹¹²

Forces de l'API :

- les gestes qui proviennent de l'API sont faciles à implémenter ;
- les utilisateurs peuvent interagir d'une seule manière avec leur ordinateur ;
- beaucoup d'applications sont disponibles.

Faiblesse de l'API :

- il n'est pas intrinsèquement compatible avec la plupart des navigateurs ;
- la plupart des gestes sont imprécis et imprévisibles ;
- le nombre de gestes est limité avec un développement web ;
- il n'est pas possible de créer des gestes sur mesure.

Opportunités de l'API :

- il existe beaucoup de possibilité d'amélioration ;
- la technologie est nouvelle et peut remplacer potentiellement le toucher d'écran.

Le SDK Javascript dans sa forme actuelle est assez limité pour le développement et pour une expérience utilisateur.

5.3 Programmes Java Leap Motion

Plusieurs programmes java ont été conçus afin de tester les capacités de reconnaissance du contrôleur *Leap Motion*.

Le premier programme a pour but de tester le contrôleur (*controller*) et l'écouteur (*leaplistener*) du *Leap Motion* et de donner quelques paramètres du cadre.

Dans le programme principal de nouveaux objets écouteur (*leaplistener*) et contrôleur (*controller*) sont créés. Au contrôleur créé, il est adjoint le nouvel écouteur (*listener*). Si la touche *enter* est enfoncée : « *Press enter to quit* », le contrôleur est détaché de l'écouteur (*listener*) et l'application prend fin.

Dans la classe écouteur (*leaplistener*), le contrôleur (*controller*) est initialisé par la méthode *onInit* puis la méthode *onConnect* connecte le capteur et le contrôleur permet la détection de gestes « *Connected to Motion Sensor* ». La méthode *on Disconnect* déconnecte le contrôleur « *Motion Sensor disconnected* ». La méthode *onExit* permet la sortie du contrôleur « *Exited* ».

Le contrôleur est adjoint au cadre de référence grâce à la méthode *onFrame*.

Pour chaque cadre, les paramètres suivants sont imprimés :

- l'identification du cadre (*Frame id*) ;
- le nombre de mains (*Hands*) ;
- le nombre de doigts (*Fingers*) ;
- le nombre d'outils (*Tools*) ;
- le nombre de gestes (*Gestures*).

Le second programme teste la reconnaissance des paramètres principaux de la main. Pour chaque cadre, le programme affiche la détection d'une main droite ou d'une main gauche. Les paramètres d'identifiant de la main détectée et de position tridimensionnelle de la paume de la main sont également imprimés.

Le troisième programme teste la reconnaissance des gestes prédéfinis de l'interface de programmation API du *Leap Motion*.

Ces gestes sont : le « *Swipe* » (geste de balayage), le « *Circle* » (réalisation d'un cercle), le « *Screen Tap* » (geste de toucher d'écran) et le « *Key Tap* » (geste de toucher de touche).

Pour le cercle, le programme identifie s'il se fait dans un sens horlogique ou dans un sens anti-horlogique. Ensuite, il imprime l'identifiant du cercle, l'état du cercle (est-il ou non démarré), la progression du cercle, le rayon du cercle, l'angle en radians et le sens horlogique.

Pour le balayage, le programme imprime l'identifiant, l'état, la position, la direction et la vitesse.

Pour le toucher d'écran, le programme imprime l'identifiant, l'état, la position et la direction.

Pour le toucher de touche, le programme imprime l'identifiant, l'état, la position et direction.

En outre, si le programme n'identifie aucun des gestes de base, il l'indique.

Le quatrième programme utilise les gestes de base pour interagir avec l'écran et la carte.

L'index s'il est détecté, permet de visualiser une position sur l'écran (comme la flèche de la souris).

Pour se faire, le volume d'interaction est défini et la position de l'index est stabilisée.

Le cercle (*Circle*) est associé au zoom dans le sens horlogique et au dézoom dans le sens anti-horlogique. Il est possible de l'utiliser pour une interaction sur la carte.

Le balayage (*Swipe*) est associé à l'ouverture d'une fenêtre Windows.

Le toucher d'écran (*Screen Tap*) est programmé pour interagir avec un bouton sur l'écran : l'utilisateur le sélectionne en positionnant l'index au-dessus, ensuite il appuie sur ce bouton puis le relâche en déplaçant l'index.

5.4 Tests des programmes Java Leap Motion

Tests des programmes Java Leap Motion			
Programme	Fonctions et données affichées	Tests	Résultats
Tous les programmes	Fonctions de mise en route et arrêt : connexion et déconnexion.	Test des fonctions par initialisation de l'appareil, par connexion et déconnexion, par la touche entrée pour la sortie.	Plusieurs essais ont été réalisés pour le test des fonctions à chaque fois, il a été concluant.
Programme général	Données : identifiant du cadre, temps nombre de mains, nombre de doigts, nombre d'outils, nombre de gestes.	Démarrage de l'application, placement d'une ou deux mains, doigts tendus ou pas, réalisation des gestes de base, placement d'outil (stylo), sortie de l'application.	Plusieurs essais ont été réalisés. L'identifiant du cadre et le temps s'affichent correctement. Si l'application est déconnectée, l'identifiant du cadre et le temps sont réinitialisés. Le programme affiche correctement le nombre de mains présentes qu'il s'agisse d'une main droite ou d'une main gauche mais avec un décalage temporel. Le nombre de doigts qui s'affiche n'est pas correct. Si une main est détectée le nombre de doigts est toujours de 5 et si les deux mains sont détectées alors nombre de doigts est toujours de 10 que les doigts soient tendus ou pas. L'outil n'est jamais détecté. Certains gestes sont détectés d'autres jamais. Les gestes de toucher d'écran et de touches ne sont jamais détectés.
Programme de détection des mains	Données : main droite, main gauche, identifiant et position de la paume.	Démarrage de l'application, placement d'une main droite ou d'une main gauche.	Le type de main et la position de la paume sont correctement affichés avec un décalage temporel.

Tests des programmes Java Leap Motion

Programme	Fonctions et données affichées	Tests	Résultats
Programme de détection des gestes	Données : type de geste; cercle : sens, état, rayon, angle ; balayage : position, direction, vitesse; toucher de touche et toucher d'écran : état, position, direction.	Démarrage de l'application, réalisation des différents gestes : cercle, balayage, toucher d'écran et de touche.	Les touchers d'écran et de touche ne sont pas identifiés. Le cercle est identifié et les données sont affichées. Le balayage est identifié mais dépend de la vitesse, il y a souvent des défauts. Parfois un balayage est identifié à la place d'un cercle. Il y a des données incohérentes.
Programme d'interaction avec la carte	Fonctions : position à l'écran de l'index, zoom et dé zoom de la carte par un cercle, toucher de touche, balayage pour introduire un menu, interaction avec le toucher d'écran.	Démarrage de l'application et tests des différentes fonctions implémentées.	La position à l'écran de l'index se réalise correctement de même que le zoom et le dé zoom de la carte. Le toucher de touche et d'écran ne fonctionne pas. Le balayage ouvre un menu à l'écran mais il faut répéter l'opération quelques fois.

6. Etude de cas (Use case)

L'étude de cas vise à permettre l'application des connaissances acquises sur le *Leap Motion* et sur son interface de programmation API par la résolution d'un problème concret. Les conditions de l'efficacité de l'étude de cas reposent sur la qualité du problème choisi. L'étude de cas illustre les capacités de reconnaissance de gestes du capteur *Leap Motion*.

Les avantages d'une étude de cas sont les suivants :

- elle discrète ;
- il n'y a pas d'intervention dans le déroulement des événements ;
- il est possible d'étudier le cas sous différentes facettes ;
- elle est peu coûteuse ;
- elle permet le partage d'un savoir professionnel ;
- la collecte des données ne se limite pas à des faits quantifiables ;
- une analyse statistique est possible ;
- elle est bien adaptée à l'appréhension de systèmes complexes ;
- elle peut être utile à de nouvelles hypothèses de recherche.

Les différentes étapes de l'étude de cas sont :

- Etablir la pertinence de l'étude de cas c'est-à-dire vérifier que la problématique du *Leap Motion* se prête bien au cas choisi : connaissance des utilisateurs et de l'interaction, compréhension de l'appareil et des bases théoriques ;
- Assurer la véracité des résultats : ils sont rigoureux, méthodiques, fiables et représentatifs de la réalité observée. Eviter les erreurs d'interprétation et les interférences pour obtenir des résultats valides ;
- Préparer et présenter l'étude de cas aux utilisateurs ;
- Laisser chaque utilisateur s'exprimer librement face questions posées par l'étude de cas et collecter les données. Etablir une relation de confiance entre l'observateur et les participants ;
- Traiter les données récoltées : déduire des informations significatives, faire des tests de concordance, vérifier la fiabilité et la qualité des données ;
- Interpréter les données : conceptualiser et rechercher des constatations simples à retenir et utiles à la compréhension et à la résolution d'une situation de même genre.

L'étude de cas est créée de manière à identifier les éléments importants dans le cadre du développement d'une application d'interaction par les gestes avec le *Leap Motion*.

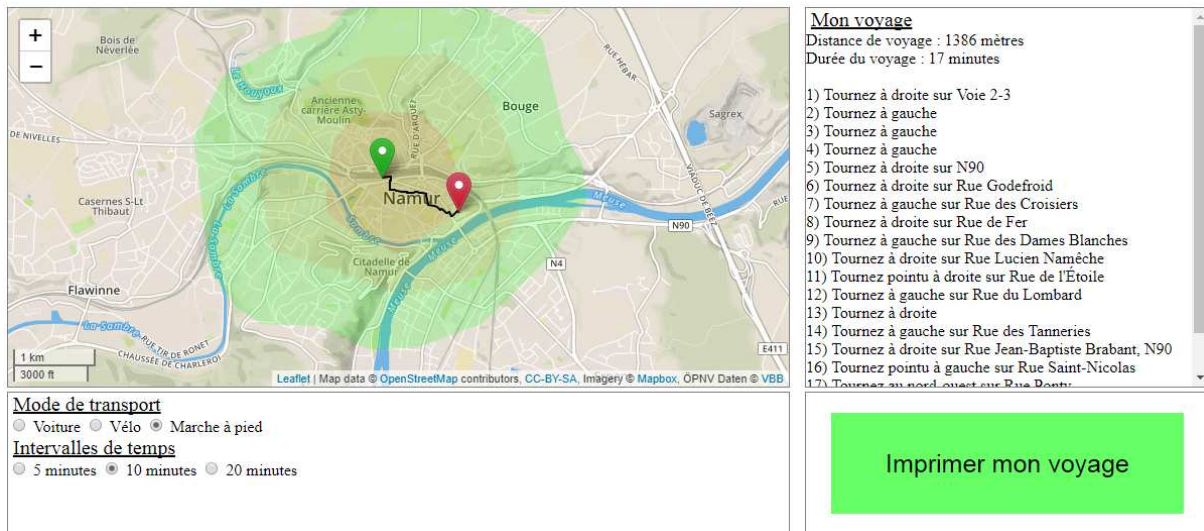
L'étude de cas consiste à déplacer une carte dans les quatre directions, à la zoomer et à la dézoomer.

Le choix de l'étude est motivé par le fait de pouvoir réaliser une enquête utilisateur portant sur des personnes de condition sociale, de profession, d'âge et de sexe divers. Le but de l'enquête est d'identifier les gestes les plus intuitifs et d'établir un catalogue de gestes qui répondent à telle ou telle opération sur la carte.

Enfin, les gestes les plus utilisés d'après l'enquête utilisateur ont été implémentés de manière à obtenir une interface utilisateur naturelle qui rend l'interaction gestuelle plus acceptable.

6.1 Méthodologie de l'enquête avec utilisateurs

L'image PNG « travaux de mobilité » reprenant la carte de Namur avec les points de départ et d'arrivée du voyage (en haut à gauche), le descriptif du voyage (en haut à droite), le choix du mode de transport et des intervalles de temps (en bas à droite), le bouton d'impression (en bas à gauche) a été présentée à chaque utilisateur potentiel sur un ordinateur portable à l'aide de la visionneuse de photos de Windows. Cette image provient des travaux d'Antoine Clarinval chercheur à UNamur.



Le panel des utilisateurs potentiels comprend quinze hommes et quinze femmes d'âge et de profession divers.

Chacun des utilisateurs a reçu les explications suivantes :

- La carte présentée se trouve sur une borne interactive dans la gare de Namur ;
- Il faut déplacer cette carte avec des gestes de la main ou des mains vers le haut, le bas, la droite et la gauche ;
- L'utilisateur ne peut ni toucher l'écran ni utiliser le clavier ou le pavé tactile pour effectuer ces déplacements ;
- Le but de l'étude est de déceler puis de programmer les gestes les plus intuitifs afin qu'ils soient détectés par un capteur de reconnaissance de gestes pour ensuite effectuer le déplacement de l'image dans la direction indiquée ;
- Ce programme n'est pas encore implémenté dans l'ordinateur ;
- La zone d'interaction dans laquelle se font les gestes est décrite comme un rectangle situé devant et dix centimètres au-dessus du clavier de l'ordinateur portable avec les dimensions suivantes : quinze centimètres de profondeur, vingt-cinq centimètres de largeur et de hauteur ;
- L'ordinateur ne dispose pas du logiciel pour interagir avec les gestes de la main ou des mains.

Ensuite, à l'aide du clavier, l'image a été déplacée dans chacune des directions afin de montrer à l'utilisateur les résultats souhaités dus à l'interaction des gestes.

Aux utilisateurs qui n'ont pas compris exactement ce qu'il est attendu, une vidéo a été présentée. Elle montre à gauche de l'écran le squelette de reconnaissance des mains dans un casque de réalité virtuelle et à droite de l'écran les mains effectuant des gestes dans l'air.

Après avoir noté les gestes des utilisateurs pour déplacer l'image, les utilisateurs ont montré leurs gestes pour la zoomer et la dézoomer. La zone d'interaction est la même que celle des déplacements et le programme n'est pas non plus implémenté.

A l'aide du bouton loupe de la visionneuse, l'image a été zoomée et inversement dézoomée afin de montrer à l'utilisateur le résultat attendu.

La description des gestes de zoom et de dé zoom de chaque utilisateur a été notée.

Le tableau récapitulatif ci-après reprend les données suivantes : numéro de l'utilisateur testeur, prénom, âge, sexe, profession, gestes de déplacement de la carte dans chaque direction, gestes de zoom et de dézoom de chaque utilisateur. Les données sont classées chronologiquement par un numéro correspondant à la date d'acquisition des données de chaque utilisateur testeur.

6.2 Enquête utilisateur magicien d'Oz

Enquête utilisateur magicien d'Oz

Numéro	Prénom	Age	Sexe	Profession	Geste de déplacement de la carte vers le haut	Geste de déplacement de la carte vers le bas	Geste de déplacement de la carte vers la gauche	Geste de déplacement de la carte vers la droite	Geste du zoom de la carte	Geste du dézoom de la carte
1	Eric L.	54	Masculin	Dessinateur	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main droite puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	De la main droite fermée à la main droite ouverte	De la main droite ouverte à la main droite fermée
2	Olivier W.	51	Masculin	Manager	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main droite puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement
3	Marie-T. G.	82	Féminin	Retraitée	Déplacement de la main droite verticale vers le haut	Déplacement de la main droite verticale vers le bas	Balayage de la main droite vers la gauche	Balayage de la main droite vers la droite	De la main droite fermée à la main droite ouverte	De la main droite ouverte à la main droite fermée
4	Josy A.	58	Masculin	Dessinateur	Pointage avec l'index et le majeur de la main droite puis déplacement de ceux-ci vers le haut	Pointage avec l'index et le majeur de la main droite puis déplacement de ceux-ci vers le bas	Pointage avec l'index et le majeur de la main droite puis déplacement de ceux-ci vers la gauche	Pointage avec l'index et le majeur de la main droite puis déplacement de ceux-ci vers la droite	De la main droite fermée à la main droite ouverte	De la main droite ouverte à la main droite fermée
5	Colette L.	50	Féminin	Dessinatrice	Déplacement de la main droite horizontale vers le haut	Déplacement de la main droite horizontale vers le bas	Balayage de la main droite vers la gauche	Balayage de la main droite vers la droite	Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement
6	Laurent M.	51	Masculin	Ouvrier	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main droite puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	Les deux mains jointes s'écartent progressivement pour zoomer	Les deux mains écartées se rapprochent progressivement pour dézoomer
7	Bernard D.	61	Masculin	Architecte	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main droite puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	De la main droite fermée à la main droite ouverte	De la main droite ouverte à la main droite fermée
8	David P.	43	Masculin	Ouvrier	Déplacement de la main droite horizontale vers le haut	Déplacement de la main droite horizontale	Balayage de la main droite vers la gauche	Balayage de la main droite vers la droite	Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement
9	Axelle G.	40	Féminin	Secrétaire handicap main droite	Index et pouce écartés de la main gauche puis déplacement de la main vers le haut	Index et pouce écartés de la main gauche puis déplacement de la main vers le bas	Index et pouce écartés de la main gauche puis déplacement de la main vers la gauche	Index et pouce écartés de la main gauche puis déplacement de la main vers la droite	De la main droite fermée à la main droite ouverte	De la main droite ouverte à la main droite fermée
10	Cédric T.	9	Masculin	Elève	Déplacement de la main droite verticale vers le haut	Déplacement de la main droite verticale vers le bas	Balayage de la main droite vers la gauche	Balayage de la main droite vers la droite	Les deux mains jointes s'écartent progressivement pour zoomer	Les deux mains se rapprochent progressivement pour dézoomer
11	Guillaume T.	7	Masculin	Elève	Déplacement de la main droite horizontale vers le haut	Déplacement de la main droite horizontale vers le bas	Balayage de la main droite vers la gauche	Balayage de la main droite vers la droite	Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement
12	Fabien T.	43	Masculin	Employé de banque	Déplacement de la main droite verticale vers le haut	Déplacement de la main droite verticale vers le bas	Balayage de la main droite vers la gauche	Balayage de la main droite vers la droite	La main droite verticale se rapproche de l'écran	La main droite verticale s'éloigne de l'écran
13	Anne-C. W.	42	Féminin	Architecte	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main droite puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	De la main droite fermée à la main droite ouverte	De la main droite ouverte à la main droite fermée
14	Dominique L.	48	Masculin	Architecte	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main droite puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement
15	Béatrice S.	47	Féminin	Secrétaire	Déplacement de la main droite verticale vers le haut	Déplacement de la main droite verticale vers le bas	Balayage de la main droite vers la gauche	Balayage de la main droite vers la droite	De la main droite fermée à la main droite ouverte	De la main droite ouverte à la main droite fermée

Enquête utilisateur magicien d'Oz

Numéro	Prénom	Age	Sexe	Profession	Geste de déplacement de la carte vers le haut	Geste de déplacement de la carte vers le bas	Geste de déplacement de la carte vers la gauche	Geste de déplacement de la carte vers la droite	Geste du zoom de la carte	Geste du dézoom de la carte
16	Christian S.	46	Masculin	Ouvrier	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main droite puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement
17	Joëlle H.	48	Féminin	Fonctionnaire	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main gauche puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	De la main droite fermée à la main droite ouverte	De la main droite ouverte à la main droite fermée
18	Magali D.	42	Féminin	Employée	Déplacement de la main droite verticale vers le haut	Déplacement de la main droite verticale vers le bas	Balayage de la main gauche vers la gauche	Balayage de la main droite vers la droite	Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement
19	Cathy M.	50	Féminin	Dessinatrice	Pointage avec le pouce main droite puis déplacement de celui-ci vers le haut	Pointage avec le pouce main droite puis déplacement de celui-ci vers le bas	Pointage avec le pouce main droite puis déplacement de celui-ci vers la gauche	Pointage avec le pouce main droite puis déplacement de celui-ci vers la droite	De la main droite fermée à la main droite ouverte	De la main droite ouverte à la main droite fermée
20	Céline N.	26	Féminin	Architecte d'intérieur	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main gauche puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement
21	Liliane F.	60	Féminin	Architecte	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main gauche puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	Les deux index s'écartent progressivement pour zoomer	Les deux index se rejoignent progressivement pour dézoomer
22	Jean M.	57	Masculin	Ingénieur	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main gauche puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	De la main droite fermée à la main droite ouverte	De la main droite ouverte à la main droite fermée
23	Jean-Charles L.	34	Masculin	Ingénieur	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main gauche puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	De la main droite fermée à la main droite ouverte	De la main droite ouverte à la main droite fermée
24	Laurent F.	32	Masculin	Ingénieur	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main gauche puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement
25	Christophe F.	34	Masculin	Ingénieur	L'index et le pouce des deux mains cadrent l'image puis déplacement de ceux-ci vers le haut	L'index et le pouce des deux mains cadrent l'image puis déplacement de ceux-ci vers le bas	L'index et le pouce des deux mains cadrent l'image puis déplacement de ceux-ci vers la gauche	L'index et le pouce des deux mains cadrent l'image puis déplacement de ceux-ci vers la droite	L'index et le pouce des deux mains cadrent l'image puis s'écartent progressivement	L'index et le pouce des deux mains cadrent l'image puis se rejoignent progressivement
26	Anne-L. O.	57	Féminin	Employée	Pointage avec le majeur main gauche puis déplacement de celui-ci vers le haut	Pointage avec le majeur main gauche puis déplacement de celui-ci vers le bas	Pointage avec le majeur main gauche puis déplacement de celui-ci vers la gauche	Pointage avec le majeur main gauche puis déplacement de celui-ci vers la droite	Le majeur et l'index de la main gauche fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main gauche fermée sont écartés puis se rejoignent progressivement
27	Flore D.	33	Féminin	Employée	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main gauche puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement
28	Laetitia W.	26	Féminin	Employée	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main gauche puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement
29	Fabienne N.	58	Féminin	Employée	Déplacement de la main droite horizontale vers le haut	Déplacement de la main droite horizontale vers le bas	Balayage de la main droite vers la gauche	Balayage de la main droite vers la droite	La main droite verticale se rapproche de l'écran	La main droite verticale s'éloigne de l'écran
30	Fabienne L.	46	Féminin	Employée	Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	Pointage avec l'index main gauche puis déplacement de celui-ci vers la gauche	Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement

Geste de déplacement de la carte vers le haut	Nombre	Pourcentage du total	Pourcentage total hommes	Pourcentage total femmes	Moyenne âge du total	Moyenne âge des hommes	Moyenne âge des femmes
Pointage avec l'index main droite puis déplacement de celui-ci vers le haut	16	53,33%	60,00%	46,67%	45	48	40
Déplacement de la main droite verticale vers le haut	5	16,67%	13,33%	20,00%	45	26	57
Déplacement de la main droite horizontale vers le haut	4	13,33%	13,33%	13,33%	40	25	54
Pointage avec l'index et le majeur de la main droite puis déplacement de ceux-ci vers le haut	1	3,33%	6,67%	0,00%	58	58	-
Index et pouce écartés de la main gauche puis déplacement de la main vers le haut	1	3,33%	0,00%	6,67%	40	-	40
Pointage avec le pouce main droite puis déplacement de celui-ci vers le haut	1	3,33%	0,00%	6,67%	50	-	50
L'index et le pouce des deux mains cadrent l'image puis déplacement de ceux-ci vers le haut	1	3,33%	6,67%	0,00%	34	34	-
Pointage avec le majeur main gauche puis déplacement de celui-ci vers le haut	1	3,33%	0,00%	6,67%	57	-	57
Total	30	100,00%	100,00%	100,00%	44,5	42	47

Geste de déplacement de la carte vers le bas	Nombre	Pourcentage du total	Pourcentage total hommes	Pourcentage total femmes	Moyenne âge du total	Moyenne âge des hommes	Moyenne âge des femmes
Pointage avec l'index main droite puis déplacement de celui-ci vers le bas	16	53,33%	60,00%	46,67%	45	48	40
Déplacement de la main droite verticale vers le bas	5	16,67%	13,33%	20,00%	45	26	57
Déplacement de la main droite horizontale vers le bas	4	13,33%	13,33%	13,33%	40	25	54
Pointage avec l'index et le majeur de la main droite puis déplacement de ceux-ci vers le bas	1	3,33%	6,67%	0,00%	58	58	-
Index et pouce écartés de la main gauche puis déplacement de la main vers le bas	1	3,33%	0,00%	6,67%	40	-	40
Pointage avec le pouce main droite puis déplacement de celui-ci vers le bas	1	3,33%	0,00%	6,67%	50	-	50
L'index et le pouce des deux mains cadrent l'image puis déplacement de ceux-ci vers le bas	1	3,33%	6,67%	0,00%	34	34	-
Pointage avec le majeur main gauche puis déplacement de celui-ci vers le bas	1	3,33%	0,00%	6,67%	57	-	57
Total	30	100,00%	100,00%	100,00%	44,5	42	47

Geste de déplacement de la carte vers la gauche	Nombre	Pourcentage du total	Pourcentage total hommes	Pourcentage total femmes	Moyenne âge du total	Moyenne âge des hommes	Moyenne âge des femmes
Pointage avec l'index main droite puis déplacement de celui-ci vers la gauche	16	53,33%	60,00%	46,67%	45	48	40
Balayage de la main droite vers la gauche	9	30,00%	26,67%	33,33%	43	26	56
Pointage avec l'index et le majeur de la main droite puis déplacement de ceux-ci vers la gauche	1	3,33%	6,67%	0,00%	58	58	-
Index et pouce écartés de la main gauche puis déplacement de la main vers la gauche	1	3,33%	0,00%	6,67%	40	-	40
Pointage avec le pouce main droite puis déplacement de celui-ci vers la gauche	1	3,33%	0,00%	6,67%	50	-	50
L'index et le pouce des deux mains cadrent l'image puis déplacement de ceux-ci vers la gauche	1	3,33%	6,67%	0,00%	34	34	-
Pointage avec le majeur main gauche puis déplacement de celui-ci vers la gauche	1	3,33%	0,00%	6,67%	57	-	57
Total	30	100,00%	100,00%	100,00%	44,5	42	47

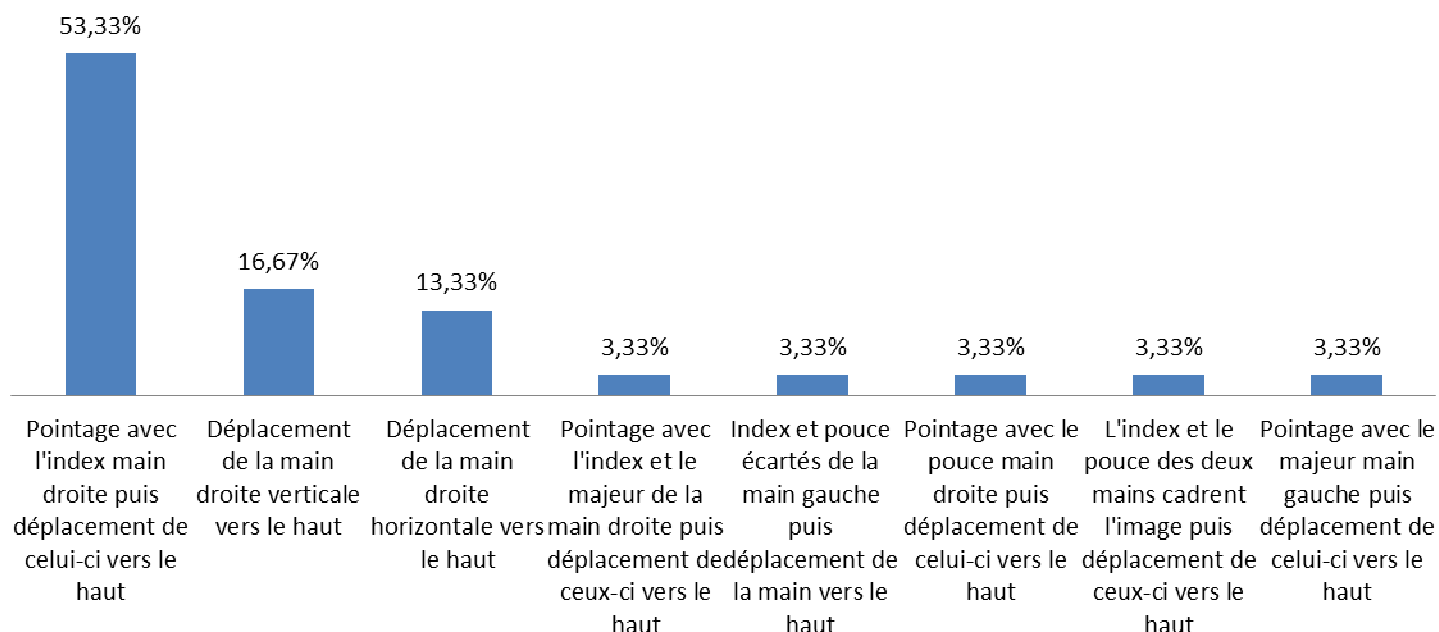
Geste de déplacement de la carte vers la droite	Nombre	Pourcentage du total	Pourcentage total hommes	Pourcentage total femmes	Moyenne âge du total	Moyenne âge des hommes	Moyenne âge des femmes
Pointage avec l'index main droite puis déplacement de celui-ci vers la droite	16	53,33%	60,00%	46,67%	45	48	40
Balayage de la main droite vers la droite	9	30,00%	26,67%	33,33%	43	26	56
Pointage avec l'index et le majeur de la main droite puis déplacement de ceux-ci vers la droite	1	3,33%	6,67%	0,00%	58	58	-
Index et pouce écartés de la main gauche puis déplacement de la main vers la droite	1	3,33%	0,00%	6,67%	40	-	40
Pointage avec le pouce main droite puis déplacement de celui-ci vers la droite	1	3,33%	0,00%	6,67%	50	-	50
L'index et le pouce des deux mains cadrent l'image puis déplacement de ceux-ci vers la droite	1	3,33%	6,67%	0,00%	34	34	-
Pointage avec le majeur main gauche puis déplacement de celui-ci vers la droite	1	3,33%	0,00%	6,67%	57	-	57
Total	30	100,00%	100,00%	100,00%	44,5	42	47

Geste du zoom de la carte	Nombre	Pourcentage du total	Pourcentage total hommes	Pourcentage total femmes	Moyenne âge du total	Moyenne âge des hommes	Moyenne âge des femmes
Le pouce et l'index de la main droite fermée se touchent puis s'écartent progressivement	12	40,00%	40,00%	40,00%	37,5	38	37
De la main droite fermée à la main droite ouverte	11	36,67%	33,33%	40,00%	52	53	51,5
Les deux mains jointent s'écartent progressivement pour zoomer	2	6,67%	13,33%	0,00%	30	30	-
La main droite verticale se rapproche de l'écran	2	6,67%	6,67%	6,67%	50,5	43	58
Les deux index s'écartent progressivement pour zoomer	1	3,33%	0,00%	6,67%	60	-	60
L'index et le pouce des deux mains cadrent l'image puis s'écartent progressivement	1	3,33%	6,67%	0,00%	34	34	-
Le majeur et l'index de la main gauche fermée se touchent puis s'écartent progressivement	1	3,33%	0,00%	6,67%	57	-	57
Total	30	100,00%	100,00%	100,00%	44,5	42	47

Geste du dézoom de la carte	Nombre	Pourcentage du total	Pourcentage total hommes	Pourcentage total femmes	Moyenne âge du total	Moyenne âge des hommes	Moyenne âge des femmes
Le pouce et l'index de la main droite fermée sont écartés puis se rejoignent progressivement	12	40,00%	40,00%	40,00%	37,5	38	37
De la main droite ouverte à la main droite fermée	11	36,67%	33,33%	40,00%	52	53	51,5
Les deux mains écartées se rapprochent progressivement pour dézoomer	2	6,67%	13,33%	0,00%	30	30	-
La main droite verticale s'éloigne de l'écran	2	6,67%	6,67%	6,67%	50,5	43	58
Les deux index se rejoignent progressivement pour dézoomer	1	3,33%	0,00%	6,67%	60	-	60
L'index et le pouce des deux mains cadrent l'image puis se rejoignent progressivement	1	3,33%	6,67%	0,00%	34	34	-
Le pouce et l'index de la main gauche fermée sont écartés puis se rejoignent progressivement	1	3,33%	0,00%	6,67%	57	-	57
Total	30	100,00%	100,00%	100,00%	44,5	42	47

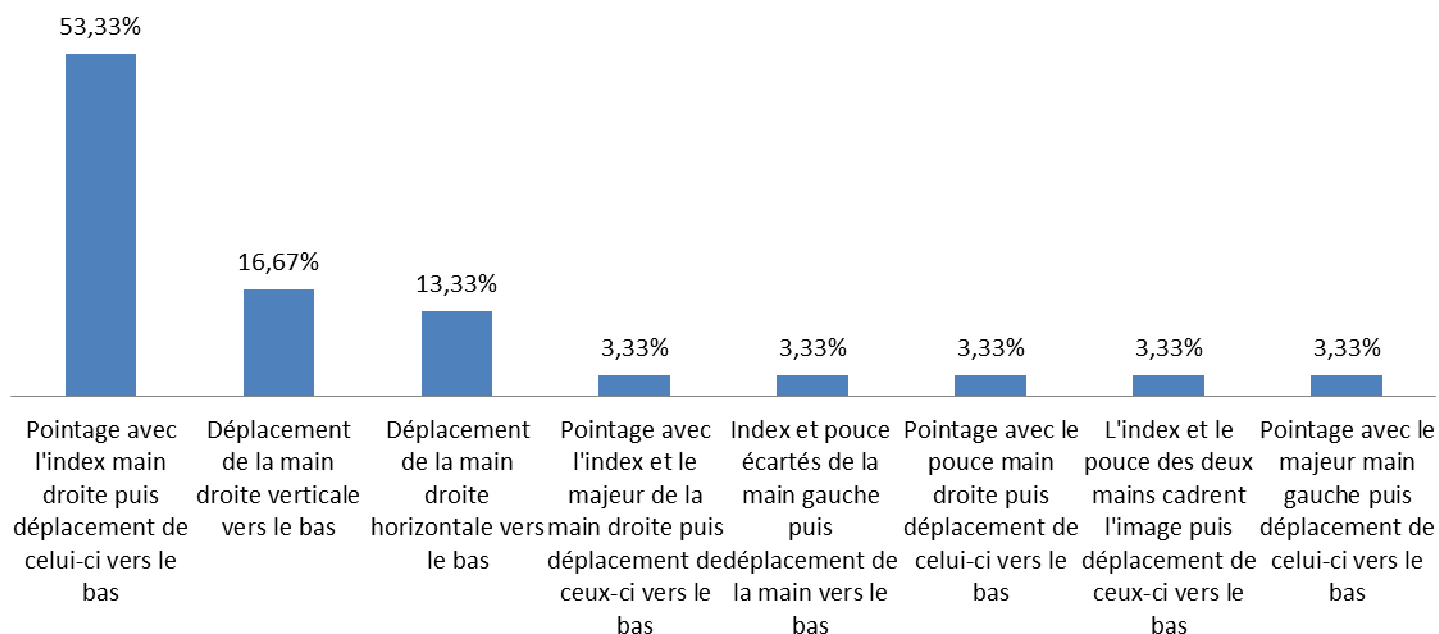
Geste de déplacement de la carte vers le haut

■ Pourcentage du total



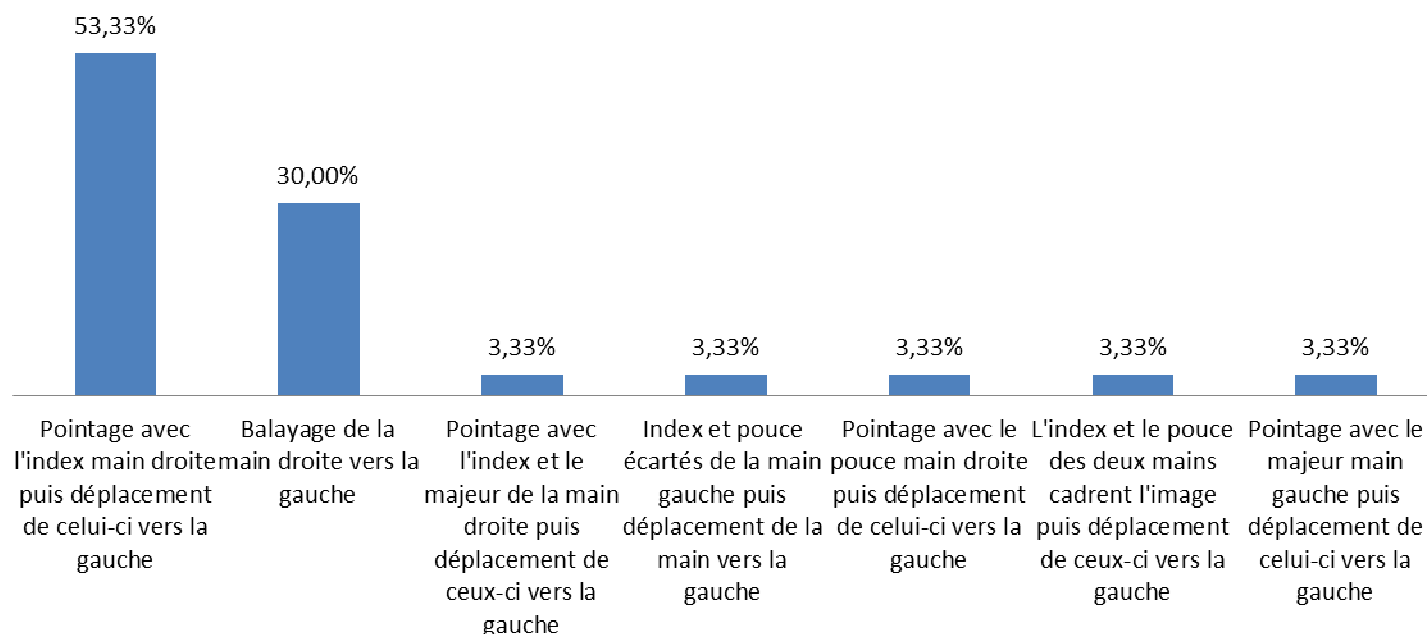
Geste de déplacement de la carte vers le bas

■ Pourcentage du total



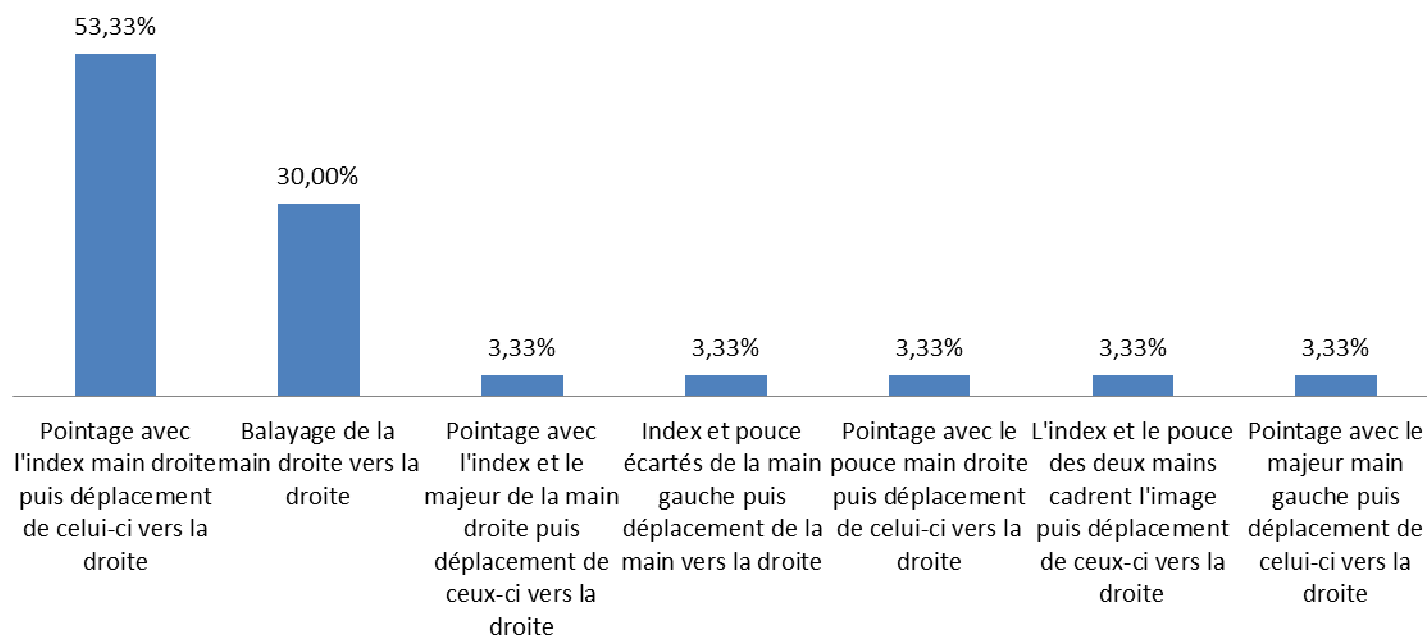
Geste de déplacement de la carte vers la gauche

■ Pourcentage du total



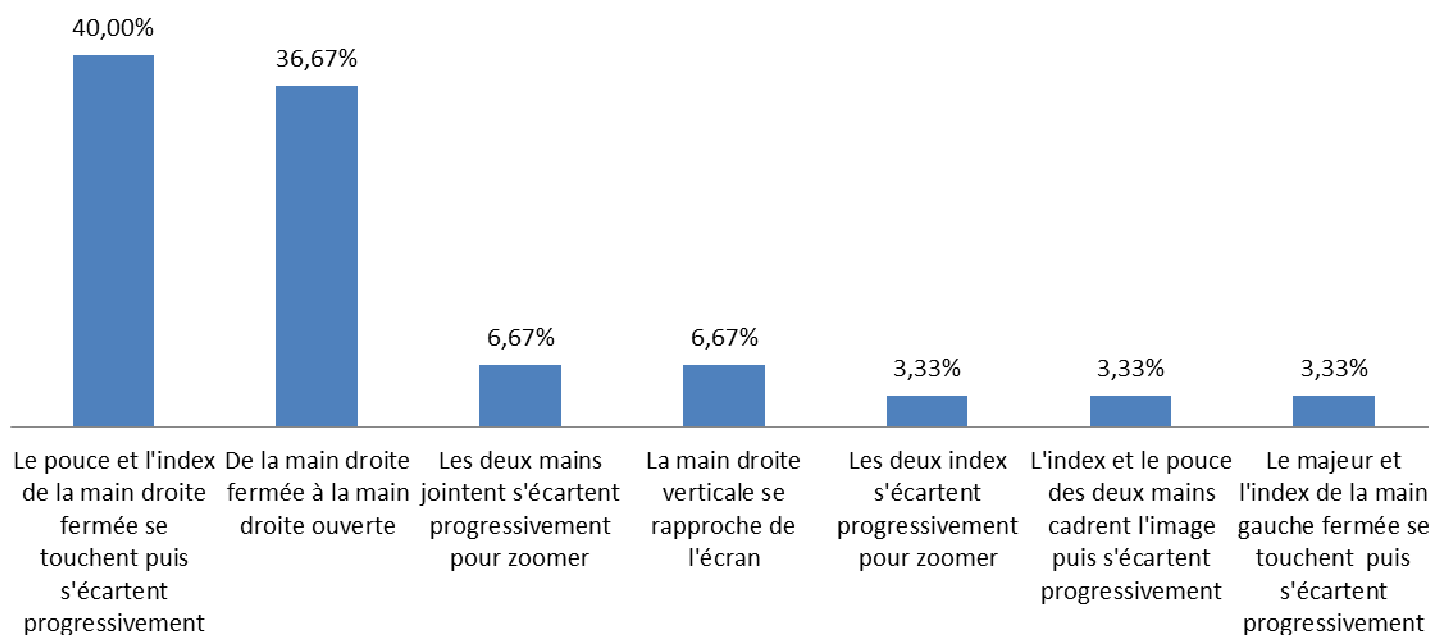
Geste de déplacement de la carte vers la droite

■ Pourcentage du total



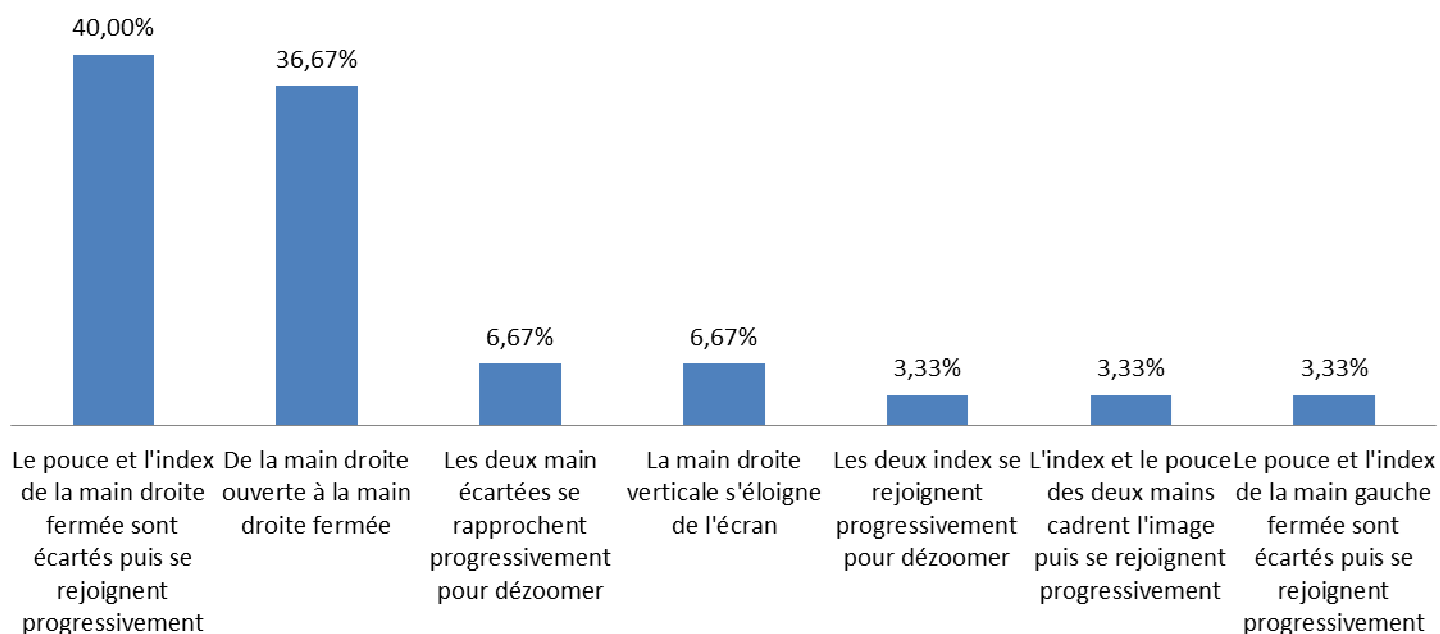
Geste du zoom de la carte

■ Pourcentage du total



Geste du dézoom de la carte

■ Pourcentage du total



6.3 Conclusion de l'enquête avec utilisateurs

1. Pour le déplacement de la carte dans une direction, l'enquête révèle que les utilisateurs ont choisi plusieurs possibilités de gestes :

- Il y a huit gestes différents de la main ou des mains pour les déplacements de la carte vers le haut ou vers le bas et sept gestes pour les déplacements de la carte vers la droite ou vers la gauche ;
- Les gestes demeurent quasi les mêmes pour le déplacement dans une direction. La seule différence est que les utilisateurs ont choisi un geste supplémentaire pour le déplacement vers le bas ou vers le haut en déplaçant leur main droite verticalement ou horizontalement.

Certaines possibilités de gestes sont plus employées que d'autres pour déplacer la carte dans une direction donnée :

- Le pointage avec l'index de la main droite et le déplacement dans la direction voulue représente 53,33 % des utilisateurs potentiels ;
- Le balayage de la main droite vers la droite ou vers la gauche pour le déplacement de la carte à droite ou à gauche représente 30,00% des utilisateurs potentiels ;
- Pour le déplacement de la carte vers le haut, 16,67 % des utilisateurs potentiels déplacent leur main droite verticalement vers le haut et 13,33 % déplacent leur main droite horizontalement vers le haut ;
- De même, pour le déplacement de la carte vers le bas, 16,67 % des utilisateurs potentiels déplacent leur main droite verticalement vers le bas et 13,33 % déplacent leur main droite horizontalement vers le bas.

Les autres gestes utilisés pour le déplacement de la carte dans une direction ne concernent qu'un utilisateur du panel et sont, de ce fait, assez marginaux.

Le pointage de l'index de la main droite et le déplacement dans la direction voulue résulte d'une influence technologique : celle de l'utilisation de la tablette ou du smartphone.

Pour le déplacement de la carte dans une direction, le sexe de l'utilisateur n'influence pas l'ordre en pourcentage. Seules les femmes avec une moyenne d'âge plus élevée réalisent des gestes particuliers et marginaux.

Les gestes les plus utilisés le sont par des utilisateurs qui ont un âge proche de la moyenne des âges de l'échantillon. Seuls les utilisateurs plus âgés (les femmes) réalisent des gestes particuliers et marginaux.

2. Pour le zoom et le dé zoom de la carte, il y a sept gestes différents de la main ou des mains et deux gestes plus employés que les autres :

- Le pouce et l'index de la main droite et l'index de la main droite fermée qui s'écartent progressivement dans le cas du zoom ou se rejoignent progressivement dans le cas du dé zoom (*pinch*) constitue les gestes la plus représentatifs avec 40% des utilisateurs potentiels. Ce pourcentage de 40% est indifférent selon les hommes ou les femmes. Les utilisateurs qui ont choisi ce geste sont les plus jeunes de l'échantillon. Ce geste résulte également de l'influence technologique de l'usage de la tablette ou du smartphone (*pinch* sur l'écran) ;
- La main droite fermée qui s'ouvre dans le cas du zoom et la main droite ouverte qui se referme dans le cas du dé zoom constitue le geste le plus employé avec 36,67 % des utilisateurs potentiels. Le sexe n'influence pratiquement pas ce pourcentage avec 33,33% d'hommes et 40,00% de femmes. L'âge moyen des utilisateurs de ce geste est plus élevé que la moyenne ;
- Les autres gestes utilisés pour le zoom et le dé zoom de la carte concernent deux et un utilisateur du panel et sont, de ce fait, assez marginaux.

L'échantillon d'utilisateurs proposé (quinze hommes et quinze femmes) est suffisamment représentatif de la population. Les conclusions de l'enquête portant sur l'échantillon peuvent être généralisées à toute la population.

6.4 Directives de conception

Il est important de présenter aux utilisateurs des options intuitives pour qu'ils ressentent la possession du contrôle de l'application. Il ne faut pas que l'interaction de l'utilisateur soit confuse, frustrante ou laborieuse. L'enquête utilisateur a permis d'identifier les gestes de contrôle les plus intuitifs.

Pour augmenter la clarté de l'interaction demandée à l'utilisateur, il est possible de prévoir un menu pourvu de boutons de choix, un contraste de ces boutons, un bouton de sortie de l'application et une icône représentative du geste d'interaction demandé. Des zones d'interaction peuvent aussi être renseignées sur l'écran. Il peut aussi être utile de montrer la meilleure position des mains qui permet une interaction optimum.

Le retour des utilisateurs est vraiment essentiel pour s'assurer que les orientations prises pour l'application soient claires et accessibles. Ce retour peut être un commentaire, un comportement.

Pour une application professionnelle d'une certaine complexité, il est recommandé d'inclure une vidéo qui montre ses différentes fonctionnalités et qui aide l'utilisateur, via des exemples, à comprendre l'interaction attendue. Une autre possibilité consiste à utiliser un didacticiel composé d'une série d'écrans successifs comportant les directions et tâches possibles de l'application pour montrer ses mécanismes. L'aide à l'écran ne doit pas être systématique mais est utilisée quand et où cela est vraiment nécessaire. Plus l'application est complexe plus l'apprentissage de l'utilisateur est important.

6.5 Implémentation de l'interface suite aux résultats de l'enquête

1. Zoom et dézoom de la carte

Suivant les conclusions de l'enquête, le pouce et l'index de la main droite fermée qui s'écartent l'un de l'autre (zoom) ou se rapprochent l'un de l'autre (dézoom) est le premier geste à implémenter pour l'interaction avec la carte. Le *pinch* utilisé pour le zoom et le dézoom représente 40 % des utilisateurs observés.

La main droite fermée qui s'ouvre dans le cas du zoom et la main droite ouverte qui se referme dans le cas du dé zoom constitue le second geste à implémenter. Il représente 36,67 %. Ces deux possibilités utilisent des données de la main.

Il y a trois possibilités de récupérer les données de la main :

- A partir d'un cadre (Frame)

Create a new frame

```
Frame = controller.frame();
```

Create a handlist

```
HandList hands = frame.hands();  
Hand firstHand = hands.get(0);
```

- A partir d'un identifiant connu ID

```
Hand knownHand = frame.hand (handID) ;
```

- A partir de la position relative de la main dans le cadre : la plus à gauche *leftmost*, la plus droite *rightmost*, la plus en avant du capteur *frontmost*.

Create a new frame

```
Frame frame = controller.frame() ;
```

Create a handlist

```
HandList hands = frame.hands();  
Hand leftmostHand = hands.leftmost();  
Hand rightmostHand = hands.rightmost();  
Hand frontmostHand = hands.frontmost();
```

« Frontmost » signifie la main la plus au-dessus du capteur (valeur de y la plus élevée).

« Rightmost » signifie la main la plus à droite du capteur (valeur de x la plus élevée).

« Leftmost » signifie la main la plus à gauche du capteur (valeur de s la plus faible).

La main la plus à droite n'est pas forcément une main droite de même pour la main la plus à gauche.

Dans le cas de la carte, l'utilisateur interagit seulement avec la main droite pour le zoom et le dé zoom. Il n'y a qu'une seule main dans le cadre. L'identifiant de la main n'est pas connu, il est incrémenté à chaque fois qu'une nouvelle main est présentée dans le champ de vision du capteur. La récupération des données de la main se fera à partir d'un cadre (Frame).

L'objet Hand peut être invalide, cela signifie qu'il ne contient pas des données correspondant à une main physiquement présente. Par exemple une main ayant un identifiant ID d'un cadre précédent mais qui n'existe pas dans le nouveau cadre ou un objet Hand créé par le constructeur. Il faut alors tester la validité de la main avec la fonction `isValid()` ;

Afin d'initialiser une main pour la rendre valide et comme il s'agit d'une main droite, la commande suivante est exécutée :

```
Hand rightmostHand = frame.hands().rightmost() ;
```

Cela permet de ne pas tester la validité de la main avec la fonction `isValid()` ;

Par généralisation, si les cas particuliers sont permis, la main gauche peut être utilisée pour les deux types de geste. Les données de la main sont alors obtenues à partir de la main la plus à droite.

Create a new frame

```
Frame frame = controller.frame() ;
```

Create a handlist

```
HandList hands = frame.hands();  
Hand rightmostHand = hands.rightmost();
```

Cela revient à initialiser l'objet Hand d'un Frame.

Cas du zoom et du dézoom par ouverture et fermeture de la main

Dans le cas du zoom par ouverture et fermeture de la main, la récupération des doigts n'est pas nécessaire. Le paramètre à obtenir est float `grabAngle` obtenu par la méthode `grabAngle()`.

L'angle obtenu est celui entre les doigts et la posture de saisie de la main.

Il est calculé en regardant l'angle entre la direction des quatre doigts et la direction de la main. Le pouce n'est pas pris en compte lors du calcul de l'angle. L'angle est de 0 radian pour une main ouverte, et atteint pi radians lorsque la main est fermée (poing).

Le zoom se réalise par le passage d'une main fermée (paramètre `grabAngle` de pi) à une main ouverte (paramètre `grabAngle` de 0). Le dézoom se réalise par le passage d'une main ouverte (paramètre `grabAngle` de 0) à une main fermée (paramètre `grabAngle` de pi).

Il est possible de récupérer le paramètre `grabAngle` à l'aide de la commande :

```
float grabAngle = hand.grabAngle() ;
```

Réalisation du zoom et du dé zoom grâce à la commande de souris :

zoom

robot.mouseWheel (-1) scroll up (vers écran) pour le zoom.

Temps d'attente grâce à un thread `try{ Thread.sleep(50) ;}catch (Exception e){}`

Passage du `grabAngle()` de pi à 0 : vérifier que le geste soit entamé en testant le `grabAngle()`.

dézoom

robot.mouseWheel(1) scroll down (vers utilisateur) pour le dé zoom.

Temps d'attente grâce à un thread `try{ Thread.sleep(50) ;}catch (Exception e){}`

Passage du `grabAngle()` de 0 à pi : vérifier que le geste soit entamé en testant le `grabAngle()`.

```
If (grabAngle ()<= Math.PI/4)
```

```
  dézoom
```

```
else
```

```
  zoom
```

Cas du zoom et du dézoom par le pinch

Il nécessaire de récupérer les données des doigts ainsi que la distance entre le pouce et index.

Dans le cas du zoom par le pinch, le paramètre à obtenir est float `pinchDistance` obtenu par la méthode `pinchDistance()`. Ce paramètre est donné en millimètres.

Dans le cas du zoom, `pinchDistance` passe de 0 mm à plus ou moins 100 mm.

0 mm dans le cas où le pouce et l'index sont l'un contre l'autre et 100 mm lorsqu'ils sont écartés au maximum.

Dans le cas du dézoom, `pinchDistance` passe de plus ou moins 100 mm à 0 mm.

Il est possible de récupérer le paramètre `pinchDistance` à l'aide de la commande :

```
float pinchDistance = hand.pinchDistance();
```

Réalisation du zoom et du dé zoom grâce à la commande de souris :

zoom

robot.mouseWheel (-1) scroll up (vers écran) pour le zoom.

Temps d'attente grâce à un thread `try{ Thread.sleep(50) ;}catch (Exception e){}`

Passage du `pinchDistance()` de 0 à 100 : vérifier que le geste soit entamé en testant le `pinchDistance()`.

dézoom

robot.mouseWheel(1) scroll down (vers utilisateur) pour le dé zoom.

Temps d'attente grâce à un thread `try{ Thread.sleep(50) ;}catch (Exception e){}`

Passage du pinchDistance() de 100 à 0 : vérifier que le geste soit entamé en testant le pinchDistance().

```
if (pinchDistance()<= 25)
  dézoom
else
  zoom
```

2. Déplacement de la carte dans une direction

Les deux gestes les plus utilisés d'après les résultats de l'enquête utilisateurs sont :

- Le pointage avec l'index de la main droite et le déplacement dans la direction voulue. Il représente 53.3 % des utilisateurs de l'enquête ;
- Le balayage de la main droite dans la direction voulue pour le déplacement de la carte.

Le balayage de la main droite vers la droite ou vers la gauche pour le déplacement de la carte à droite ou à gauche représente 30,0 % des utilisateurs de l'enquête.

Le balayage de la main droite vers le haut ou vers le bas sans tenir compte de l'orientation de la main pour le déplacement de la carte vers le haut ou vers le bas représente 33,34 % des utilisateurs de l'enquête pour le déplacement de la carte vers le haut et 26,66 % des utilisateurs de l'enquête pour le déplacement de la carte vers le bas.

Un premier programme est soumis aux utilisateurs afin de tester l'interaction sur la carte.

Trois gestes de l'interface de programmation ont été implémentés :

- Le cercle pour zoomer et dézoomer la carte. Le but est de voir si ce geste peut être employé pour cette fonctionnalité et s'il peut convenir pour une autre ;

```
if (g.type()==Type.TYPE_CIRCLE){
  CircleGesture circle = new CircleGesture (g);
  if(circle.pointable().direction().angleTo(circle.normal())<= Math.PI/4){
    robot.mouseWheel(1);
    try{Thread.sleep(50);}catch (Exception e){}
  }else{
    robot.mouseWheel(-1);
    try{Thread.sleep(50);}catch (Exception e){}
  }
}
```

- Le toucher d'écran *Screen Tap* pour vérifier que l'utilisateur a une bonne perception de la zone de toucher dans l'espace. Cette interaction correspond à un clic de souris sur un écran. L'utilisateur peut agir sur la carte en la sélectionnant pendant un certain laps de temps. Il est aidé par l'affichage sur la console du message « un screen Tap » lors de la reconnaissance du geste ;

```

if (g.type() == Type.TYPE_SCREEN_TAP){
    System.out.println("un screen Tap");
    robot.mousePress(InputEvent.BUTTON1_MASK);
    robot.delay(5000);
    robot.mouseRelease(InputEvent.BUTTON1_MASK);
}

```

Les paramètres du “screen Tap” peuvent être adaptés à l’aide de commandes de configuration : la vitesse minimum `Gesture.ScreenTap.MinForwardVelocity` en mm/sec, le renouvellement de l’historique en seconde `Gesture.ScreenTap.HistorySeconds` et la distance minimum en millimètres `Gesture.ScreenTap.MinDistance` .

```

public void onConnect (Controller c) {
    c.enableGesture(Gesture.Type.TYPE_SCREEN_TAP);

    c.config().setFloat("Gesture.ScreenTap.MinForwardVelocity",5.0f);
    c.config().setFloat("Gesture.ScreenTap.HistorySeconds", 0.5f);
    c.config().setFloat("Gesture.ScreenTap.MinDistance", 1.0f);
    c.config().save();
}

```

- Le balayage pour montrer l’importance de la vitesse d’exécution. Celui-ci ouvre et ferme le menu Windows par l’enfoncement et le relâchement de la touche Windows ;

```

if (g.type()==Type.TYPE_SWIPE && g.state () == State.STATE_START){
    robot.keyPress (KeyEvent.VK_WINDOWS);
    robot.keyRelease(KeyEvent.VK_WINDOWS);
}

```

Le programme suit la position de l’index comme pointeur de la souris à l’écran. La position est stabilisée et le déplacement s’effectue dans la boîte d’interaction qui est paramétrée.

```

Frame frame = c.frame();
InteractionBox box = frame.interactionBox();
for (Finger f:frame.fingers()){
    if (f.type ()==Finger.Type.TYPE_INDEX){
        Vector fingerPos = f.stabilizedTipPosition ();
        f.stabilizedTipPosition();
        Vector boxFingerPos = box.normalizePoint(fingerPos);
        Dimension screen = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
        robot.mouseMove((int)(screen.width * boxFingerPos.getX()),
            (int)(screen.height- boxFingerPos.getY()*screen.height));
    }
}

```

Un second programme permet de tester le déplacement de la carte dans le cas du balayage.

Une classe `MapView` sert à afficher l’image de la carte `travaux_mob.PNG` dans un cadre `JFrame`. La méthode `moveMap` déplace la carte vers une position X,Y.

```

public void moveMap(int X, int Y) {
    this.setLocation(X, Y);
}
public int getXLocation() {
    return (int) this.getLocation().getX();
}
public int getYLocation() {
    return (int) this.getLocation().getY();
}
}
}

```

La classe principale `MapApplication` initialise le `Controller` et le `Maplistener` et affiche le déplacement de la carte.

La classe `Maplistener` détecte un balayage de la main et sa position et sa position de départ est lue. Cette position est utilisée pour le déplacement de la carte dans une direction. Elle suit le mouvement du balayage et fait une translation dans sa direction. Si le balayage n'est pas détecté, le programme en attend un nouveau. Le balayage convient pour le déplacement dans une direction.

```

if (g.type() == Type.TYPE_SWIPE) {
    if(g.state() == State.STATE_START) {
        SwipeGesture swipeGesture = new SwipeGesture(g);
        Vector currentSwipePosition = swipeGesture.position();

        start_swipe_pos_X = currentSwipePosition.getX();
        start_swipe_pos_Y = currentSwipePosition.getY();
        start_map_pos_X = map.getXLocation();
        start_map_pos_Y = map.getYLocation();

    } else if (g.state() == State.STATE_UPDATE || g.state() == State.STATE_STOP) {
        SwipeGesture swipeGesture = new SwipeGesture(g);
        Vector currentSwipePosition = swipeGesture.position();

        Dimension screen = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
        map.moveMap(start_map_pos_X + (int) (screen.width * (currentSwipePosition.getX() -
start_swipe_pos_X) ),
start_map_pos_Y + (int) (screen.height * (currentSwipePosition.getY() - start_swipe_pos_Y)
));
    }
}
}
}

```

6.5.1 Résultats des tests

Résultats des tests de l'étude de cas

N°	Prénom	Age	Sexe	Profession	Choix de zoom/dézoom	Préférence	Commentaire	Choix déplacement de la carte	Commentaire	Interaction avec les gestes
1	Christian S.	46	M	Ouvrier	Le pouce et l'index de la main droite fermée puis s'écartent progressivement et inversement (pinch)	Préférence pour le pinch car plus réactif	pour les deux programmes pinch et main ouverte fermée, il y a un certain temps de réaction. Il faut recharger la carte au changement de programme	Pointage avec l'index de la main droite et déplacement dans la direction voulue	Difficulté de percevoir la zone d'interaction avec la sélection de la carte touche pour le déplacement avec l'index. Plusieurs essais ont été fait.	Intéressant car interaction avec les mains
2	David P.	43	M	Ouvrier	Main droite fermée à ouverte et inversement	Préférence pour le pinch car plus réactif	Le geste de la main droite ouverte à est moins réactif. Le geste est plus grand.	Déplacement horizontal de la main droite en haut et en bas. Balayage gauche/droite de la main droite.	Bonne stabilité Le pointage de touche ne marche pas.	Intérêt de la commande avec les gestes Commande de sélection
3	Laurent M.	51	M	Ouvrier	Main droite fermée à ouverte et inversement	Préférence pour le pinch car plus réactif.	Les deux programmes sont réactifs.	Pointage avec l'index main droite et déplacement dans la direction voulue.	Interaction avec le toucher de touche est difficile	Interaction avec les gestes est assez innovante
4	Eric G.	50	M	Ouvrier	Main droite qui monte ouverte et inversement	Pas de préférence	Utilisation d'un geste non programmé peu de réaction. Le programme main ouverte fermée n'est pas assez réactif Le programme pinch est trop réactif.	Balayage main droite dans la direction voulue	Manque de contrôle et de précision pour le toucher de touche	Interaction avec les gestes est intéressante
5	Cathy M.	50	F	Dessinatrice	Main droite fermée à ouverte et inversement	Préférence pour le pinch	Les deux programmes sont trop sensibles. Le programme main droite ouverte fermée est moins intuitif. Problème de conditions d'éclairage.	Pointage avec le pouce main droite et déplacement dans la direction voulue	Interaction avec le toucher de touche est difficile Le balayage est non fonctionnel	Interaction avec les gestes est intéressante pour un pointage en remplacement d'une souris, dans un environnement 3D et pour la revalidation de personne
6	Christophe F.	34	M	Ingénieur	L'index et le pouce cadrent la carte puis s'écartent ou se rapprochent	Préférence pour la main droite ouverte fermée et inversement	Il serait intéressant d'avoir un réglage de la position du zoom. Le pinch est aussi réactif mais moins pratique.	L'index et le pouce cadrent la carte puis se déplacent dans la direction voulue.	La sélection est difficile. Voir les transitions de gestes. Bien d'avoir une posture pour les marquer. Avoir un choix parmi un catalogue de gestes. Ne pas avoir de gestes contradictoires de transitions.	L'étude de cas est bien choisie. Interaction par les gestes bien pour les présentations sur écran. (si du matériel n'est pas autorisé)
7	Béatrice S.	47	F	Secrétaire	Main droite fermée à ouverte et inversement	Préférence pour la main droite ouverte fermée et inversement	Les deux programmes sont assez réactifs	Déplacement de la main droite verticale vers le haut et le bas. Balayage de la main droite vers la gauche et vers la droite.	Interaction avec le toucher de touche est difficile mais c'est une bonne idée Le balayage est fonctionnel	Interaction avec les gestes est intéressante. L'étude de cas est bien choisie.
8	Cédric T.	9	M	Elève	Les deux mains jointes s'écartent pour zoomer et se rapprochent pour dézoomer	Préférence pour la main droite ouverte fermée et inversement	Le zoom et le dézoom se réalise à chaque fois. Le pinch est plus compliqué.	Déplacement de la main droite verticale vers le haut et le bas. Balayage de la main droite vers la gauche et vers la droite.	Le pointage est bon et est assez stable. Interaction avec le toucher de touche est difficile. La zone d'interaction doit être renseignée. Cercle moins bon pour le zoom.	Exemples d'applications : allumer l'éclairage, télévision, jeux, contrôle des appareils ménagers.
9	Guillaume T.	7	M	Elève	Le pouce et l'index de la main droite fermée puis s'écartent progressivement et inversement (pinch)	Préférence pour le pinch	Les deux programmes sont assez réactifs	Déplacement de la main droite horizontale vers le haut et le bas. Balayage de la main droite vers la gauche et vers la droite.	le balayage réagit bien. Difficulté avec le toucher de touche.	Exemples d'applications : allumer l'ordinateur, jeux, contrôle des appareils ménagers.
10	Anne-C W.	42	F	Architecte	Main droite fermée à ouverte et inversement	Préférence pour la main droite ouverte fermée et inversement	Le programme est réactif. Le zoom se réalise à chaque fois. L'habitude d'utilisation de tablette encourage à placer la main proche du <i>Leap Motion</i> Position de la main difficile à trouver Le pinch est moins intuitif. La position des doigts n'est facile à trouver	Pointage de l'index main droite et déplacement dans la direction voulue	Balayage réagit bien. Le cercle pour le zoom n'est pas intuitif et demande un temps de reconnaissance. Le toucher de touche est plus compliqué à réaliser. Il nécessite un apprentissage et une interface visuelle.	La carte est un bon cas d'étude. Exemples d'applications : une impression d'écran, mise en route d'appareils ménagers.

Résultats des tests de l'étude de cas

11	Fabien T.	43	M	Employé de banque	Main droite fermée à ouverte et inversement	Préférence pour la main droite ouverte fermée et inversement	Le programme main ouverte et fermée réagit bien à chaque fois. Il y a moins d'erreur possible. Le pinch est plus compliqué.	Déplacement de la main droite verticale vers le haut et le bas. Balayage de la main droite vers la gauche et vers la droite.	Le cercle est assez aléatoire. Position index et balayage corrects. L'interaction avec une touche est difficile.	Exemples d'applications : Commande GPS, allumer l'éclairage et changer l'intensité. L'étude de cas est bien choisie.
12	Thérèse G.	82	F	Retraitée	Main droite fermée à ouverte et inversement	Préférence pour le pinch	Le geste choisi au départ n'est pas le geste implémenté. Une fois compris, les programmes sont réactifs. Le pinch est plus simple d'utilisation.	Déplacement de la main droite verticale vers le haut et le bas. Balayage de la main droite vers la gauche et vers la droite.	Le programme reconnaît un cercle alors qu'il n'y en a pas. Le balayage est reconnu. Bonne stabilité. L'interaction avec le toucher de touche est difficile et requiert une aide à l'écran.	Exemples d'applications : recherche dans un annuaire, situation sur une carte, sélection facile d'un docteur.
13	Olivier W.	51	M	Manager	Le pouce et l'index de la main droite fermée puis s'écartent progressivement et inversement (pinch)	Préférence pour la main droite ouverte fermée et inversement Plus intuitif.	Les deux programmes sont assez réactifs. Le zoom se fait toujours au même endroit. Il n'est pas possible de faire un zoom intermédiaire.	Pointage avec l'index de la main droite et déplacement dans la direction voulue	L'index comme pointeur de souris fonctionne bien. Le toucher d'écran est aléatoire et difficile. Une amélioration sonore ou visuelle est la bien venue. Le balayage se fait correctement. Le cercle pour zoomer est contre nature.	Interaction gestuelle non utile si un périphérique existe (souris). Idéal un mixte du <i>Leap Motion</i> et un périphérique. Applications possibles : commandes GPS, chauffage voiture. Environnement non sécurisé afin d'éviter les risques de vol.
14	Eric L.	51	M	Dessinateur	Main droite fermée à ouverte et inversement	Préférence pour la main droite ouverte fermée et inversement	Les deux programmes sont assez réactifs.	Pointage avec l'index de la main droite et déplacement dans la direction voulue	L'index comme pointeur de souris fonctionne bien. Le toucher d'écran est aléatoire et difficile. Le balayage se fait correctement. Le cercle pour zoomer est contre nature mais peut convenir pour une autre application.	Interaction avec les gestes est intéressante. Applications possibles : commandes GPS de voiture.
15	Josy A.	58	M	Dessinateur	Main droite fermée à ouverte et inversement	Préférence pour la main droite ouverte fermée et inversement	Meilleur réglage pour le zoom avec la main. Il est plus intuitif et convivial. Il est mieux adapté aux grands écrans. Le pinch est trop réactif. Il est possible d'agir sur la vitesse. Il manque la possibilité d'avoir un zoom intermédiaire.	Pointage avec l'index et le majeur de la main droite puis déplacement dans la direction voulue	Le zoom avec le cercle est bien pour cibler comme une loupe. L'interaction de toucher de touche est difficile. La représentation d'un index serait plus parlante que le pointeur souris. Le balayage réagit bien. Il faudrait régler la vitesse.	Interaction avec les gestes est bien pour l'aspect dynamique. Applications possibles : commandes GPS de voiture et lecture de livres par balayage.
16	Alain B.	42	M	Dessinateur	Le pouce et l'index de la main droite fermée puis s'écartent progressivement et inversement (pinch)	Préférence pour le pinch.	Le pinch est plus précis. Les zoom et dézoom se font plus facilement. Les transitions pour les déplacements sont plus faciles. Les deux programmes sont réactifs.	Balayage main droite dans la direction voulue	Le zoom avec le cercle pour une capture d'écran. L'interaction de toucher de touche n'est pas toujours reconnue. Le balayage peut être utile (dépend de la fonctionnalité).	Interaction avec les gestes est intéressante. Application : Capture d'écran pour le stockage d'images afin de les réutiliser par la suite par balayage.
17	Laurent W.	46	M	Ingénieur	Le pouce et l'index de la main droite fermée puis s'écartent progressivement et inversement (pinch)	Préférence pour le pinch.	Le pinch est plus précis. Le geste est déjà connu et ne nécessite pas d'apprentissage. Il est plus intuitif et naturel. Les deux programmes fonctionnent.	Pointage avec l'index de la main droite et déplacement dans la direction voulue	Le toucher d'écran nécessite beaucoup de précision de position pour l'utilisateur. Le balayage est assez intuitif. Le cercle est un geste facile à effectuer et est bien reconnu. Le cercle a une dimension et ne peut être utilisé pour des tâches nécessitant plusieurs dimensions. Il est bien pour le défilement de texte	Interaction avec les gestes est intéressante. Applications : Toutes les applications de réalité virtuelle et de réalité augmentée. Les applications qui emploient des bômes interactives . Les commandes de menu dans un restaurant peuvent se faire à l'aide de gestes de la main.
18	Joëlle H.	48	F	Employée	Main droite fermée à ouverte et inversement	Préférence pour la main droite ouverte fermée et inversement	Les deux programmes sont assez réactifs	Pointage avec l'index de la main droite et déplacement dans la direction voulue	Interaction avec le toucher de touche est difficile. Le balayage est fonctionnel. Le cercle ne convient pas pour le zoom.	Interaction avec les gestes est intéressante. L'étude de cas est bien choisie.

6.5.2 Conclusion des tests de l'étude de cas

En ce qui concerne le zoom et le dézoom de la carte, il n'y a pas une préférence nette pour le choix du geste. Un testeur n'a pas de préférence. Les autres choisissent le *pinch* (pouce et index de la main droite qui s'écartent progressivement pour le zoom et inversement pour le dézoom) dans 50 % des cas et la main droite fermée à ouverte (et inversement) dans 44 % des cas.

Dans la moitié des cas, les testeurs gardent le même geste identifié lors de l'enquête. Dans l'autre moitié des cas, les testeurs choisissent un autre geste soit parce que le geste donné pendant l'enquête n'était ni le *pinch* ni la main ouverte fermée (38 %) ; soit parce que le geste choisi était différent (62 %).

En cas de changement de geste, 80 % des personnes ont opté pour le *pinch* à la place de la main droite ouverte fermée.

La reconnaissance de geste pour le zoom et dézoom nécessite un temps de réaction dû au chargement des données. Tous les programmes doivent être fermés. Les conditions d'éclairage sont importantes. Les deux programmes réagissent bien. Dans les deux cas, il n'est pas possible d'obtenir une position de zoom intermédiaire. Du fait d'une image fixe, le zoom s'effectue toujours sur un même point.

L'habitude culturelle liée à l'usage des smartphones, tablettes, GPS, ... impose l'usage du *pinch*. Ce geste ne nécessite plus d'apprentissage. Le *pinch* est mieux adapté aux déplacements de la carte.

Le geste de la main droite ouverte fermée est mieux adapté aux grands écrans. Certaines personnes trouvent qu'il engendre moins d'erreur.

Le fait de former un cercle est un geste facile à utiliser et est assez bien reconnu. Il y a un temps de reconnaissance.

Ce n'est pas un geste intuitif pour réaliser un zoom. Il ne convient pas pour des tâches en deux dimensions comme un déplacement de la carte mais peut convenir pour un défilement de texte par exemple celui du commentaire sur le côté de la carte. Il peut être utilisé pour cibler ou pour une capture d'écran.

Dans la plupart des cas, le balayage est reconnu. Mais, le balayage pour le déplacement de la carte dans toutes les directions ne convient pas très bien car une fois identifié dans une direction, il doit poursuivre une translation dans cette même direction.

Le toucher sur l'écran est aléatoire et difficile. Il nécessite un sérieux apprentissage qui nécessite beaucoup de précision. La zone d'interaction devrait être renseignée soit visuellement soit auditivement. Malgré une adaptation des paramètres repris dans l'interface de programmation et plusieurs essais successifs, il n'est pas amélioré significativement. Cependant, une nouvelle option pour ce geste est de générer une autre implémentation avec des tolérances différentes.

L'index utilisé comme pointeur de souris fonctionne correctement. Sa position est bien stabilisée. La représentation peut être améliorée par une icône symbolique.

Les transitions entre les gestes sont importantes et ne sont pas être contradictoires. Des postures généralement mieux reconnues peuvent être utilisées pour les réaliser. Une amélioration est d'établir un catalogue de gestes sur l'écran pour la réalisation de telle ou telle tâche.

Tous les participants s'accordent à dire que l'interaction par les gestes est nécessaire et intéressante et que l'étude de cas est bien utile.

L'interaction par le geste permet de nombreuses applications telles que :

- revalidation gestuelle de personne,
- présentation et commande sur écran (ex. : menu de restaurant)
- contrôle d'éclairage, appareils ménagers, télévision,
- impression d'écran,
- commandes GPS et chauffage d'une voiture,
- lecture de livre,
- pointage en environnement 3D, réalité virtuelle et augmentée.

7. Conclusions et perspectives

7.1 Contenu

La première étape du travail aboutit aux classifications de gestes. Elles vont de la plus générale à la plus spécifique. Les trois catégories des périphériques de reconnaissance des gestes sont identifiées : les périphériques basés sur les capteurs, ceux reposant sur la vision et ceux reposant sur le toucher. Les étapes du processus de reconnaissance de gestes par la vision comme dans le cas du contrôleur *Leap Motion*, sont décrites. La reconnaissance des gestes par le toucher est brièvement commentée.

La seconde étape du travail consiste à décrire les caractéristiques du contrôleur *Leap Motion* et ses possibilités de développement.

La troisième étape du travail évalue les performances du contrôleur *Leap Motion*. Les critères d'évaluation adoptés sont la précision, la fiabilité, la robustesse, les capacités de pointage et d'interaction en 3D pour le suivi de gestes en statique et en dynamique. Ensuite, le contrôleur *Leap Motion* est comparé à la souris pour les tâches de pointage et au *Microsoft Kinect* pour la reconnaissance des gestes de la main. Cette étape se termine par des recommandations personnelles afin d'améliorer les capacités de reconnaissance du capteur *Leap Motion*.

La quatrième étape du travail fournit un état de l'art des applications du contrôleur *Leap Motion* à partir d'une quarantaine de références scientifiques sélectionnées. Les applications sont catégorisées dans un tableau selon les thèmes reconnus.

La cinquième étape du travail décrit l'interface de programmation API java du contrôleur *Leap Motion* et évalue les capacités de reconnaissance de gestes pour les applications en local et les applications web.

La sixième étape du travail consiste en une étude de cas : l'interaction avec une carte installée sur une borne dans le hall de la gare de Namur. Ce cas est étudié via une enquête auprès d'utilisateurs variés et sur base de la reconnaissance de gestes inclus dans l'interface de programmation. L'interaction sur la carte consiste à la déplacer dans les quatre directions, à la zoomer et à la dézoomer par des gestes. La création d'un prototype permet d'implémenter les gestes les plus utilisés d'après l'enquête utilisateur.

7.2 Synthèse

Les points à retenir sont les suivants :

- Le *Leap Motion* basé sur l'émission par trois LED de lumière infrarouge captée par deux caméras infrarouges reconnaît en trois dimensions les gestes de la main.
- Le *Leap Motion* est compact et peu coûteux en comparaison avec d'autres capteurs de reconnaissance de gestes.
- Le *Leap Motion* est précis et fiable pour un suivi de points statiques. La précision est moindre pour le suivi de points dynamiques d'autant plus que l'utilisateur s'éloigne de l'appareil.

La précision en toucher de cible est moins bonne que sans toucher.

Comparé à une souris, le *Leap Motion* est meilleur pour une tâche de pointage tridimensionnelle et moins bon pour une tâche de pointage bidimensionnelle. Une limitation importante de performance est l'inconsistance de sa fréquence d'échantillonnage.

- Des applications du *Leap Motion* apparaissent comme plus prometteuses.

Celles-ci sont :

- les applications médicales dont l'imagerie médicale où l'emploi du *Leap Motion* pour la sélection d'images assure un environnement stérile et la réhabilitation de personne par le jeu ;
 - le langage des signes : un catalogue de signes des doigts correspondant à une lettre est établi. Le *Leap Motion* est utilisé pour reconnaître ces signes et la lettre correspondante. L'ensemble des lettres reconnues forme un mot ;
 - les applications de réalité virtuelle : les jeux, les manipulations d'objets virtuels, un outil de visualisation des données ;
 - les applications musicales : instruments musicaux augmentés complétés par des capteurs et les quasi-instruments musicaux virtuels commandés par les gestes.
- Pour toute application professionnelle, il est nécessaire de développer un algorithme de reconnaissance spécifique.
 - L'interface de programmation (API) java du *Leap Motion* est imprécise.
 - Il ne reconnaît pas systématiquement les gestes de base car il les confond avec d'autres gestes ; il n'appréhende pas le démarrage du geste ou la vitesse du geste n'est pas optimum.
 - De mauvaises conditions de luminosité réduisent la précision et le suivi.
 - Il peut y avoir des problèmes d'occlusion, de faux positif (reconnaissance alors qu'il n'y en a pas) ou de faux négatif (non reconnaissance alors qu'il y en a).
 - Le programme est assez gourmand en capacité de traitement.

- Les bonnes pratiques à utiliser pour la création d'une application de reconnaissance de gestes de la main avec le *Leap Motion* sont les suivantes :
 - Il est important de réaliser une enquête auprès d'utilisateurs. L'échantillon de l'enquête doit être représentatif de la population concernée par l'application. L'enquête est statistiquement représentative et reproductible. L'enquête établit un catalogue de gestes résultant d'interactions sur un prototype ou contrôlées par un humain (*magicien d'Oz*) ;
 - Les gestes les plus représentatifs de l'enquête auprès d'utilisateurs déterminent les gestes les plus intuitifs. Ces gestes sont implémentés dans l'application afin que l'interface utilisateur soit la plus intuitive possible ;
 - L'enquête a révélé que l'aspect culturel détermine le choix de gestes : le *pinch* (écart et rapprochement du pouce et de l'index) employé sur les tablettes et les smartphones est le geste le plus employé pour le zoom et le dé zoom de la carte ;
 - Il est toujours préférable de réaliser un calibrage du *Leap Motion* ;
 - Il faut veiller à obtenir de bonnes conditions d'éclairage ;
 - Il est préférable d'utiliser une seule main afin d'empêcher certains problèmes d'occlusion ;
 - La position de la main a son importance : inclinaison, distance du *Leap Motion*. Elle doit se trouver dans le volume d'interaction ;
 - Certains gestes comme le toucher d'écran sont difficiles à réaliser car ils nécessitent de connaître la zone d'interaction du toucher. Il est possible d'aider l'utilisateur à la percevoir grâce à une interface visuelle ;

7.3 Contributions personnelles

Le choix des classifications de gestes est volontairement établi en vue d'introduire les gestes reconnus par le contrôleur *Leap motion* : les gestes de la main. L'énumération des principaux capteurs de chaque catégorie suivant leur date d'apparition ainsi que leurs principales caractéristiques résulte d'une recherche personnelle. Une ligne du temps permet de les positionner.

Plusieurs références scientifiques sont sélectionnées sur *Google Scholar* afin d'établir une version d'évaluation résumée des performances. Cette évaluation reprend les qualités du *Leap Motion* en termes de précision, fiabilité, robustesse et facultés de pointage. Quelques pistes d'amélioration des capacités de reconnaissance documentées à partir d'autres références sont proposées.

Les applications possibles sont choisies sur *Google Scholar* sur base de leur fréquence, pertinence et mots clés. Pour chaque référence, l'application est décrite de même que le logiciel de reconnaissance. Les applications sont catégorisées dans un tableau suivant le thème, les gestes employés, le logiciel de reconnaissance et l'emploi d'autres capteurs. Les éléments du tableau sont commentés suivant les gestes employés, l'utilisation d'autres capteurs et leur emploi dans tel ou tel domaine. Des améliorations potentielles sont proposées.

Une évaluation des interfaces de programmation applicative (API) du capteur *Leap Motion* est présentée consistant en une version résumée et individuelle d'éléments factuels issus d'une quinzaine de références afin d'offrir une vision globale et synthétique de leurs facultés de reconnaissance.

Une série de programmes en java sont réalisés afin de procéder à des tests des capacités de reconnaissance de l'interface de programmation (API) du *Leap Motion*. Les résultats des tests sont repris dans un tableau.

Une étude de cas vise à mettre en pratique les connaissances acquises sur le *Leap Motion* et sur son interface de programmation. Le choix de l'étude est validé par une enquête auprès d'utilisateurs réalisée par la technique du *magicien d'Oz* (i.e. contrôlée par un humain). Les gestes de l'enquête sont identifiés dans un tableau. Leur pourcentage par rapport au total des participants, au total des hommes et des femmes est défini dans un tableau. Les gestes sont ensuite représentés graphiquement en fonction de ce pourcentage. Les conclusions de l'enquête s'en déduisent.

Des programmes en java qui implémentent les gestes les plus représentatifs sont conçus et écrits afin de les tester auprès des utilisateurs de l'enquête. Les résultats des tests présentent les conclusions personnelles suite à leur analyse.

7.4 Perspectives futures

Les possibilités de recherche et développement futurs se situent au niveau des algorithmes de reconnaissance de gestes. En effet, à chaque fois que le *Leap Motion* est utilisé pour une application professionnelle, il lui est associé un algorithme de reconnaissance particulier.

Ces algorithmes de reconnaissance de gestes sont classés en trois grandes familles : les algorithmes basés sur un modèle de référence (*Rubine*, *Dynamic Time Warping*), les algorithmes basés sur l'apprentissage machine (*Hidden Markov Models*, *Neural networks*, *Support Vector Machine*) et les algorithmes basés sur les règles (*Ladder*).

- *Rubine* : chaque geste est réduit à vecteur de caractéristiques et correspond à un point multidimensionnel. Il s'agit ensuite de classer ces points parmi les catégories de gestes.
- *Dynamic Time Warping* : cette méthode recherche un appariement optimal entre deux séries temporelles, sous certaines restrictions.
- *Hidden Markov Models* : est un modèle statistique dans lequel le système modélisé est supposé être un processus markovien de paramètres inconnus. Dans un tel processus, la prédiction du futur à partir du présent n'est pas rendue plus précise par des éléments d'information concernant le passé.
- *Neural networks* : ce sont des systèmes informatiques inspirés par les réseaux de neurones biologiques qui améliorent progressivement leurs performances en considérant des exemples.
- *Support Vector Machine* : il s'agit d'un ensemble de techniques d'apprentissage supervisé, destinées à résoudre les problèmes d'appartenance à des groupes et des relations statistiques entre variables.

Le défi futur est de choisir l'algorithme le plus fiable et le plus précis en fonction de l'application sur base de tests et d'études préalables. Une étape ultérieure est que le logiciel choisisse lui-même dynamiquement l'algorithme en fonction de paramètres prédéterminés.

Bibliographie

-
- ¹ Baudel T. and Beaudouin –Lafon M., “Charade: Remote control of objects using free-hand gestures”, Communications of the ACM, pages 28-35, July 1993.
- ² <https://fr.wikipedia.org/wiki/Geste> (La dernière modification de cette page a été faite le 8 octobre 2014 à 10:38)
- ³ Kendon A., *Gesture: Visible Action as Utterance*, Cambridge University Press, 2004.
- ⁴ Dumas B., “Advanced Human Machine Interaction: Gestures and Tabletop interaction”, University of Namur, page 9, 2018.
- ⁵ Cadoz C., *Les réalités virtuelles*, Flammarion, 1994.
- ⁶ Rime B. and Schiaratura L., “Gestures and speech, in *Fundamentals of non verbal behavior*”, Cambridge University Press, 1991.
- ⁷ Pavlovic V.I., Sharma R. and Huang T.S., “Visual interpretation of hand gestures for human-computer interaction: A review: *Pattern Analysis and Machine Intelligence*”, IEEE Transactions on 19, n°7, 1997.
- ⁸ Dumas B., “Advanced Human Machine Interaction: Gestures and Tabletop interaction”, University of Namur, page 10, 2018.
- ⁹ Kessler G D., Hodges L F. and Walker N., *Evaluation of the cyberglove as a whole hand input device*, ACM Trans. on Computer-Human Interactions, 1995.
- ¹⁰ Colaço A., Kirmani A., Gong N-W., Soo Yang H., Schmandt C., “Mime: Compact, Low-power, 3D Gesture Sensing for interaction with Head-Mounted Displays”, UIST'13, St. Andrews, UK , October 8-11, 2013.
- ¹¹ Dumas B., “Advanced Human Machine Interaction: Gestures and Tabletop interaction”, University of Namur, pages 26-33, 2018.
- ¹² <http://www-igm.univ-mlv.fr/~dr/XPOSE2013/LeapMotion> (dernière mise à jour 10 mars 2014).
developer.leapmotion.com. js.leapmotion.com. Sutherland M. “Introducing the Leap Motion Controller”, 2013.
- ¹³ Tung J.Y., Lulic T., Gonzalez D.A., Tran J., Dickerson C.R., Roy E.A., “Evaluation of a portable markerless finger position capture device: accuracy of the Leap Motion controller in healthy adults”, Department of Mechanical and Mechatronics Engineering, Department of Kinesiology, University of Waterloo, 2015.
- ¹⁴ Guna J., Jakus G., Pogačnik M., Tomažič S., Sodnik J., “An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking”, Sensors, 2014.
- ¹⁵ Weichert F., Bachmann D., Rudak B., Fisseler D., “Analysis of the accuracy and robustness of the leap motion controller”, Sensors, mdpi.com, 2013.
- ¹⁶ Bachmann D., Weichert F., Rinkenauer G., “Evaluation of the leap motion controller as a new contact-free pointing device”, Sensors, mdpi.com, 2014.
- ¹⁷ Coelho J.C., Verbeek F.J., “Pointing task evaluation of leap motion controller in 3d virtual environment”, Creating the Difference, 2014.
- ¹⁸ Adhikarla V.K., Sodnik J., Szolgay P., Jakus G., “Exploring direct 3d interaction for full horizontal parallax light field displays using leap motion controller”, Sensors, mdpi.com, 2015.
- ¹⁹ Brown M.A., Stuerzlinger W., “The performance of un-instrumented in-air pointing”, Proceedings of Graphics Interface 2014, dl.acm.org, 2014.
- ²⁰ Marin G., Dominio F., Zanuttigh P., “Hand gesture recognition with leap motion and kinect devices”, Image Processing (ICIP), ieeexplore.ieee.org, 2014.
- ²¹ Hernoux F., Bearee R., Gajny L., Nyiri E., “Leap Motion pour la capture de mouvement 3D par spline L1”, hal.inria.fr, 2013.
- ²² Avola D., Cinque L., Levialdi S., Petracca A., “Markerless Hand Gesture Interface Based on LEAP Motion Controller”, pdfs.semanticscholar.org, 2014.
- ²³ Zhang X., Liu H., Yuan J., Huang X., “A Non-parametric RDP Algorithm Based on Leap Motion”, dpi-proceedings.com, 2016.
- ²⁴ Branstett J., Gagneux V., Leleu A., Levadoux B., “ConductHome: Gesture Interface Control of Home Automation Boxes”, waset.org, 2015.

-
- ²⁵ Tung J.Y., Lulic T., Gonzalez D.A., Tran J., Dickerson C.R., Roy E.A., "Evaluation of a portable markerless finger position capture device: accuracy of the Leap Motion controller in healthy adults", Department of Mechanical and Mechatronics Engineering, Department of Kinesiology, University of Waterloo, 2015.
- ²⁶ Guna J., Jakus G., Pogačnik M., Tomažič S., Sodnik J., "An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking", *Sensors*, 2014.
- ²⁷ Weichert F., Bachmann D., Rudak B., Fisseler D., "Analysis of the accuracy and robustness of the leap motion controller", *Sensors*, mdpi.com, 2013.
- ²⁸ Weichert F., Bachmann D., Rudak B., Fisseler D., "Analysis of the accuracy and robustness of the leap motion controller", *Sensors*, mdpi.com, 2013.
- ²⁹ Guna J., Jakus G., Pogačnik M., Tomažič S., Sodnik J., "An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking", *Sensors*, 2014.
- ³⁰ Guna J., Jakus G., Pogačnik M., Tomažič S., Sodnik J., "An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking", *Sensors*, 2014.
- ³¹ Tung J.Y., Lulic T., Gonzalez D.A., Tran J., Dickerson C.R., Roy E.A., "Evaluation of a portable markerless finger position capture device: accuracy of the Leap Motion controller in healthy adults", Department of Mechanical and Mechatronics Engineering, Department of Kinesiology, University of Waterloo, 2015.
- ³² Guna J., Jakus G., Pogačnik M., Tomažič S., Sodnik J., "An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking", *Sensors*, 2014.
- ³³ Guna J., Jakus G., Pogačnik M., Tomažič S., Sodnik J., "An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking", *Sensors*, 2014.
- ³⁴ Guna J., Jakus G., Pogačnik M., Tomažič S., Sodnik J., "An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking", *Sensors*, 2014.
- ³⁵ Bachmann D., Weichert F., Rinkenauer G., "Evaluation of the leap motion controller as a new contact-free pointing device", *Sensors*, mdpi.com, 2014.
- ³⁶ Adhikarla V.K., Sodnik J., Szolgay P., Jakus G., "Exploring direct 3d interaction for full horizontal parallax light field displays using leap motion controller", *Sensors*, mdpi.com, 2015.
- ³⁷ Marin G., Dominio F., Zanuttigh P., "Hand gesture recognition with leap motion and kinect devices", *Image Processing (ICIP)*, ieeexplore.ieee.org, 2014.
- ³⁸ Hernoux F., Bearee R., Gajny L., Nyiri E., "Leap Motion pour la capture de mouvement 3D par spline L1", hal.inria.fr, 2013.
- ³⁹ Avola D., Cinque L., Levaldi S., Petracca A., "Markerless Hand Gesture Interface Based on LEAP Motion Controller", pdfs.semanticscholar.org, 2014.
- ⁴⁰ Zhang X., Liu H., Yuan J., Huang X., "A Non-parametric RDP Algorithm Based on Leap Motion", dpi-proceedings.com, 2016.
- ⁴¹ Branstett J., Gagneux V., Leleu A., Levadoux B., "ConductHome: Gesture Interface Control of Home Automation Boxes", waset.org, 2015.
- ⁴² Lyu R., Ze Y., Chen W., "A flexible finger-mounted airbrush model for immersive freehand painting", *IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS) Wuhan, China*, 2017.
- ⁴³ Sutton J., "Air painting with corel painter freestyle and the leap motion controller: A revolutionary new way to paint!" *ACM SIGGRAPH 2013 Studio Talks*, dl.acm.org, 2013.
- ⁴⁴ Branstett J., Gagneux V., Leleu A., Levadoux B., "ConductHome: Gesture Interface Control of Home Automation Boxes", waset.org, 2015.
- ⁴⁵ Georgoulas C., Raza A., Güttler J., "Home environment interaction via service robots and the leap motion controller", search.proquest.com, 2014.
- ⁴⁶ Yu N., Xu C., Wang K., Yang Z., "Gesture-based telemanipulation of a humanoid robot for home service tasks", ieeexplore.ieee.org, 2015.
- ⁴⁷ Boyali A., Hashimoto N., Matsumoto O., "Hand posture control of a robotic wheelchair using a leap motion sensor and block sparse representation based classification", academia.edu, SMART 2014, 2014.
- ⁴⁸ Khademi M., Mousavi Hondori H., "Free-hand interaction with leap motion controller for stroke rehabilitation", dl.acm.org, 2014.

-
- ⁴⁹ Karashanov A., Manolova A., Neshov N., “Application for hand rehabilitation using leap motion sensor based on a gamification approach”, *International Journal of Advance Research in Science and Engineering*, researchgate.net, 2016.
- ⁵⁰ Mousavi Hondori H., Khademi M., “A review on technical and clinical impact of Microsoft Kinect on physical therapy and rehabilitation”, *Journal of Medical Engineering*, hindawi.com, 2014.
- ⁵¹ Blažica B., Novak F., Biasizzo A., Bohak C., “3D serious games for Parkinson's disease management”, Faculty for Computer and Information Science, University of Ljubljana, 2016.
- ⁵² Manolova A., “System for touchless interaction with medical images in surgery using Leap Motion”, e-university.tu-sofia.bg, 2014.
- ⁵³ Bizzotto N., Costanzo A., Bizzotto L., Regis D., Sandri A., “Leap motion gesture control with osirix in the operating room to control imaging first experiences during live surgery”, *Surgical innovation*, 2014.
- ⁵⁴ Rosa GM., Elizondo ML., “Use of a gesture user interface as a touchless image navigation system in dental surgery: Case series report”, *Imaging science in dentistry*, 2014.
- ⁵⁵ Gauthier Y., Marcy J., Duprat D., “Interactions gestuelles pour StripTIC, un environnement tangible pour les contrôleurs aériens”, hal-enac.archives-ouvertes.fr, 2014.
- ⁵⁶ Chatterjee I., Xiao R., Harrison C., “Gaze + gesture: Expressive, precise and targeted free-space interactions”, *Conference on Multimodal Interaction*, dl.acm.org, 2015.
- ⁵⁷ Sridhar S., Feit AM., Theobalt C., “Investigating the dexterity of multi-finger input for mid-air text entry”, dl.acm.org, 2015.
- ⁵⁸ Khan FR., Ong HF., Bahar N., “A sign language to text converter using Leap motion”, media.neliti.com, 2016.
- ⁵⁹ Chuan CH., Regina E., “American sign language recognition using leap motion sensor”, ieeexplore.ieee.org, 2014.
- ⁶⁰ Elons AS., Ahmed M., Shedid H., “Arabic sign language recognition using leap motion sensor”, ieeexplore.ieee.org, 2014.
- ⁶¹ Potter LE., Araullo J., Carter L., “The leap motion controller: a view on sign language”, dl.acm.org, 2013.
- ⁶² Fanini B., “A 3D interface to explore and manipulate multi-scale virtual scenes using the leap motion controller”, researchgate.net, 2014.
- ⁶³ Regenbrecht H., Collins J., Hoermann S., “A leap-supported, hybrid AR interface approach”, dl.acm.org, 2013.
- ⁶⁴ Donalek C., Djorgovski SG., “Immersive and collaborative data visualization using virtual reality platforms”, 2014 IEEE International Conference on Big Data, ieeexplore.ieee.org, 2014.
- ⁶⁵ Ritter M., Aska A., “Leap motion as expressive gestural interface”, ICMC, smc.afim-asso.org, 2014.
- ⁶⁶ Han J., Gold NE., “Lessons Learned in Exploring the Leap Motion™ Sensor for Gesture-based Instrument Design”, discovery.ucl.ac.uk, 2014.
- ⁶⁷ Hantrakul L., Kaczmarek K., “Implementations of the Leap Motion in sound synthesis, effects modulation and assistive performance tools”. ICMC, pdfs.semanticscholar.org, 2014.
- ⁶⁸ Tormoen D., Thalmann F., Mazzola G., “The Composing Hand: Musical Creation with Leap Motion and the BigBang Rubette.”, NIME, nime.org, 2014.
- ⁶⁹ De Pra Y., Fontana F., Tao L., “Infrared vs. Ultrasonic Finger Detection on a Virtual Piano Keyboard.”, ICMC, air.uniud.it, 2014.
- ⁷⁰ Silva ES, De Abreu JAO., De Almeida JHP., “A preliminary evaluation of the leap motion sensor as controller of new digital musical instruments”, pdfs.semanticscholar.org, 2013.
- ⁷¹ Françoise J., Schnell N., Borghesi R., “Probabilistic models for designing motion and sound relationships”, hal.archives-ouvertes.fr, 2014.
- ⁷² Berndt A., Waloschek S., Hadjakos A., “Hand Gestures in Music Production”, cemfi.de, 2016.
- ⁷³ Liu YK., Zhang YM., “Toward welding robot with human knowledge: A remotely-controlled approach”, *IEEE Transactions on Automation Science*, ieeexplore.ieee.org, 2015.
- ⁷⁴ Zubrycki I., Granosik G., “Using Integrated Vision Systems: Three Gears and Leap Motion, to Control a 3-finger Dexterous Gripper”, researchgate.net, 2014.
- ⁷⁵ Jin H., Chen Q., Chen Z., Hu Y., Zhang J., “Multi-LeapMotion sensor based demonstration for robotic refine tabletop object manipulation task”, *CAAI Transactions on Intelligence*, Elsevier, 2016.

-
- ⁷⁶ Du G., Zhang P. “A markerless human–robot interface using particle filter and Kalman filter for dual robots”, IEEE Transactions on Industrial Electronics, ieeexplore.ieee.org, 2015.
- ⁷⁷ Figueiredo M., Sousa L., Cardoso P., “Learning technical drawing with augmented reality and holograms”, pdfs.semanticscholar.org, 2014.
- ⁷⁸ Aşut S., “Enhanced Human-Computer Interaction for Digital Conception and Manufacturing in Architectural Education”, Design By Making, ecaade2014, papers.cumincad.org, 2014.
- ⁷⁹ Chan A., Halevi T., Memon N., “Leap motion controller for authentication via hand geometry and gestures”, Springer, 2015.
- ⁸⁰ Vatuvu R.D., Zaiti I.A., “Leap Gestures for TV : Insights from an Elicitation Study” , University of Suceava, Romania 2014.
- ⁸¹ Marin G., Dominio F.; and Zanuttigh P., “Hand gesture recognition with jointly calibrated leap motion and depth sensor”, Multimedia Tools and Application, pages 1-25, 2015.
- ⁸² Marin G., Dominio F.; and Zanuttigh P., “Hand gesture recognition with jointly calibrated leap motion and depth sensor”, Multimedia Tools and Application, pages 1-25, 2015.
- ⁸³ Gergoulas C., Raza A., Güttler J., “Home environment interaction via service robots and the leap motion controller”, search.proquest.com, 2014.
- ⁸⁴ Mousavi Hondori H., Khademi M., “A review on technical and clinical impact of Microsoft Kinect on physical therapy and rehabilitation”, Journal of Medical Engineering, hindawi.com, 2014.
- ⁸⁵ Gauthier Y., Marcy J., Duprat D., “Interactions gestuelles pour StripTIC, un environnement tangible pour les contrôleurs aériens”, hal-enac.archives-ouvertes.fr, 2014.
- ⁸⁶ Donalek C., Djorgovski SG., Cioc A., “Immersive and collaborative data visualization using virtual reality platforms”, Big Data, ieeexplore.ieee.org, 2014.
- ⁸⁷ Boyali A., Hashimoto N., Matsumoto O., “Hand posture control of a robotic wheelchair using a leap motion sensor and block sparse representation based classification”, academia.edu, 2014.
- ⁸⁸ I Chatterjee, R Xiao, C Harrison, “Gaze + gesture: Expressive, precise and targeted free-space interactions”, Conference on Multimodal Interaction, dl.acm.org, 2015.
- ⁸⁹ S Aşut, *Design By Making*, eCAADe 32, papers.cumincad.org, 2014.
- ⁹⁰ The Razer Hydra is a motion and orientation detection game controller developed by Sixense Entertainment, https://en.wikipedia.org/wiki/Razer_Hydra, this page was last edited on 20 July 2017, at 20:43.
- ⁹¹ Sabir K., Stolte C., Tabor B., O'Donoghue I., “The molecular control toolkit: controlling 3D molecular graphics via gesture and voice”, IEEE Symposium, 2013.
- ⁹² Arduino est une marque qui couvre des cartes matériellement libres sur lesquelles se trouvent un microcontrôleur, <https://fr.wikipedia.org/wiki/Arduino>, la dernière modification de cette page a été faite le 27 mars 2018 à 21:07.
- ⁹³ Fonteles J. H., Silva E. S., Rodrigues M. A. F., “Gesture-driven interaction using the Leap Motion to conduct 3D particle system: Evaluation and analysis of an orchestral performance”, SBC Journal on Interactive Systems, 2015.
- ⁹⁴ Shao L., “Hand movement and gesture recognition using Leap Motion Controller”, Stanford EE267, 2016.
- ⁹⁵ Les machines à vecteurs de support ou séparateurs à vaste marge (Support Vector Machine : SVM) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression. https://fr.wikipedia.org/wiki/Machine_à_vecteurs_de_support, la dernière modification de cette page a été faite le 8 avril 2018 à 12:50.
- ⁹⁶ <https://developer.leapmotion.com/javascript/index.html>
- ⁹⁷ https://developer.leapmotion.com/documentation/skeletal/csharp/devguide/Leap_Tracking.html
- ⁹⁸ <https://www.leapmotion.com/>
- ⁹⁹ Shao, L., “Hand movement and gesture recognition using Leap Motion Controller”, Stanford EE 267, 2016.
- ¹⁰⁰ <https://www.developer.leapmotion.com/documentation>
- ¹⁰¹ Guna J., Jakus G., Pogačnik M., Tomažič S., Sodnik J., “An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking”, Sensors, 2014.
- ¹⁰² <http://blog.leapmotion.com/understanding-latency-part-1/>
- ¹⁰³ Weichert F., Bachmann D., Rudak B., Fisseler D., “Analysis of the accuracy and robustness of the leap motion controller”, Sensors, mdpi.com, 2013.

¹⁰⁴ F Weichert F., Bachmann D., Rudak B., Fisseler D., “Analysis of the accuracy and robustness of the leap motion controller”, Sensors, mdpi.com, 2013.

¹⁰⁵ Shao, L, “Hand movement and gesture recognition using Leap Motion Controller”, Stanford EE 267, 2016.

¹⁰⁶ Guna J., Jakus G., Pogačnik M., Tomažič S., Sodnik J., “An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking”, Sensors, 2014.

¹⁰⁷ Tung J.Y, Lulic T., Gonzalez D.A., Tran J., Dickerson C.R., Roy E.A., “Evaluation of a portable markerless finger position capture device: accuracy of the Leap Motion controller in healthy adults” ,Department of Mechanical and Mechatronics Engineering; Department of Kinesiology, University of Waterloo, Waterloo, ON, N2L 3G, 2015.

¹⁰⁸ Han J., Gold N., “Lessons Learned in Exploring the Leap Motion Sensor for Gesture-based Instrument Design” University College London, 2014.

¹⁰⁹ Bachmann, F Weichert, G Rinkenauer D, “Evaluation of the leap motion controller as a new contact-free pointing device”, Sensors, mdpi.com, 2014.

¹¹⁰ H Jin, Q Chen, Z Chen, Y Hu, J Zhang, “Multi-LeapMotion sensor based demonstration for robotic refine tabletop object manipulation task”, CAAI Transactions on Intelligence, Elsevier, 2016.

¹¹¹ <https://developer.leapmotion.com/documentation/>

¹¹² <https://developer.leapmotion.com/.../javascript/>