9-22-2016

Discovering Users' Intent in Interactive Virtual Environments

Frol Periverzov
frol.pv@gmail.com

Discovering Users' Intent in Interactive Virtual Environments

Frol Periverzov,PhD

University of Connecticut, 2016

Abstract:

A virtual reality(VR) environment is defined as a computer generated representation of reality that is sensitive to the actions of its observer. As the computing power of our machines follows an ever growing trend, the simulation power of our VR applications and their impact on the development of our society continues to grow in a remarkable fashion. Along with our computing capabilities, the data that needs to be spatially manipulated continuously increases in size and diversity. To keep up with this trend of increasing complexity we need to develop new 3D user interfaces (3DUIs) that allow users to employ the full manipulative capabilities of their natural hand gestures when manipulating such data. Today we can approach this goal by tracking the natural hand gestures of our users and inferring their manipulative intentions. However, human natural hand gestures exhibit a large variability that is aggravated by hand placement inaccuracies and body tracking uncertainties. Additionally, there is a non-unique mapping between human gestures and the underlying manipulative intentions.

In this dissertation I lay out the foundation of a general manipulative intention inference framework. New metrics are proposed for quantifying a set of human behavioral cues that characterize general goal directed actions. The relationship between these behavioral cues and a user's manipulative intent is modeled using machine learning techniques in novel fashion. The practical value of these techniques is demonstrated by developing new virtual object manipulation methods that are driven by intention inference. By means of intention inference, the proposed interaction techniques automatically adapt to the user's subjective needs for various enhancements such as hand placement fault tolerance and hand positioning precision enhancement. The performance of the resulting virtual object manipulation techniques has been tested in a statistically significant

manner by means of user studies.

The work presented here advances the state of the art in 3DUIs towards more user-friendly or even person centered user interfaces by developing user adaptable interfaces driven by intention inference. This can dramatically shorten the time required by a novice user to start performing efficient virtual object manipulations.

Discovering Users' Intent in Interactive Virtual Environments

Frol Periverzov

B.Sc. Technical University of Cluj-Napoca, 2008

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2016

APPROVAL PAGE

Doctor of Philosophy Dissertation

Discovering Users Intent in Interactive Virtual Environments

Presented by

Frol Periverzov

Major Advisor _____
Horea T. Ilieş

Associate Advisor _____
Krishna Pattipati

Associate Advisor _____
Kristine Nowak

Associate Advisor _____
Jiong Tang

Associate Advisor _____
George Lykotrafitis

University of Connecticut

2016

Acknowledgements

First and foremost, I would like to thank my major academic advisor Dr. Horea Ilies, Professor in the Mechanical Engineering department at UCONN. This dissertation thesis was made possible by his generous support and belief in my abilities. I feel grateful and privileged to have benefited from the freedom of development that I was offered under his supervision. At the same time, the insightful discussions we carried saved me from walking into sterile research fields. Also, his excellent writing style and the guidance I received helped me express the concepts I developed in a more rigorous, and eloquent fashion.

I would like to thank Dr. Krishna Pattipati, Professor of Electrical and Computer Engineering from whom I learned a large part of what I know in the field of machine learning. The Neural Networks and Optimization class along with the independent study in Advanced Probabilistic Approaches to Machine Learning that I have taken with Professor Pattipati have inspired the intention inference methods developed in this dissertation. I appreciate the guidance I received from Dr. Pattipati in the fields of his expertise as well as at a personal development level.

My gratitude also goes to my thesis associate advisors Dr. Kristine Nowak, Dr. Jiong Tang and Dr. George Lykotrafitis for their insightful and motivating feedback. I am very thankful to Dr. Kristine Nowak who also helped me strengthen the rigor of the user studies I ran and expand my research domain.

Special thanks go to Ian Greenshields, alumni Professor of Computer Science and

Engineering at UCONN who introduced me to image and signal processing concepts. This thesis heavily relies on such concepts when processing the human body tracking data to extract characteristic behavioral cues.

My thanks also go to Dr. Yaakov Bar-Shalom who helped me understand the physical foundations of statistical reasoning. The machine learning techniques used in this dissertation rely on a well-defined stochastic apparatus.

I am especially grateful to my wife, to whom I dedicate this work, for her unconditional support in all aspects of my life. With her by my side hard work is sweeter and the life-work balance takes a natural form.

Contents

Chapter 1

Introduction

In general terms, a virtual reality (VR) environment is defined as a computer generated representation of reality that is sensitive to the actions of its observer [1]. As the computing power of our machines follows an ever growing trend, the simulation power of our VR applications and their impact on the development of our society continues to grow in a remarkable fashion. The CAD/CAE environments used in mechanical, chemical, electrical, optical and other engineering fields are notorious examples. VR developments are changing various other domains like digital arts and entertainment, physics and human sciences (psychology, pedagogy, medicine) etc.

Simulation is the central purpose of any VR system. Simulations are not only used to represent reality but also to evaluate hypotheses about reality. Interestingly, in this hypothesis evaluation process we can incorporate ground truth, or real factors, that cannot be accurately defined. For example, the human behavior, the particle or/and wave behavior of light etc. Due to the interactive character of a VR simulation such real factors can be measured online and dynamically influence the outcome of the simulation by becoming an active part of the simulation process. As we can see, high fidelity interfaces between real factors and virtual environments open the door towards a new type of tools for exploring the unknown.

Along with our computing power the complexity of our simulations increases, and therefore, the data that needs to be spatially manipulated increases in size and diversity. To keep up with this trend of increasing complexity we need to develop new 3DUIs that allow users to employ the full manipulative capabilities of their natural hand gestures[1] when manipulating the 3D data. Software giants, such as Apple, Google and Microsoft, have patented their concepts of a 3D desktop interface for their operating systems, and the gaming industry appears to have the lead in commercializing novel 3DUIs for interaction with spatial data. The common agreement seems to be that human gestures form a powerful paradigm for building more intuitive 3D user interfaces for manipulating synthetic spatial information.

This, of course, is not at all surprising. We start 'waving' our hands and use hand gestures to interact with our 3D environment before we can speak. Later on, we use hand gestures when we tell stories, or provide nuances or deeply embedded categories of meaning (Figure 1.1), and there is data showing that hand gestures are closely related to our spatial perception and visualization [3].

Figure 1.1: Hand gestures have deeply embedded categories of meaning [4].

Consequently, one can conjecture that one of the most promising 3DUI paradigms for manipulating 3D information is one in which the user interacts with the spatial data with his/her bare hands, i.e., without the need of wearable hardware.

[1] In this manuscript the term *natural hand gesture* refers to those hand gestures that are commonly performed by people while trying to manipulate physical objects. A more detailed discussion can be found in [2].

There is a large amount of effort being spent on developing hand gesture-based 3DUIs for manipulating 3D information, and the available technologies are evolving rapidly. The earlier techniques used 2D images and required the user to wear wired or wireless hardware, which has been proving cumbersome and ergonomically challenging for any spatial tasks of reasonable complexity. On the other hand, the advances in computing power and computer vision hardware and software opened the door to novel 3DUIs that do not require wearable hardware, and hence, they do not restrict the hand movement. Nevertheless, these methods can be computationally expensive, and can be influenced by many of the standard environmental factors that affect the performance of computer vision systems.

Building fully automated systems for tracking and recognizing natural hand gestures and intentions requires robust and efficient 3D imaging techniques as well as potent shape classifiers. These tasks require hardware and software that must handle several key issues:

- Managing the frequent finger/hand occlusions;
- Embedding robustness against changing illumination conditions and background appearance;
- High-speed sensing: common human hand gestures may exhibit translational speeds up to 8 m/s and angular speeds up to 300 degrees/second[5, 6];
- Sensing resolution: hands and fingers have a relatively small size compared to the upper and lower limbs and fingers are often clustered;
- Modeling complex hand movements: the high-dimensionality of the models used for gesture recognition;
- Complex gesture semantics: gestures can be static, dynamic or both; are intrinsically ambiguous; vary from person to person as well as execution contexts.

Hence, any systems designed to track and recognize hand gestures must generate and handle large amounts of data in an efficient manner in order to minimize latency, and must be robust against changing environments as well as partial occlusions. By tracking and recognizing human gestures we can build interfaces that enhance users' ability to express themselves and manipulate virtual objects.

The existing methods for grasping and manipulating virtual objects without using hand held devices show two main trends and one of them heavily relies on recognizing symbolic gestures. More specifically this class of approaches includes methods in which the manipulation tasks are triggered by symbolic gestures such as a thumb up gesture used for object selection. We will use the term *symbolic methods* to refer to this class of techniques. The second group of approaches include those in which the interaction with the virtual objects relies primarily on simulating the effect of the contact forces that occur between the model of the hand and the virtual objects. We refer to this class of approaches as *physics dominant methods*. The simulations mentioned above requires accurate hand placement and collision detection. If we aim at avoiding the use of held devices that constrain the manipulation capabilities of our natural hand gestures the requirements listed above become prohibitive for reasons explained in section 1.1. Also, the freedom of expression offered by the symbolic techniques is not sufficient to enable our users to employ the full manipulative capabilities of their natural hand gestures unless the system is capable to interpret users' natural hand gestures and infer their manipulative intentions.

In this thesis I demonstrate that, by relying on body tracking techniques based on 3D imaging along with adequate machine learning methods, we can robustly infer the manipulative intentions of the user while natural hand gestures are employed to manipulate and assemble virtual objects.

1.1 Challenges

Several important challenges need to be overcome in this endeavor. First, it is well-known that without physical support and/or haptic feedback, the user has difficulties in placing and holding his/her hands at the precise location required for virtual manipulation [7, 8, 9]. A common consequence of this problem is the penetration of the virtual objects that the user intends to grasp [10]. To alleviate this issue, the 3DUI must tolerate hand positioning imprecision and compensate for the lack of haptic feedback during the virtual object manipulation procedure. At the same time, such a system should afford the user a detailed space/depth perception.

The second important group of challenges is posed by the inherent noise and uncertainties that affect the 3D imaging methods and the related stochastic tracking algorithms. These *tracking uncertainties* occur mainly for parts of the body that are affected by occlusion, imaging noise, low reflectivity, light glare, or body parts that cannot be distinguished from others due to the perceived similarities caused by low imaging resolution or other factors.

Furthermore, the natural gestures used to perform the same manipulation tasks show large variability when executed by different people or even when performed by the same person but in different manipulation contexts. For example, you can look at the various ways in which an object can be grasped [11], pushed, etc. This large gesture variance, magnified by the effects of the aforementioned tracking uncertainties and hand placement faults, is increasing the ambiguity of the observed gestures and, therefore, the difficulty of the gesture classification task, particularly when it has to be accomplished online, in real time.

1.1.1 Virtual Object Selection

The selection task is one of the most common duties of our daily life. We select/choose our paths, our goals or objects of interest each time we decide to pursue a goal or to interact with this abstract object called "goal". Similarly, the selection of virtual objects is a common task of paramount importance for any virtual object manipulation process. The efficiency of the selection procedure directly impacts the performance of all other manipulation tasks such as virtual assembly. In this thesis I propose a new selection method that allows users to employ natural hand gestures in free space to select virtual objects. This virtual object selection method facilitates the manipulation of virtual objects by means of natural hand gestures, and does not require the use of any hand held devices that would constrain the manipulative capabilities of the user's hands.

To achieve these, we resolve the challenges listed in section 1.1 in a novel manner. The proposed virtual object selection technique identifies the objects that are targeted during the selection process by relying on a set of behavioral cues that have been documented in the neuropsychology literature for general goal directed actions. Such behavioral cues enable our method to tolerate hand placement and tracking faults. Some of the cues documented so far include facial cues [12], action efficiency [13, 14], action persistence [15, 16], effort invested, action duration [16], etc. By means of user studies we evaluate the relevance of two of the most promising cues in the context of the virtual object selection tasks. Specifically, our action efficiency cue estimates the effort required to select an object, while our action persistence behavioral cue estimates the level of perseverance with which the user tries to select a particular object. User studies show that by relying on the action efficiency cue our method affords the selection of objects that have their largest dimensions as small as 0.6 cm even when they are located in environments in which the distance to neighboring objects is approximately 0.1cm. Furthermore, embedding the action persistence

cue along with the previous behavioral cue into our selection method enables users to select objects 45% faster and more efficiently than the case in which the action persistence cue is removed. The persistence cue allows our methods to detect the targeted objects during challenging selection tasks, when users show jittery or hesitant hand movements, in spite of the tracking noise that affects our system.

1.1.2 Gesture Recognition and Intention Inference

The aforementioned gesture variability that becomes aggravated by the unavoidable hand placement and body tracking faults, makes the gesture recognition endeavor a challenging task. In this thesis I propose the use of characteristic behavioral cues that have been documented in the neuropsychology literature for specific manipulative intentions and gestures to develop metrics for classifying the observed hand movement into motion primitives corresponding to manipulative gestures. I demonstrate the practical advantages offered by these behavioral cues by developing new virtual object manipulation techniques based on them. The proposed techniques are designed to compensate for the tracking instabilities introduced by the imaging methods, and for the problem of the loss in hand positioning precision shown by users in these virtual environments.

To achieve these we perform the action/task segmentation by employing an efficient task boundary[2] detection approach which uses the behavioral cues shown during general arm reaching motion as our boundary features. As shown in the section 5.1, the arm reaching movement proves to be a good action segmentation feature for general object manipulation, such as assembly tasks, as well as in more common activities such as placing a virtual mug on a virtual table.

[2]The term task boundary refers to the boundary of the time interval corresponding to a specific gesture.

1.1.3 Virtual Object Assembly

In order to enable users to assemble virtual objects in a natural manner, our system needs to estimate which are the elements of the manipulated objects that our users intend to couple, as well as the particular manner in which they intend to couple these elements. The *coupling elements* are geometric primitives like faces, edges and vertices which can be used to build 3D objects of arbitrary complexity. It is worth noting that in the case when we attempt to predict the fashion in which the user intends to assemble two simple parallelepipeds or brick models, there are 52 coupling elements that can join in hundreds of different ways. In the context of the aforementioned hand placement and body tracking faults, estimating the manner in which the user intends to assemble virtual objects becomes a complex problem.

To estimate the particular manner in which our users intend to assemble virtual objects, we model their behavior in a novel fashion. Namely I use a CRF probabilistic graphical model whose potential functions are defined based on behavioral cues that are representative for general goal directed actions. In order to evaluate the strength of such behavioral cues I have developed a set of new metrics. The resulting technique enables natural virtual object assembly procedures in which every geometric primitive of the manipulated object can be coupled with any other geometric primitive. The performance of the proposed constraint recognition method has been tested by means of user studies and the results show that this method offers a success rate 25.78% higher than current state of the art alternatives.

1.2 Summary of Contributions

The virtual object manipulation methods that I propose in this thesis incorporate and balance the strengths of the physics dominant manipulation techniques and the symbolic manipulation methods while alleviating their key shortcomings. Specifically,

our techniques provide a wide spectrum of the manipulation versatility and the natural form of the physics dominant methods, while approaching the levels of robustness that can be achieved by symbolic methods, and offer a built in tolerance to the user's hand placement and body tracking faults.

The intention inference techniques developed in this manuscript are robust against common gesture variability and support the use of 3D imaging techniques for tracking the user's body by offering a much higher tolerance to tracking uncertainties than the physics based methods. Importantly, our virtual object manipulation methods do not constrain the user to place his/her hands with the same level of precision as the physics dominant methods do, and offer a higher computational efficiency by relying on much coarser collision detection queries as explained later on. I demonstrate that these attributes enable our methods to afford common types of manipulations that are difficult or even impractical to achieve with the physics dominant methods. The user studies described in this manuscript prove that our methods are much faster and more efficient than the physics dominant methods. If compared with the symbolic methods, our techniques do not confine the users to employ a predefined set of gestural symbols as the symbolic methods do, but allow instead the use of free natural hand gestures. Therefore, our system requires a lower learning effort from the user and affords much more flexible manipulation procedures.

The virtual object selection technique I introduce here affords the use of natural hand gestures to select virtual objects. Our user studies show that the smallest objects that can be repeatedly selected are spheres of 0.6 cm diameter located in an environment in which the distance to the neighboring objects is approximately 0.1 cm. This performance is achieved in spite of the fact that the depth sensing resolution of the used body tracking technique is 1 cm. We arrive at this result by developing a seamless selection disambiguation method which does not remove the environmental context during the selection procedure. Furthermore, the conceived

method automatically adapts to the user's subjective need for hand placement fault tolerance.

I estimate the manner in which our users intend to assemble virtual objects by relying on machine learning techniques to evaluate the correlation between the proposed behavioral cues and a user's intent. The resulting technique enables natural virtual object assembly procedures in which every geometric primitive of the manipulated object can be coupled with any other geometric primitive. When compared to the existent constraint recognition techniques the proposed method shows an average success rate that is 25.78% higher and a stronger assembly intention disambiguation.

These performances are made possible by the novel use of characteristic behavioral cues which have been documented in the neuropsychology literature for specific manipulative intentions and gestures to develop new metrics for classifying in real time the observed hand movement into motion primitives corresponding to manipulative gestures. The cues consist of characteristic motion features such as trajectories, speed or acceleration profiles that are shown by the user's body during virtual object manipulation. As explained section5.1, the proposed behavior cues embed the following strengths into our intention inference methods:

- Tolerance to hand placement faults: The cues do not depend on absolute position parameters, and therefore, they can be identified even if a user's hands end up in a misplaced position.

- Robustness against tracking uncertainties: Due to the large volume of information that defines these behavioral cues, the likelihood that tracking uncertainties will affect a significant part of the relevant tracking data is strongly reduced.

- Robustness against gesture variability: Our discriminative cues are representative for general forms of specific natural gestures like hand reaching [17, 18, 19, 20], grasping [21] and others. This means that our behavioral cues are present

in these specific gestures across most of the forms/variations that these gestures can take. Therefore, our system is able to infer the manipulative intention by analyzing the user's natural hand gestures.

Importantly, part of our behavioral cues are characterizing general goal directed actions. Therefore, such cues can be applied to automatically identify common types of general intentional actions. Furthermore, the probabilistic graphical model presented in this thesis is capable of approximating the relationship between quantifiable factors of arbitrary nature and human intent.

In this thesis I introduce a set of techniques that classify and handle in *real time* 6 manipulative intentions: grasping, assembling intention, grasp release, guiding push, precise hand positioning and punching. Each of these intentions controls a set of virtual object manipulation methods whose performance has been empirically tested and documented. The work presented here advances the state of the art in 3DUIs towards more user-friendly or even person centered user interfaces by developing user adaptable interfaces driven by intention inference. This can dramatically shorten the time required by a novice user to start performing efficient virtual object manipulations.

Chapter 2

Related Work

2.1 Virtual Object Selection

There are two main approaches used to select a virtual object: the virtual hand selection metaphor and selection by pointing. In the first case the selection is performed through distance evaluations between a virtual hand model and the surrounding objects, while the latter approach measures the proximity with respect to a ray that is defined implicitly by the user. For example, the ray direction can be provided by the line that joins two points on the user's body, such as the eye to hand tip direction, or it can be projected from a tracked device, such as a stylus. The virtual hand selection metaphor affords the use of natural gestures, while the virtual pointing approach can offer selection procedures that lower the arm fatigue at the expense of a less natural selection procedure [22]. Many of the published selection methods use as input devices hand held hardware, which constrain the manipulative capabilities of our natural hand gestures. Since we are interested in developing 3DUIs that offer the manipulation flexibility provided by our natural hand gestures, we will mainly focus on virtual hand selection (VHS) approaches that do not make use of such hand held devices.

The VHS method is used in most 3DUIs in which the manipulation of virtual objects is controlled by simulating the physical interactions between a virtual hand model and the manipulated objects. These techniques usually use wearable input devices such as data gloves. In its most simple form the VHS method will select the objects that intersect the virtual hand model [23, 24]. The Go-Go technique [25] adopts a similar approach, but it allows the user to select objects outside the volume defined by the arm reach by elongating the virtual representation of the arm. In [26] the selection is activated once the virtual hand model intersects the object(s), and a pinch gesture is detected, while the 3D Bubble Cursor [27] method selects the object that is closest to the center of a selection sphere.

In [28] an abstract selection model is presented that aims at representing a large group of existing selection techniques. The model is composed of two main factors: (1) The relative position between some selection volume and the object that is targeted during selection, and (2) An abstract function of the history of the two factors. This model has a promising power of representation, but has not been tested yet. The work in [29] is concerned with identifying the movement phases of the users' hands during general selection tasks. This article offers a comparative analysis between the behavior shown by users while reaching to select objects in real environments and virtual environments.

The methods presented in [30] use as input the data offered by a Kinect camera, and do not require the use of any hand held devices. The selection is accomplished by using a selection cone whose apex is kept fixed while the center of the cone base can be moved in a vertical plane by the movement of the user's hand. The objects that are intersecting the cone can be selected, and a menu based selection disambiguation method similar to SQAD [9] is employed.

2.1.1 Selection Disambiguation

With the method mentioned above [30] the user can perform a hand pull gesture in order to display a 2D menu that lists the objects being intersected by the selection cone. Then the selection is accomplished by picking one of the listed items. The smallest objects that were selected using this method were spheres of 10cm diameter placed at a minimum distance of 30cm from all other objects. While such menu based disambiguation methods can be extremely accurate, they remove the environmental context from the selection procedure, and reduce the user's sense of presence in the virtual environment.

The Expand method [31] is addressing this problem by displaying the objects that intersect a 3D cursor in a grid pattern that overlays the image of the virtual environment. The selection is then completed by pointing a hand held controller towards the targeted object. The IntenSelect [32] method projects a selection cone from a hand held stylus, and the selection disambiguation is accomplished using a scoring function, which depends on the location of each object with respect to the cone axis and apex, previous scoring values and other tunable factors. In the Starfish [33] method the four closest objects to a 3D cursor are joined by a guiding surface. Once a target object is intersected by this surface, the user can press a button to lock the position of the guiding surface. Then, the 3D cursor is constrained to move inside this guiding surface. In this manner, the effort of positioning the 3D cursor with the high accuracy required by certain selection cases is significantly reduced. On the other hand, the steps involved in the selection process do not allow us to interact with virtual objects by means of free natural gestures. There are many other pointing techniques that have been proposed [34, 9], and most of them are reviewed in [22].

2.2 Grasping and Manipulating Virtual Objects

Grasping virtual objects using our natural hand gestures while relying on 3D imaging to track our body parts can prove surprisingly difficult due to the hand placement and tracking faults discussed in section 1.1. Unless otherwise specified, all techniques discussed below require the user to be geared with physical sensors for body motion tracking, and/or data gloves for finger motion tracking.

As explained in section 1 the existing methods for grasping and manipulating virtual objects without using hand held devices can be classified into two categories: the physics dominant methods and the symbolic techniques. The aim of the physics dominant methods is to achieve a high degree of realism in the interaction with virtual objects through faithful simulation of the physical interaction. However, since the standard friction models depend on the normal vectors at the point(s) of contact, tracking the user's hand must be accurate and robust. The main advantage of the physics dominant methods is that they, in principle, allow any physically correct grasps. On the other hand, such a grasp requires an accurate and precise positioning of the hand model [23, 24], which is made difficult by the tracking uncertainties and the loss in hand placement precision discussed in the introduction. One of the common consequences of this lack of accurate grasp configuration is the penetration of the virtual object by the hand model, which is typically approached through the use of a spring hand model. For example, in [35] the object penetration by the hand model is prevented, while the system forces the hand model to remain on the surface of the object that would be otherwise penetrated. Unfortunately, such an approach prevents other basic manipulative gestures like push or punch. This issue is alleviated in [24] where the virtual objects are built of subsets that respond to either push or grasp. The objects that are too small to be partitioned into functional subsets are defined as either graspable or pushable.

In order to improve the robustness of the grasp intention detection in [26] the

collision detection is combined with sensor based pinch detection. A grasp gesture is detected between certain fingers if a pinch gesture was detected between these fingers and at least one of these fingers collides with the grasped object. This approach, like many others, trades off the versatility and the natural form of the grasping gesture for a triggering mechanism that is simpler and more robust.

Data gloves are used by [36] to measure the amount of flexion shown by each finger during grasping. Using this raw data as feature vectors they have tested 28 classifiers in an attempt to classify 6 representative grasp types. Promising results have been documented, revealing the behavior of each classifier with respect to this problem. However, generalizing this approach to unrestricted grasping gestures, which could include grasp types on which the classifier had not been trained, could prove difficult.

To increase the precision with which virtual objects are manipulated once grasped several techniques are proposed in [7], where the operator uses his hands in an asymmetric manner. For example, a precise rotation is executed by using the right hand to select by pinching the targeted object, while the motion of the left hand is mapped to a rotation. In this manner, large arm movement can be used to control a fine rotation of the object. Another symbolic approach is proposed in [37] where the finger pointing gesture is used for object selection. We note that the symbolic manipulation methods tend to be more robust than the physics dominant methods both for grasping as well as manipulation. Symbols are typically chosen to intuitively suggest their function (think of shaking a smart phone to shuffle a playlist) or even gestures that are relatively 'close' to the motion shown by the user's hand when a similar manipulative action is carried out in the physical world. For example, the open/closed states of the user's hand are being identified in [38] by relying solely on 3D imaging data. This symbolic gesture is used to trigger the grip or selection of the objects in the proximity of a 3D cursor. Broadly, all symbolic methods use predefined symbols to trigger virtual actions, which do not capture the invariable characteristics of a general

natural gesture. For example, the user is required in [37] to point the index finger at all the objects that have to be grabbed, which can be performed rather robustly. By contrast, the physics dominant methods offer, in principle, a more flexible and realistic interaction, but are less robust today than symbolic methods.

Given the advantages and disadvantages of various interaction methods, a compromise is proposed in [8] where the interaction technique is selected based on the context in which the manipulation is performed. The computing context is defined by components such as the required level of control needed in manipulation, the manipulation workspace, the frame of reference with respect to which the task is performed and so on. Each particular manipulation context is specified by using explicit menus and widgets while for automatic context recognition this article mentions the Go-Go [25] and PRISM [34] techniques.

2.3 Assembling Virtual Objects

Among the different techniques that have been developed to assemble virtual objects, there are two main trends which are similar to the trends described in the previous section. Namely we have the physics based assembly methods and the constraint based assembly. The physics based methods fall into the category of the physics dominant manipulation methods while the constraint based assembly techniques are focused on identifying the assembly constraints which the user intends to apply during the assembly procedure. Considering our goals and the previously explained limitations of the physics dominant methods we will focus our manuscript on constraint based assembly techniques. Detailed reviews on physics based assembly techniques can be found in [39, 40].

In the early constraint recognition approaches a constraint was selected if there was an overlap between the bounding boxes of the two body parts which support

the constraint [41, 42]. Most of the current constraint recognition methods select a specific constraint if the value of a specific distance function that was defined for that constraint falls in a certain range of values. Such distance functions are often built out of the geometric parameters that characterize a particular constraint type. For example, a coplanar constraint is selected if the angle and the distance between the evaluated planes fall within certain ranges [43, 44, 45]. Therefore, in order to build an assembly, the users need to position the manipulated objects such that the constraint they intend to apply is closer to being satisfied than any other potential constraint. This task becomes challenging if the objects that need to be assembled support a moderate number of constraints. In this manuscript we will use the term *geometry dominant constraint recognition* to refer to this class of approaches.

If we aim to enable natural or flexible assembly procedures in which every geometric primitive can couple with any other geometric primitive, the number of potential constraints that can occur increases dramatically even for simple object models. For example, a parallelepiped or a brick model has 26 primitives (such as faces, edges and vertices) and each primitive may couple with the primitives of the manipulated bodies in multiple ways. In the context of the aforementioned hand placement and body tracking faults, the large number of potential constraints that are supported by such primitives becomes a challenging factor. For example, if we try to partially overlap two faces of two brick models, it is likely that a corner of one face will touch the other face before the faces overlap. Therefore, a constraint recognition method that is based solely on geometric parameters will infer that we are trying touch one face of one body with the corner of the other body although we do not aim to enforce that constraint. As you can see, the pure geometric functions are not sufficient to infer user's assembly intent, although they are important. Therefore, we extend the current state of the art constraint recognition techniques by modeling user's behavior using a probabilistic graphical model which relies on human behavioral cues as well

as constraint specific distance functions.

Chapter 3

Input Technologis for 3DUIs

Various 3D user interfaces have been proposed so far, and based on the afforded freedom of expression they can be arranged in the hierarchy shown in figure 3.1.

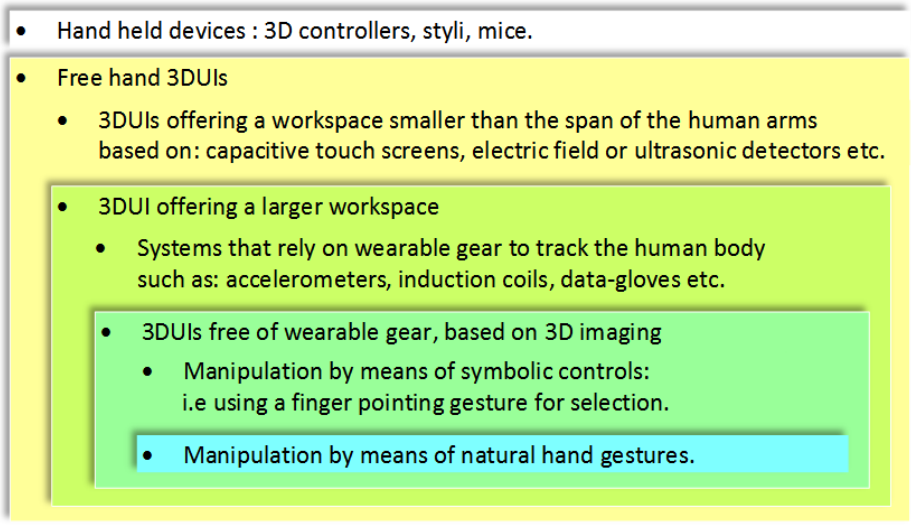

Figure 3.1: A 3DUI hierarchy based on to the freedom of expression they afford.

We see that the type of human activity sensing solution strongly affects the capabilities afforded by the UI. Therefore, in this section we will present an in depth analysis of the capabilities shown by the different 3D imaging techniques.

3.1 Commercially Available 3DUI Solutions

3.1.1 3DUIs Based on Hand Held Devices

Probably the earliest HCI designed for manipulating 3D information is the 3D mouse. This solution constrains the user to maintain contact with the input device. The 3D mice have been designed to enable their user to employ fine finger movement in order to manipulate 3D data, but its functionality is limited to standard tasks like select, pan, zoom and rotate the model or camera. There are many variations available on the market such as [46, 47, 48, 49]. The electronics and gaming industries have begun to commercialize a set of hand-held devices that offer control of multiple degrees of freedom, ranging from 3D pointing devices to 6 DOF input devices such as Nintendo's Wii and Sony's Playstation controllers. These handheld devices are effectively complex 3D mice and are not designed to capture natural hand gestures.

A new set of controllers are those providing not just multiple DOF input, but also haptic feedback through spatial forces and even torques around the three coordinate axes. Several manufacturers provide a wide range of solutions for the personal and professional uses such as Novint [50], Sensable [51], and Haption [52]. These devices are actuated mechanisms and their workspace is limited by the specific geometric limitations imposed by the kinematics of the mechanisms.

Data Gloves have been proposed as an input device for capturing the motion of a user's hand and finger. As the name suggests, these HCIs are wired gloves that sense the position and orientation of the hand in 3D space, as well as the data needed to compute the angles representing the bending of fingers with a resolution higher than 1 degree. There are both wired and wireless versions available [53, 54, 55, 56], but the user must carry the hardware that may include the power source for the wireless transmitters. Furthermore, data gloves attached to a powered mechanical exoskeleton have been developed for applications requiring haptic feedback. These HCIs can offer

a very large workspace and high data tracking resolution. The downside is the impact on usability driven by the often heavy hardware that the user is required to attach to his/her hands, which, in turn, leads to the so called 'Gorilla Arms' when used for a prolonged period of time.

3.1.2 3DUIs Free of Hand Held Devices

3DUIs with a Workspace Smaller than the Span of the User's Arms

- *Capacitive Touch Screen:* Cypress introduced TrueTouch [57], which is a touch screen able to detect the 3D position of a body that is placed within 5 centimeters from the screen. The sensing is done by means of an array of capacitive sensors integrated into the screen. Due to the fact that the entire interface can be included into a touch screen, it could be used as a 3D hand interface for portable devices as the tablet PCs. However, the relatively small workspace places limitations on the usability of this interface and the range of gestures that can be used for manipulating 3D information.

- *Ultrasonic Detectors:* Both Elipticlabs [58] and Nokia have patented their versions of a gesture based touchless interface that can detect the position of the user's hand in a space located within 20 cm from the sensing device. Their interface is based on ultrasonic emitters that can detect the position of the hand by analyzing the acoustic wave reflected by the hand, and the distance is computed by using the standard principle of triangulation. Elipticlabs have demonstrated to date the detection of waving gestures, and the movement of the hand towards or away from the ultrasound receivers.

- *GestureCube:* IDENT Technology AG is developing GestureCube [59], a commercial product in the shape of a cube that has mounted displays on 5 of its sides and incorporates several multimedia features. The sensing uses an electric

field around the cube and the sensors read the disturbances in that field created by the user's hands. The most interesting aspect seems to be that its user interface is able to detect hand gestures in close proximity of the cube. Although the GestureCube is not designed to be used as an HCI for manipulating objects in 3D, it was demonstrated that it can detect translational and rotational hand movements. This interface can be used to detect some finger movement as well.

- *The Bi Directional (BiDi) Screen* [60] has been developed by MIT Media Labs and relies on one of the technologies currently used in multitouch LCD screens. The detection of multiple simultaneous touching events occurs by using an array of light sensors integrated into the pixel array of the screen. The depth of view of this sensor array has been recently increased, which allows the sensors to detect the depth map of an object located in front of the screen at a distance smaller than approximately 50 cm. A commercial product with similar characteristics is built by Evoluce [61], but the company does not provide the details of the sensing technology being employed. Nevertheless, is has been demonstrated that by using this HCI one can detected gestures like waving left-right, up-down as well as and and the movement of the hand towards or away from the screen. Due to the relatively larger sensing range, this technology might enable the detection of more elaborate gestures.

3DUIs With Larger Work Space

- *Data Gloves:* As the name suggests these HCIs are gloves that provide information about the position and orientation of the hand in the 3D space. Such technology can measure the bending of the fingers with a resolution higher than 1 degree. Several models come with the option of attaching a powered mechanical exoskeleton that can offer haptic feedback for the user. However such exoskeletons often introduce fatigue and become cumbersome due to their large

size.

- *Marker-Based Tracking Systems:* Oblong Industries has released the g-speak^{TM}[62] platform that can be used to detect the user's hand gestures in a virtually unlimited by tracking passive markers. The position of these markers can be determined in space by performing photogrammetry of data streamed from 2 or more video cameras. Oblong Industries implements this method by incorporating the reflective markers into a regular thin glove which the user needs to wear while manipulating virtual objects. This system can track finger gestures in a space limited only by the number of the video cameras used.

Specialized Software Development Kits (SDK)

There are several SDKs that have been launched on the market in the past few years that can be used, in conjunction with appropriate 3D imaging techniques, to track the motion of the human body and provide the kinematic parameters through an articulated kinematic model of the body. Importantly, these methods do not require wearable hardware to perform the tracking. There are four such SDK released so far, namely iisu^{TM}by Softkinetic [63], Kinect for Windows by Microsoft [64], Maestro3D by Gesturetek [65], and Bekon by Omek [66]. Microsoft Research has recently introduced KinectFusion [67] , which uses the depth data from a handheld Kinect sensor to track the sensor and reconstruct the physical scene in real time on a GPU implementation. Moreover, several other companies, such as Mgestyk [68] are working on developing their own SDKs and we expect quick developments and a very strong competition in this area for the next several years. These SDKs support several 3D cameras that are commercialized by several companies at the time when this survey was written, such as PrimeSense [69], Baumer [70], Canesta, MESA Imaging [71], PMD Tec [72], Panasonic [73], Optex [74], SoftKinetic [75] and LeapMotion [76]. The 3D cameras commercialized by these companies are using Time Of Flight (TOF), structured light

(SL) or active illumination stereo 3D imaging principle, which are three of the many 3D imaging techniques. These and other methods are analyzed in section 3.2.

In principle, systems built using TOF, SL or active illumination stereo cameras along with the SDKs mentioned above should be capable to detect hand and finger gestures without requiring wearable hardware in a workspace that is sufficiently large for most practical applications. In the next section we analyze these sensing techniques as well as the potential alternatives with respect to performance parameters specific to our hand gestures tracking task.

3.2 State of the Art 3D Imaging Techniques

Three dimensional imaging of physical objects has been a very active area of research for over two decades and is going through a period of heightened attention due to multiple advances in imaging hardware, software and computing power. The basic task of 3D imaging methods is to construct a geometric model of a physical scene being observed.

In the broadest sense, the existing approaches to 3D imaging either require physical contact with the object, such as the coordinate measuring methods, or compute geometric properties from data collected by non-contact sensors as summarized in Figure 3.2. Since we focus here on 3D imaging for hand gesture recognition, we do not discuss the approaches that exploit the transmissive properties of the objects being observed, or those that rely on non-optical reflective properties of the object surface.

All optical 3D imaging methods analyze interactions between (electromagnetic) radiation and the scene under observation. The passive approaches exploit the existing illumination in the scene being investigated, and tend to work 'well' under near-ideal illumination conditions. These methods look for visual cues in the images

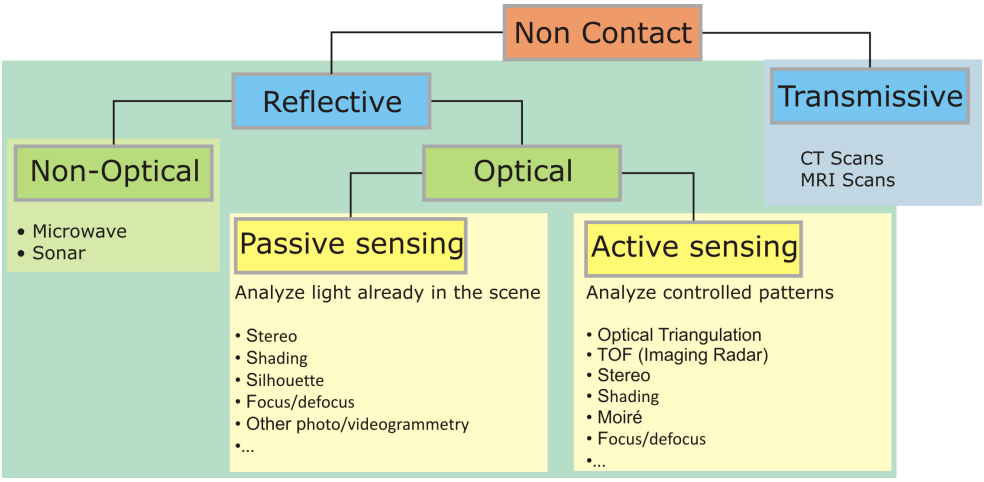

Figure 3.2: Taxonomy of 3D imaging approaches after [77].

or sequence of images that they operate on to extract the geometric properties of the object. On the other hand, the active imaging methods project an electromagnetic wave, typically in visible or infrared spectrum, onto the scene and measure the changes in specific properties of the reflected waves, which are then mapped to geometric quantities. As a general rule, the performance of all optical imaging methods depends on the illumination conditions and specific surface properties of objects, such as differential properties, reflectance, and opacity, as well as specific hardware capabilities. A detailed historical overview of the active methods can be found in [78].

For the rest of this section, we present the working principles of those methods that can have the largest impact on our task of 3D imaging of hand gestures, and analyze the relevant differences of potential approaches with respect to factors affecting the resolution, reliability, and computational cost of the algorithms that are needed to process the raw data output by the accompanying sensors.

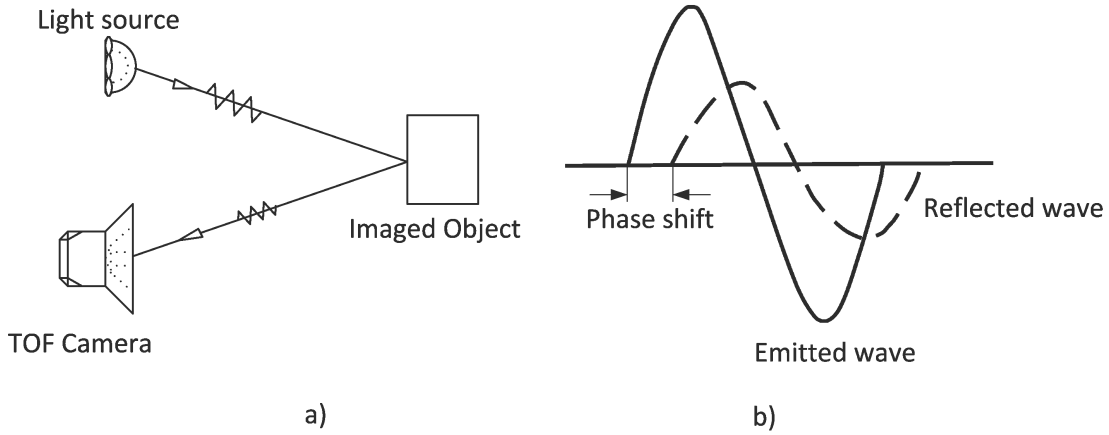

Figure 3.3: TOF range measurement principle: a) Typical hardware setup; b) The phase shift measurement

3.2.1 3D Measurement Principles

Time of Flight (TOF) 3D Imaging

The TOF measurement principle relies on measuring the time that a controlled, typically electromagnetic wave needs to be reflected by the target object and reach the sensing device [79] as illustrated in Figure 3.3. Due to the fact that the speed of light is constant, this time measurement can be easily transformed into a distance value. The measurement procedure goes through the following steps: a modulated wave is emitted from the source; the wave reflected by the object and captured by the sensing device has the same frequency as the emitted wave, but a smaller amplitude and a different phase [80]; the phase shift between the emitted and captured waves, which is proportional to the distance between the TOF sensor and the object, is mapped to a distance value.

The TOF cameras capture entire scenes with each laser or light pulse rather than performing sequential scanning. The time of flight is measured from either phase differences between modulated emitted and imaged pulse captured by CMOS or CCD sensors or a dedicated shutter system [81, 82]. These TOF cameras output evenly

distributed range and intensity images, and avoid the correspondence problems of stereo vision or structured light systems as discussed below and in [83]. Furthermore, the surface reflectance of the objects in the scene has a much smaller influence on the range data output by the TOF cameras than for other optical systems.But as described in [79] the depth measurement resolution depends on the modulation frequency and the measurement non ambiguity range(NAR) trough law 3.1 where φ is the measured phase shift. As an example a resolution of 1 cm can be achieved for a working range of 0.3-7m [72]. To double the resolution to 5mm we need to halve the depth of the camera's working range.

$$\Delta R = NAR \cdot \frac{\Delta\varphi_0}{360^o} \tag{3.1}$$

Optical Triangulation with Laser and Structured Light

Triangulation is one of the fundamental principles used by a number of range sensing techniques, and laser triangulation is one its most common applications. Lasers are compact, and offer great control of both wavelength and focus at large distances. There are many embodiments of this principle that differ in the type and structure of the illuminating source, and of the sensor.

The measurement procedure, as described for example in [84], uses a laser module as a light source and a regular video camera that are set up as shown in Figure 3.4. All geometric parameters that are known at the start of the measurement procedure are shown in `blue`, including the relative position and orientation of the laser module with respect to the camera. The laser module projects a planar laser 'sheet' onto the inspected object. Point S' represents the image of point S as 'seen' by the camera, and, consequently, its image plane coordinates are known. By determining the angles $\angle SFL$ and $\angle FLS$, and by observing that distance LF is known, one can determine the spatial coordinates of S from the LSF triangle. The coordinates of the visible

boundary points are obtained by repeating this process while sweeping the object with the laser plane. Some of the typical concerns of these popular methods are the time needed to mechanically sweep the scene, eye safety when dealing with human users, as well as noise and depth resolution.

Rather than projecting a laser plane onto the object, one can project a 2D pattern onto the inspected surfaces, or so called 'structured light', and use measured changes in the reflected pattern to compute distances [85]. Some of the commonly used patterns are fringes [86], square grids [87] and dots [88], while the wavelength of the structured light can be inside or outside the visible spectrum. These methods need to assign each element of the captured light pattern to the correct element of the emitted light pattern as illustrated in Figure 3.5. Establishing this correspondence is one of the major challenges of this technique, and several pattern coding strategies have been proposed. These solutions strongly affect the performance of the structured light imaging technique as discussed below. Methods based on structured light tend to have low spatial resolution as patterns become sparser with distance [89].

Stereo Vision

The typical stereo technique uses 2 cameras whose relative position is known, and looks for the differences in the images observed by the cameras in a manner similar with how human vision functions. The stereo vision can either be: (a) passive, case in which the features observed by cameras under natural illumination conditions are matched, or (2) active, by using structured light to improve feature matching. Both approaches use the triangulation principle to produce the depth map. The two-camera active stereo vision suffers from ambiguity in the interpretation leading to false matches [77], which can be improved by using multiple cameras [90], which is known as photogrammetry.

The construction of the depth map depends on matching the image points cap-

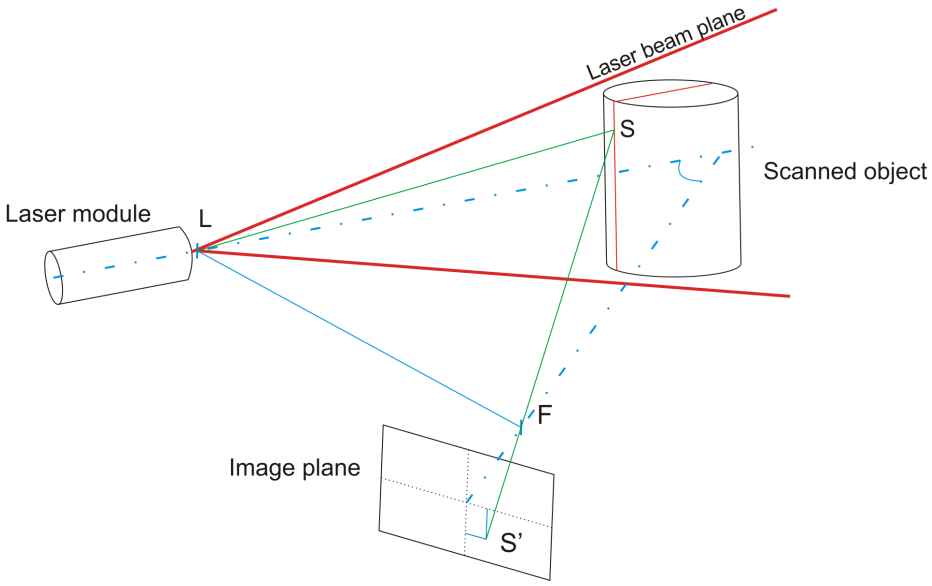

Figure 3.4: The principle of laser triangulation

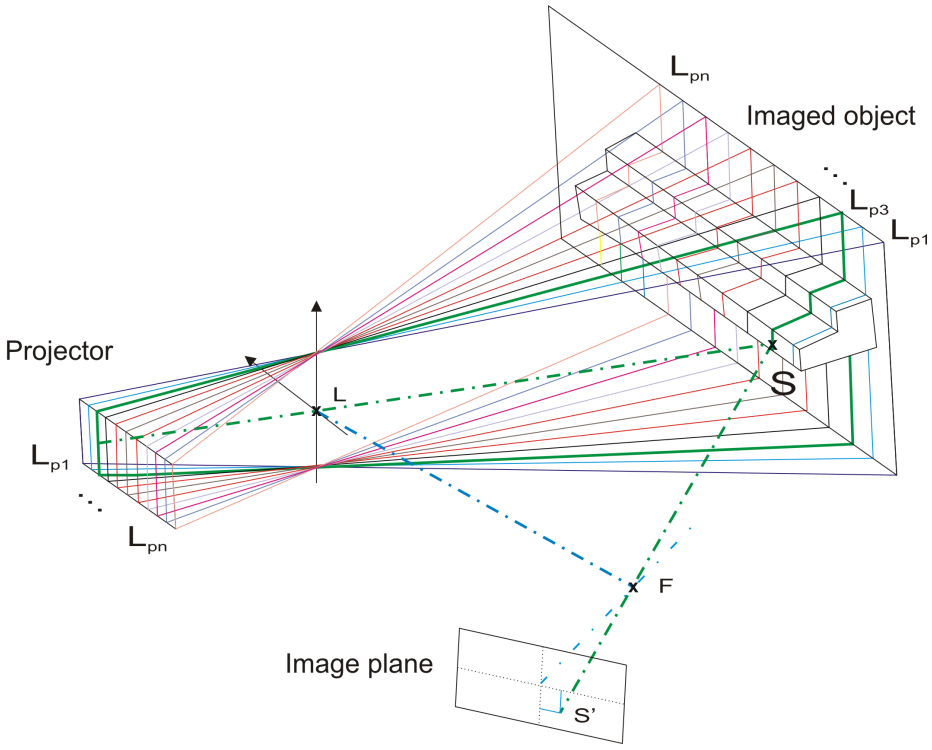

Figure 3.5: Structured light 3D imaging with fringe patterns.

tured by the cameras to the corresponding physical point, and uses the triangulation principle. As shown in Figure 3.6, the image planes may be aligned[1], and the range information Z is obtained from two similar triangles SS'S" and SO_1O_2 as follows:

$$d = x_1 - x_2 \qquad \frac{b}{Z} = \frac{b + x_1 - x_2}{Z - f} \qquad Z = \frac{b \cdot f}{d} \tag{3.2}$$

For a 2-camera passive setup, the accuracy of the depth measurement decreases according to a quadradic law:

$$\partial Z = \frac{-Z^2}{f \cdot b} m \tag{3.3}$$

where **f** is the focal length of the lenses mounted on the two cameras; **b** is the stereo baseline; and **m** is the correlation accuracy that depends on the specific resolution [91].

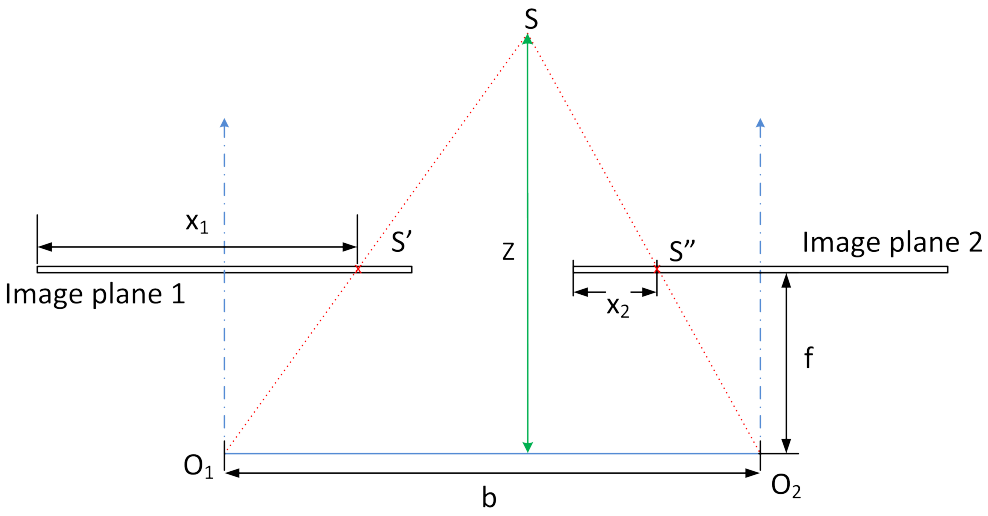

Figure 3.6: The triangulation for a 2-camera passive stereo setup

The passive (dual or multi-view) stereo systems are using triangulation algorithms that require feature matching among the various viewpoints, which is an open problem

[1]Note that non-aligned image planes is sometimes known as *PhotogrammetryKoch2009combining*.

[92]. Realtime stereo systems are emerging [93], but difficulties in matching features continue to influence their robustness.

Optical Interferometry and Moiré Methods

Optical interferometry techniques project a light pattern (e.g., monochromatic, multiple wavelength, white-light) onto a scene and analyze the interference of the reflected patterns with a prescribed reference pattern. From the phase difference that occurs between the reference and the reflected patterns one can infer the depth map with a resolution on the order of nanometers [94, 95]. Digital holography can be used for inspecting large volumes by capturing only one instantaneous 2D intensity image. Methods based on digital holography can achieve an image resolution in the range of micrometers but the reconstruction process of the 3D scene requires seconds for each 3D frame [96, 97]. Furthermore, large scenes require lasers to generate coherent light, which, in turn, generate speckles, and problems with phase ambiguities for surfaces with sharp discontinuities [98, 99]. Moreover, lasers raise safety concerns when human users are present. For overcoming these limitations, one can use optical coherence tomography [100] that results in depth maps of micrometer resolution. However, coherence tomography can only be used for depth ranges on the order of centimeters [101, 98]. Note that optical interferometry methods require high intensity light sources, and highly stable opto-mechanical setups [102].

Moiré techniques illuminate the scene with a periodic light pattern and capture the image as seen through a high-frequency periodic grating whose orientation is prescribed [103, 104]. The geometric information is extracted from analyzing the interference in these patterns, which give accurate descriptions of changes in depth [105]. These methods have an accuracy of up to 10 microns. Ambiguities in measuring adjacent contours is typically resolved by taking multiple moiré images with repositioned gratings.

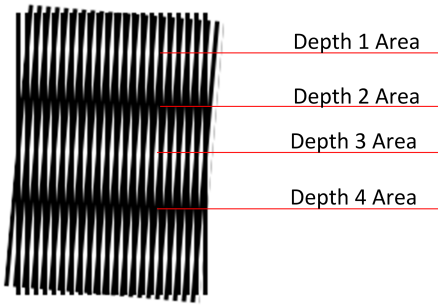

Figure 3.7: Moiré patterns

Fresnel Holograms

A very recent development in the field of 3D imaging is the so called Fresnel hologram that uses diffraction of incoherent light through Fresnel Zone Plates. By projecting concentric ring patterns of incoherent light onto the surfaces of interest [102] one can construct the 3D image of the inspected scene. The depth information of the objects of interest can be extracted from the density of the rings that are projected onto them because points that are closer to the system project less dense rings than distant points. These rings, called Fresnel Zone Plates (FZP) are parameterized with respect to the distance between the imaged surfaces and the projector. Because this holographic method does not rely on interferometry one can build multicolor 3D holograms acquired in real time [106].

Stereo Photogrammetry and Shape From Shading

These methods extract accurate 3D shape information from an imaged scene (one or more images) by using the shading cues detected in the images under controlled illuminating conditions. Recovering the geometric information requires known surface reflectance of the objects in the scene, a constrained reflectance map, or multiple images and light sources [107]. The method uses the intensity variation at each

pixel to estimate the normal of the surface at the surface point that projects to the image pixel. As described in [107] the scene needs to be illuminated from 3 distant light sources that have known, non-coplanar directions. The law that correlates the intensity of the reflected light with the normal vector of the reflecting surface is :

$$c_i(x,y) = l_i^T \cdot \mathbf{n} \int E(\lambda) R(x,y,\lambda) S(\lambda) \cdot d\lambda \tag{3.4}$$

The simbols above represent: $\mathbf{c_i}$: the observed pixel intensity under light source i, $\mathbf{l_i}$: the direction of illumination of light source i, $\mathbf{n}$: the vector normal to the surface,λ : the wave length of the light, $\mathbf{E(\lambda)}$: the spectral distribution of the light source, $\mathbf{R(x,y,\lambda)}$: the reflectance function of the inspected surface, $\mathbf{S(\lambda)}$: the response of the camera for different wavelength light

Let Z (x,y) be the law that describes the distance against the camera of each physical point of the inspected surface imaged at pixel position (x,y). Once the distribution of the surface normal vector is extracted from equation 3.4 the gradient of the range coordinate Z is determined from the following law.

$$n = \frac{1}{\sqrt{1 + |\Delta z|^2}} \begin{pmatrix} \Delta z \\ -1 \end{pmatrix} \tag{3.5}$$

Shape from shading with one light source is ill-posed in general with no unique solution [108]. On the other hand, by illuminating a surface with at least two linearly independent *known* light sources, a unique depth map can be recovered. More recent robust reconstructions from shading cues have been achieved by using multiple images with changing illumination taken from the same view-point, or by using linearly independent colored-light stereo sources whose geometric location and orientation are known [107].

3D Integral Imaging

This technique is based on the principle of integral photography [109, 110] and the 3D image is obtained by processing multiple images taken from coplanar, grid-aligned imaging sensors or lenses. The depth information is generated by analyzing the relative shift of the position of an object in these different images [111]. The reconstruction of the 3D image can be performed by back projecting the rays that have generated the different captured images. These images that are projected back will overlap and will form a sharp image only at the Z distance at which the inspected object is located. The Z value is controlled by the distance between the images representing different viewing perspectives that are back projected. A different approach is proposed by [112] where the range information is calculated only for the central pixels of these images followed by a refinement of the grid formed by these points.

Shape From Focus/Defocus

As the name implies, this class of methods detect points of the scene relative to the focal plane of a camera [113, 114].

For the shape from *focus* methods, the position and orientation of the focal plane relative to the camera are fixed and known, and the sharpest image regions are identified for specific focus settings by applying specific differential functions such as 3D gradient [115] and Laplacian [116]. It is intutive that in order to measure the boundary of an object, we must 'scan' the object with the focal plane. The resulting 3D imaging method can achieve micrometer resolution [117], but the 3D reconstruction process is relatively slow (on the order of minutes for each frame processed [113, 118]).

Shape from *defocus* techniques extract the depth information by taking two images, with two focal distances followed by an analysis of the blur differences in these images. The measurement principle uses the fact that objects located at different distances from the camera are blurred by different amounts. Blurring is typically

modeled through the diffusion equation, and the 3D scene is reconstructed by solving the inverse diffusion problem [119].

Focus/defocus based methods heavily depend on the mechanical adjustment of the focal settings of the system, which severely influences the ability to perform real time 3D image acquisition.

Shape from Texture

The last class of methods that we survey here analyze the deformation of individual texture elements, whose size and shape are constrained, and convert these deformations to a distance map. These methods require an a priori knowledge of specific properties of the textures of the objects, such as: the shape of the texels, the homogeneity[120], isotropy [121], spatial frequency [122], smoothness [123], the planarity and the state of motion [124]. However, these methods are not applicable for objects whose (existing or projected) texture is not regular.

3.2.2 Comparative Analysis of 3D Imaging Alternatives

Despite the tremendous advances in 3D imaging hardware, there are no accepted standards and protocols to measure the performance of these systems. As a consequence, NIST is currently developing a 3D imaging performance evaluation facility along with the protocols for characterizing the performance of various imaging systems [125]. The NIST efforts are focusing on analyzing the effect of several factors on the performance of the systems, namely range, angle-of-incidence, reflectivity, azimuth angle, single vs multiple point measurements, and type of imaged object. We note that in the absence of such a standard, the direct comparison of the depth and image resolution, as well as sensitivity to optical parameters of the 3D imaging systems that are published by the developers or manufacturers can be misleading.

We review here two important aspects that significantly affect the performance of

several key imaging techniques, and present a summary of current published performance indicators for several promising 3D imaging approaches.

Point Matching

All imaging methods that use triangulation require the identification of identical physical points or features across multiple images in order to determine the depth of these physical points. In a broad sense, the point matching algorithms identify regions with similar prominent features that contain the point for which we search across these images. As a consequence, these methods can not be used single handedly for objects with smooth boundaries that do not have distinctive textures or features. A comparative analysis is presented in [126].

Pattern Matching for Structured Light Techniques

Many discrete or continuous coding strategies have been proposed for determining the correspondence between the source of the pattern projected onto the imaged scene and the reflected pattern since this task can have a dramatic implication on the performance of the imaging method [127]. Most strategies use large areas of the captured pattern to be able to compute the depth of a single point. By contrast, the method proposed in [87] uses only the connectivity of adjacent points of a projected grid pattern, but with a lower measurement resolution than is what is presented in [86]. The latter work uses sinusoidal fringe patterns whose density controls the achievable depth resolution. The 3D imaging technique presented in [86] achieves 30FPS (frames per second) for 300k points per frame. According to [128], the standard fringe projection methods cannot measure multiple objects separated in space. Instead, the same work proposes a structured light imaging method based on statistical speckle patterns, that achieves reconstruction speeds of 17.25 FPS. A more detailed presentation of other alternatives is presented in [127].

The Performance of Current Implementations

Structured light techniques produce high-accuracy depth maps, but achieving real-time implementations must avoid sequential scanning of the objects. These techniques require an efficient implementation of the point matching problem, although the sensitivity of this correspondence problem to illumination and geometric parameters influence its robustness. Moreover, structured light techniques can output direct measurements only at the matched points - all others require interpolation or some local approximations, and their efficiency decreases with the increase in accuracy due to the relatively high computational overhead. In principle, the efficiency of these methods is limited only by the available computational power. On the other hand, TOF cameras are monocular, have a relatively dense depth information and constant resolution as well as high frame rates, and do not require interpolation. They have superior robustness to illumination changes, and the phase shift measurement and its mapping to distance values are straightforward and computed in the camera, which minimizes the additional processing required [129]. The depth map output by TOF cameras are largely independent from textures in the scene [130]. The current resolutions achieved by TOF cameras are lower than structured light techniques as described below. The efficiency of the depth map construction by these cameras is physically limited by the photon shot noise and modulation frequency [131, 132].

One of the fastest and most robust TOF cameras [72] offers 40 FPS at a resolution of 200x200 pixels, which is achieved in the camera, without additional computational cost. Higher frame rates of up to 80 FPS can be achieved as the resolution decreases. The typical measurement range for TOF cameras is $[0.3, 7]$ meters, with a precision (repeatability) smaller than 3mm and a depth resolution of about 10 mm. On the other hand, structured light imaging methods based on a speckle light pattern that follows a prescribed statistical distribution appear to be one of the most robust methods in this imaging class, and frame rates of 308 FPS with a measurement accuracy of

50 μm have been documented in [133]. The fast image acquisition method presented in [133] uses two 4630FPS high speed cameras and an acusto-optical laser deflector to generate the statistical speckle pattern and complete the matching problem by analyzing multiple captured images at what seems to be a significant computational cost. Low cost commercial cameras using structured light achieve depth frame rates of up to 30 FPS at a depth image resolution of 320 x 240, a usable range of $[1.2, 3.5]$ meters and a depth measurement resolution lower than the TOF cameras. These parameters, however, are driven by current cost constraints imposed by developers of commercial structured light cameras rather than technological limitations.

Major Limiting Factors Affecting System Performance

Most of the available optical imaging techniques are affected by the specific illumination conditions, and object properties affecting the reflectance. Furthermore, several imaging methods assume the relative position and orientation between the video cameras (VC) and other devices used in the imaging process to be known or computable (column 2 of table 5.14). At the same time, a number of active light imaging techniques require no variations in the illumination conditions (column 3 of table 5.14).

Method	Relative position/orientation constraints	Illumination constraints	Other minimum working conditions	Major limiting factors
Time of Flight			– Scene at least 0.3 m away from the camera [70]	– Resolution limited to about 1cm for objects placed between [0.3,7] meters. – External light with the same characteristics as the active light [79].
Structured light	x		– Location of light source determined for each measured point [86, 84]	– External light with the same characteristics as the active light [134]
Laser triangulation	x		– Imaged surfaces must be scanned. – Laser beam reflected as scattered light [90]. – Beam reflection identified into the captured image [134].	– Objects with sharp edges [90].
Passive Stereo	x		– Require point matching. Hence, smooth surfaces require projection of artificial texture or active illumination from multiple sources. – Measured points must be visible from at least 2 different locations (no occlusion) [90]; imaged object in proximity of the camera (e.g., for passive stereo vision the measurement accuracy decreases according to the quadratic law (3.3) [135];	– Camera occlusions [136]. – Objects must have distinctive features
Optical interferometry	x	x	– Environment factors affecting the light path (e.g., hot air, dense water vapors) [97]; non-interferometric background; ambiguity of reconstruction (solution to "twin-image" problem proposed in [137]).	– Mechanical vibrations – Lasers or high intensity white light sources can induce retinal damage [102]. – Robust pattern coding strategies
Fresnel holograms	x	x	– The wavelength of the projected light must be known [106].	
Moiré patterns		x	– The projected fringes need to be visible in the captured images [105].	– When data gathered from multiple images, the methods are sensitive to motion, blur and step discontinuities in the object boundaries [138].
Photometric stereo and shape from shading	x	x	– The points that are measured need to be illuminated by at least 2 light sources without shadows [107].	– Complex geometry producing shadows [107]. – Existing texture, ambiguity due to non-uniqueness.
Integral imaging	x		– Requires point matching [111, 112].	– When using synthetic aperture, relative location of cameras affect the imaging accuracy [111].
Shape from focus			– Adjustment of discrete focal settings [113]. – Object must have sharp edges that differentiate object from blured background [115].	– Discrete focal settings [113].
Shape from texture			– Texture must be known prior to measurement [123]. – Optical model of the camera must be known [122]. – Visible texture cues for all the reconstructed surfaces.	– Unexpected or non-measurable texture properties – Shadow or light patterns generated by uncontrolled sources.

Table 3.1: Major Limiting Factors Affecting System Performance

Resolution Limiting Factors

All methods presented have the in-plane measurement resolution limited by the resolution and the field of view (FOV) of the cameras. Moreover, all methods using multiple cameras or cameras and sensors that have prescribed positions and orientations in space are sensitive to the accuracy of their spatial location. The computationally intensive 3D reconstruction algorithms can be implemented on GPU cards to achieve at least the performance mentioned in the following table. This will, in turn, make the CPU available for the other processes. Unfortunately, most of the research articles in this field present their speed performance indicators in terms of FPS of the reconstructed 3D scene, or the ambiguous "real-time" attribute. We observe that these parameters heavily depend on specific hardware and software decisions used in each specific case, but these details are not provided in the literature. A summary of the major resolution limiting factors and of current performance indicators is presented in table 3.2.

Method	Resolution limiting factors	Current Resolutions
Time of Flight	Unavailability of low cost high-power IR-LEDs to produce higher modulation frequencies [139, 131, 132]	Highest depth resolution about 1 cm; largest sensor resolution is 200x200 pixels [72]
Structured Light	Global illumination [140]. Also, coding used for projected fringes affects the capability and resolution of these methods. In general, depth resolution is a fraction of the density of the projected fringes [141, 86]. Moreover, simple stripes require phase unwrapping which fails near discontinuities (as is the case of objects separated in space) [128]. Statistical speckle patterns can be used instead.	Depth maps of 50 μm [133] resolution are available at a frame rate of 308FPS.
Laser Triangulation	Resolution with which the laser beam can be shifted, and by the accuracy of the relative location of the camera and laser module. Speed of this imaging method is limited by the (mechanical) speed of scanning.	Highest depth resolution is 25 μm [142]
Passive Stereo Vision and Short Range Photogrammetry	Accuracy of the point matching task (can be improved by active illumination). The measurement accuracy decreases with distance from cameras.	− Micrometer level depth resolutions for distances lower than 50 cm, and mm level resolutions for ranges between 1-10 meters. The accuracy quickly decreases with distance - e.g., equation (3.3)) applies to passive stereo vision [143]. − Depth maps of lower point density and at a lower frame rates than the TOF or the structured light techniques.
Shape from Shading	− Each point must be illuminated by at least 2 sources. Moving, deformable or textured bodies are difficult to handle unless multispectral setups are used [107].	Resolutions can be achieved at sub-millimeter levels.
Moiré Methods	− Pitch of the projected fringes [105] − [105] uses only one image (rather than multiple) which speeds up the computations	−[144] shows a depth measurement resolution of 15 μm while using a commercial LCM projector capable of 1024x768 projection resolution
Optical Interferometry	Coherence and wave length(s) of light source [132]. Predominantly used over small distances up to several cm [145].	The measurement resolution can achieve very high resolution (up to tens of nanometers) for relatively large scenes [97].
Fresnel Holograms	Wavelength of the light used [106].	Depth resolution of 0.5 μm and 2048 x 2048 depth points density are documented in [102, 106];
Shape from Focus	Efficiency ad accuracy of focus adjustment. Relatively complex depth map extraction algorithms.	600μm depth measurement resolution [113]
Integral Imaging	Point matching requires textures or distinctive features; depth resolution limited by image resolution, FOV, number of the sensors used, and number of points that can be matched in different images.	2040 x 2040 depth image at a frame rate of 15 FPS discussed in [112]
Shape from Texture	Properties of the surface texture.	Lower accuracy than other popular methods [122, 120]; simple hardware setup.

Table 3.2: Resolution limiting factors and depth resolutions of current methods.

3.2.3 The Salient Advantages of Different 3D Imaging Techniques

Time of Flight Methods

- Offer one of the lowest computational cost for the acquisition of depth maps with a depth point density up to 40000 per frame and a frame rate up to 80 FPS (at a lower resolution);
- Have low sensitivity to the external light influences as it perceives only infrared light and does process only frequency or amplitude modulated light;
- Can image any object that reflects scattered light;
- Have no mechanical constraints and have a simple and compact setup.

Structured Light Methods

- Offer high resolution for depth measurement and in plane measurement;
- High measurement speed and good measurement reliability for indoor imaging;
- Relatively low resolution cameras are commercially available in simple and compact setup.

Stereo Vision Methods

- The cameras are commercially available with simple and compact setup.
- Offer a depth measurement accuracy on the order of micrometers for close ranges.

Laser Triangulation Methods

- Less sensitive to texture and color changes of the imaged objects as well as to external light variations than all the presented methods except for the TOF cameras.

Photometric Stereo and Shape From Shading

- Can offer a measurement resolution as fine as a fraction of a millimeter for all the 3 measurement directions.

Phase Shift Moire Patterns

- Offers all the advantages of the structured light method based on phase shift [86]. However, as shown in [105], this method ca use a third of the numbers of frames required in [86] for performing the same measurement.

Optical Interferometry

- It is the only method capable of building 3D images at a constant resolution of few micrometers for scenes located in a range of at least 1m [96].
- It can be applied for range measurements in the micro-metric scale [146] as well the geological scale [147]

Fresnel Holograms

- Achieve measurement resolutions similar to the interferometric holograms without having the restriction of using coherent light;
- Can be applied on a on scale ranges that go form a micrometric FOV to a FOV in the range of meters [102];
- Can build multi-color holograms in real time [106].

Shape from focus

- It offers a very compact setup which fact makes it suitable for tight spaces such as endoscopy.

3.3 Gesture-Based HCI: Challenges and Opportunities

A definition of hand gestures can be extrapolated from the usual definition of a gesture given by Meriam Webster dictionary: a gesture is 'the use of motions of the limbs or body as a means of expression'. Consequently, we define hand gestures as the use of hands as a means of expression or providing semantics.

The main goal of this research is to develop hand gesture-based human computer interface for manipulating 3D geometric information within a VR environment without requiring any hardware or gear attached to the user. In other words, the sensing and tracking required for interpreting the user's gestures is performed by natural hand gestures. The system must be insensitive to natural illumination conditions, and be capable of tracking individual finger movements. We assume that the gesture semantics is formed by both static (hand postures) and dynamic gestures. In the latter case, the movement of the hand contributes to the semantics of the gesture.

The discussion from the previous section suggests that there are two 3D imaging approaches that have the performance characteristics and sensitivity to noise as well as the compact setup and range capabilities that match the requirements for a hand gesture-based human computer interface. In this section we discuss the main requirements of such an HCI, present our approach to build natural and versatile hand gesture interface and explore the main challenges and opportunities.

3.3.1 Sensing Technology

TOF and structured light or active illumination stereo imaging methods have competing and somewhat complementing sensing capabilities. Structured light imaging technologies can, in principle, achieve high resolution with a relatively good performance, while TOF systems are fast, have low sensitivity to changes in illumination

conditions and construct the whole depth map with a relatively low computational cost required for measuring the phase shifts. Nevertheless, these capabilities are expected to rapidly evolve given the increased attention in 3D interaction. Both technologies are commercially available in compact sizes and several SDK's exist that can speed up the development of custom applications (see section 3.1.2). The performance difference also suggests that the current structured light cameras can be used for tracking the body and the limbs, while the TOF cameras can be used to track the more rapidly moving fingers.

Our system uses two SDKs for facilitating hand tracking, namely the IISU by SoftKinetic [63] as well as Microsoft's Kinect SDK [64]. The latter offers access to a simplified kinematic model of the human body and provides functions that can be used to query the model for kinematic parameters. Kinect SDK requires that the user's body be visible above the knees in order to build the kinematic model of the human body. On the other hand, the IISU SDK provides similar functions for tracking body parts, but it provides increased functionality. For example, one can define a parameterized motion in the workspace and the SDK identifies the body part that performs the predefined motion. Both SDKs offer multi-user tracking with some occlusion management. While both SDKs can be used to track motion of limbs, neither of them can at the moment represent and track finger gestures.

3.3.2 Results, Challenges and Potential Solutions

Hand segmentation is a critical step for any hand gesture recognition system. Approaches based on processing 2D images need crucial access to 'good' features, due to the richness of variations in shape, motions and textures of hand gestures. On the other hand, three dimensional sensing capabilities provide a direct approach to the segmentation problem. In our system, we track the motion of the user's body by using the structured light camera and monitor 20 joints of the skeletal model using

the Microsoft SDK for Kinect as illustrated in Figure 3.8, including the wrists, elbows, shoulders, head, hip, and knees. Moreover, we track the motion of the wrists to locate the volumes where hands are located and perform robust hand segmentation from the depth map output by the TOF cameras by processing the volumetric and grayscale values as illustrated in Figure 3.8.

The current resolutions of these cameras are insufficient for tracking fingers unless the field of view of the TOF camera is narrowed to the region in which the hands are located. This adjustment can only be done mechanically by adjusting the focal settings of the optics. In order to enlarge the FOV, one could use higher resolution structured light systems discussed in section 3.2; use the additional information captured by the high(er) resolution color camera, or employ additional fixed/mobile cameras.

Hand segmentation is followed by the semantic matching of the static and dynamic gestures, which may rely on robust pose dependent shape signatures. Other common approaches to matching gestures are reviewed in [6, 148, 149]. A comprehensive review of gesture recognition algorithms along with an evaluation of their performance over standard data set can be found in [150, 151]. Surveys focused on hand gesture recognition methods based on vision can be found in [152, 153, 154], while a review of gesture classification methods are reviewed in [155]. Good surveys of gesture interpretation and vocabularies can be found in [2, 156, 157].

It is important to note that defining a vocabulary of hand gestures must depend on the usability of such gestures for natural human computer interaction. There have been several usability studies, mostly on 2D gestures in automotive environments [158, 159], but these results are not directly applicable to virtual object manipulation problem by mans of natural hand gestures. On the other hand, the current usability studies that focus on 3D interfaces [160] have limited scope and applicability given the limitations of the gesture recognition systems on which they are based. This sug-

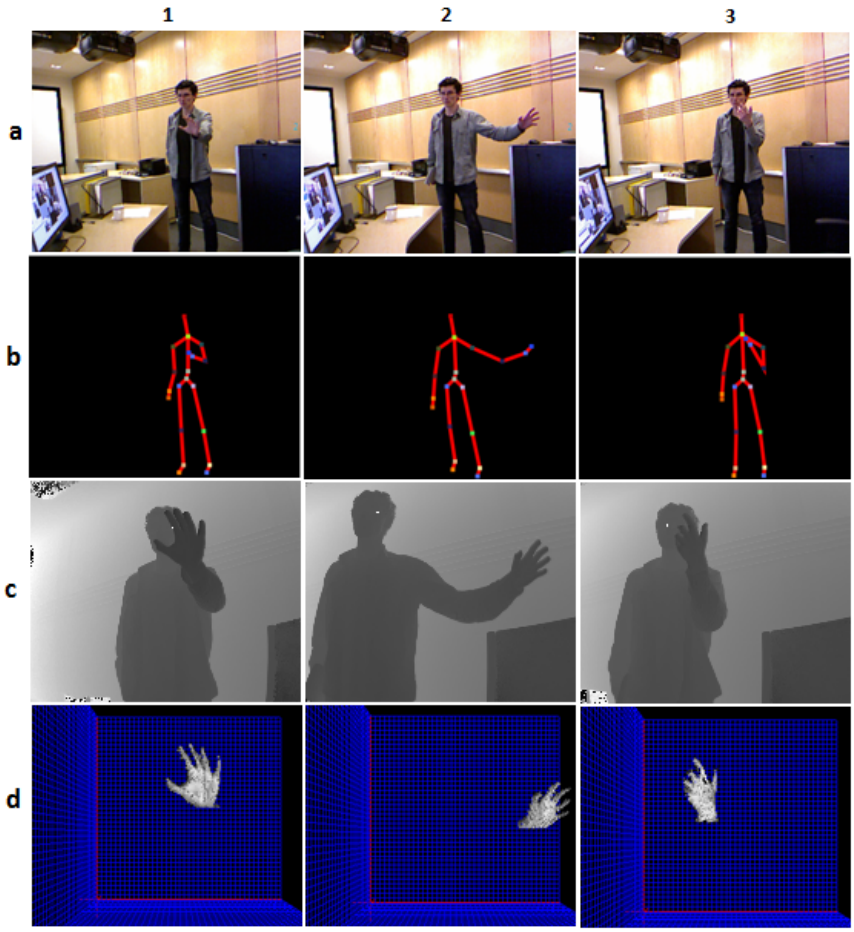

Figure 3.8: Hand segmentation of various hand poses: (a) Color stream data; (b) Skeletal representation; (c) TOF depth map; (d) segmented hand.

gests that the development of practical vocabularies for 3D hand gestures will rely on more specific usability studies that, in turn, depend on the availability of robust hand tracking and recognition technology. In order to mitigate the impact of the tracking faults introduced by the existent body tracking techniques based on 3D imaging, in the next chapters I propose developing virtual object manipulation methods based on manipulative intention inference. Our intention inference techniques are conditioned behavioral cues that characterize general goal directed movement. Driven by intention inference the proposed interaction techniques enable users to grasp, manipulate

and assemble virtual objects using their natural hand gestures in spite of the large variability shown by such gestures.

3.4 Conclusions

The key limitations of existing commercial 3D imaging cameras could be overcome by integrating several depth sensing devices into one imaging system. The difficulties induced by the relatively low resolution of these commercial cameras prove to be worth pursuing, because depth information can reliably produce segmented hand models in cases in which 2D image based methods may fail as illustrated in Figure 3.8(a). Our initial experiments show that practical low cost 3DUIs relying on natural hand gestures can be built by combining the capabilities of commercial structured light imaging hardware with the information provided by commercial time-of-flight cameras. The availability of such a low cost 3DUI can be a game changer in engineering and industrial design and provide new paradigms for the design of software and hardware interfaces, as well as for usability, technical and scientific collaboration, learning, and outreach. It is important to note that the next generation TOF sensors could quickly produce higher resolution, and low cost time of flight cameras whose costs could be comparable with the current cost of Kinect sensor, which would eliminate the need of a hybrid strategy such as the one discussed above.

By preventing wearable hardware attached to the user's hands, we eliminate the possibility of providing haptic feedback to the user. Nevertheless, the myriad of recent smartphone and gaming consumer applications keep proving the fact that users can rapidly adapt to environments that do not exploit the sense of touch for manipulating 3D information. This suggests that the availability of low cost 3DUIs based on hand gestures coupled with the difficulties of current haptic technologies in providing realistic haptic feedback may shift the demand for haptic systems in favor

of more interactive, although haptic-less, interfaces.

Chapter 4

User Adaptable Virtual Object Selection

4.1 General Concepts

The practical necessity for selecting objects that exhibit dimensions smaller than 1 cm becomes apparent when we consider manipulating vertices or edges located in a cluttered environment, or small geometric models of objects like bolts, nuts, chips, etc. Selecting such fine details proves to be surprisingly difficult in the context of the aforementioned hand placement imprecision and tracking uncertainties.

We overcome these issues by inferring the user's intent to select a particular object based on a set of behavioral cues that have been documented in the neuropsychology literature for general goal directed actions. Below we offer a conceptual description of the role played by each of the behavioral cues that we employ.

Using a metric for the efficiency of an action as an indicator for an intentional action is justified by *the principle of rational action* [14, 13]. This principle states that we, as rational beings, devise our actions such that we approach our goal in one of the most efficient manners, considering the constraints of the situation. It is

therefore likely that the object which the user intends to select is among the objects that can be more easily or efficiently selected in the situation at hand.

The work in [15, 16] reveals that the quality of action persistence, or the fact that an action repeatedly ends in a similar state, represents significant evidence of an intentional action. Therefore the persistence shown by the user in approaching a particular object represents an important clue about the object that the user intends to select. Our hypothesis is that using an action persistence metric to infer the selection target will significantly improve the tolerance of our selection method to tracking and hand placement inaccuracies, especially during challenging selection tasks. We make this assumption based on the fact that challenging tasks require persistent and often repeated selection trials. It is known that a general hand reaching movement shows a ballistic phase [18] marked by a Gaussian-like wrist speed profile and a correction phase [29] that corresponds to the oscillating movement of the hand around the target position. The more challenging the selection task becomes for a particular user, the more prominent and longer in time the correction phase becomes. In all such cases, our action persistence behavioral cue will rapidly increase in value, and bias the inference towards the targeted object at an early stage, as described in section 4.3.1. In consequence, the correction phase will be significantly reduced in time. In section 4.4 we test this hypothesis by means of user studies.

It is important to consider that different people have different dexterity skills, and personal preferences regarding the manner in which they select and manipulate objects. In order to adapt to such personal differences, our selection method automatically adjusts the offered level of hand placement fault tolerance according to the subjective needs of the user. We evaluate the user's need for a certain level of fault tolerance by estimating the level of confidence shown by each user about the position in space of his/her hand. Observe that we naturally open our hands when we are uncertain about the position in space of our hands, or when we are preparing to

grasp, and we move our finger tips closer to each other when we are ready to grasp, or when we are confident about the position of our hands. The correlation between the opening of our hand during a general reach to grasp movement and the uncertainty we feel about the placement of our hand is supported by the principle of rational action [14, 13] described above. Namely, when we are uncertain about the position in space of our hands we open them widely in the attempt to increase our chances of reaching the targeted surface and therefore increase the efficiency of our hand reaching action. A similar observation can be made for the case in which we enter a dark room and attempt to explore the space using our hands. In consequence, we can use the opening of the user's hands along with motion cues that represent hesitant or oscillatory movements to estimate the level of confidence shown by each person about the position in space of his/her hand. The oscillations of the user's hands are captured by our action persistence behavioral cue which then controls the mechanism that compensates for hand placement and tracking faults. The same principle of rational action explains the fact that a person with lower hand control is instinctively opening their hand more in the attempt to perform a coarser object selection, or a power grip instead of a precision grip on the objects of interest. Such people also show hesitation or hand oscillations as soon as they face a selection task that they perceive to be difficult. Therefore by using the above behavioral cues, our system is able to estimate the user's subjective need for hand placement fault tolerance and adapt to it, as described in section 4.3.

4.2 System Setup

The hardware setup for our virtual environment involved a 3.95x1.672m Cyviz stereoscopic projective display that offered a rendering resolution of 2480x1050 pixels. While developing our virtual object selection method we used a Kinect camera to

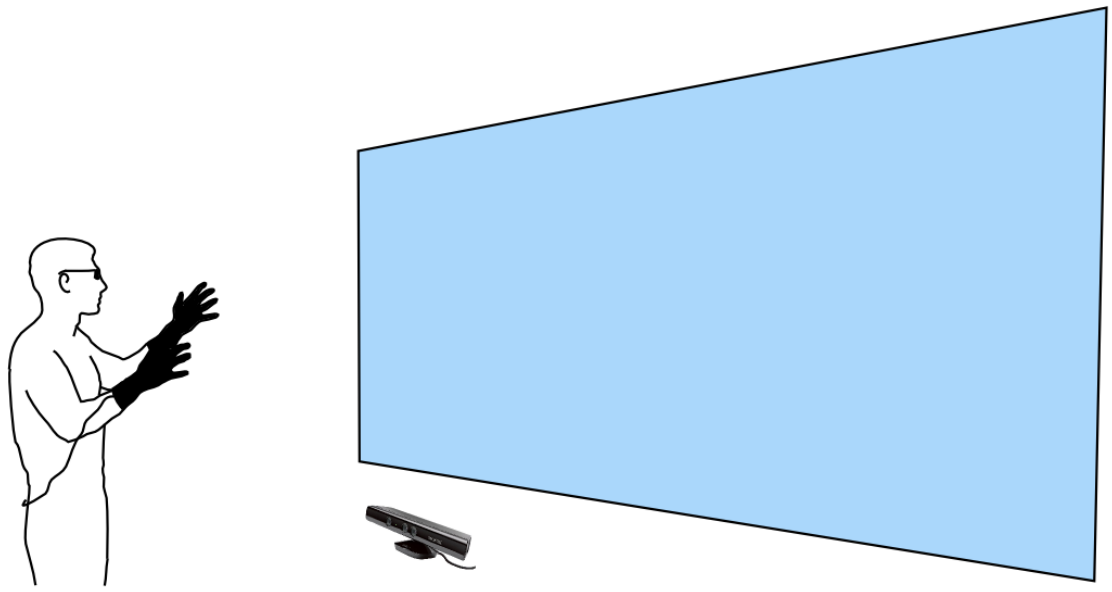

Figure 4.1: System setup.

track the user's hand joints by relying on the algorithm proposed in [161]. Finger motion tracking in 3D space was initially demonstrated by Softkinetic [63] using TOF imaging and in [162] using Kinect cameras and Leap Motion. Although these initial developments appear very promising, there is currently enough room for reliability improvements in their finger tracking capabilities. For this reason, we have developed the incipient stages of our interaction strategies using data gloves (figure 4.1) equipped with flex sensors for reading the flexure of the user's fingers. With this initial prototype our interface exhibits an effective workspace area of approximately 10m^2. Before we feed the acquired tracking data into our intention inference algorithms, we pass it through an acceleration low pass filter to eliminate data indicating tracking faults or unnatural body motion. Once filtered, the tracking data drives a virtual hand model used to simulate the manipulation of virtual objects. We evaluate the collisions between virtual objects and simulate their interactions in a physically plausible manner using the PhysX engine, and perform scene rendering using UDK.

The proper rendering of a virtual environment can significantly improve the user's spatial perception, and further mitigate the challenges induced by the lack of precision in hand positioning. It is important to note that human depth perception does not solely rely on the principles of stereo vision, but also on shading cues [163], motion cues as well as texture [164].

4.3 The Intent Driven Selection (IDS) Method

Our selection method offers hand placement fault tolerance according to the level of confidence shown by its users with respect to the position in space of their hands. Part of this tolerance is achieved by placing a proximity sphere around the simplified hand model of the user such that the fully extended fingers of the hand touch the interior surface of the sphere, as shown in figure 4.2. The proximity sphere is swept along the path described by the motion of the hand, and the objects that are intersected by it are considered to be candidate objects for selection.

The size of the proximity sphere is adjusted according to the users' level of confidence about the position of their hand. As the volume of this sphere corresponding to the fully extended fingers or hand placement uncertainty is considerably larger than the volume of the palm itself, the user can select objects with much lower hand positioning precision. Therefore the IDS method offers a higher degree of tolerance to hand positioning faults when the user shows such hand positioning uncertainty, and therefore needs a higher level of hand positioning tolerance. At the same time, once the users are confident about their hand position, the system offers them a lower hand positioning tolerance and concentrates the selection process on finer details by shrinking the proximity sphere as the hand closes (figure 4.2).

The selection of objects in cluttered environments can be controlled by placing a series of smaller proximity spheres inside of the outer proximity sphere as illustrated

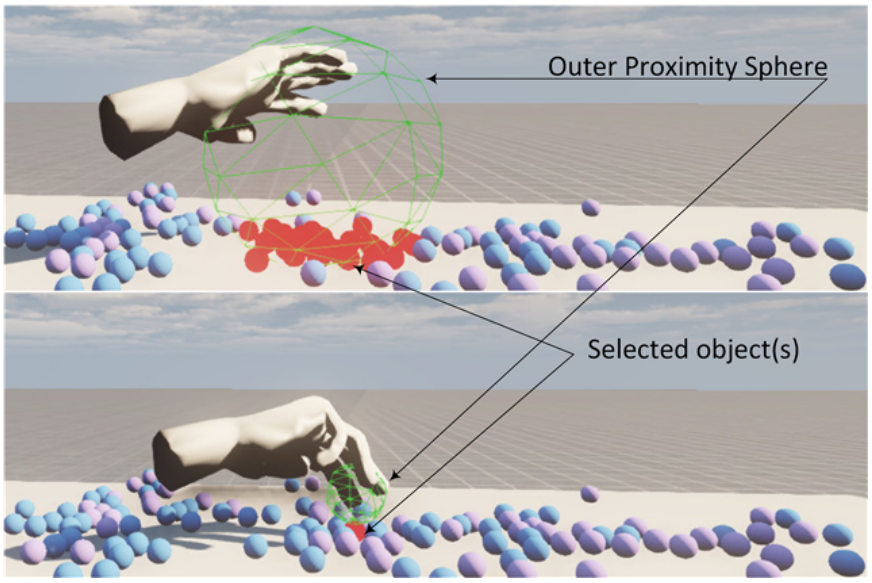

Figure 4.2: The adaptable proximity sphere

in figure 4.3. In this manner, the inner spheres of progressively smaller sizes are intersecting a subset of the objects intersected by the outer sphere as the user's hand approaches the target virtual object, which finally leads to a single object selection. The size of the inner proximity spheres automatically adapts to the user's intention in a similar manner to what was described for the outer proximity sphere. We will refer to this selection method as the IDS^1 method. During the selection process, the proximity spheres are invisible, and the selected objects change their color to red. Our tests show that the smallest objects that can be practically selected with this method are spheres of 4.5 cm diameter.

In order to select finer details, we replace the above mentioned inner proximity spheres with the selection disambiguation mechanism described in the next section. By including the outer proximity sphere the resulting selection method, named IDS^2, incorporates and extends the adaptable hand placement fault tolerance and all other strengths offered by the IDS^1 method. While employing the IDS^2 method, the candidate objects for selection are highlighted using the glowing effect shown in figure 4.4. The object that is ultimately selected is marked red, and a green guiding beam

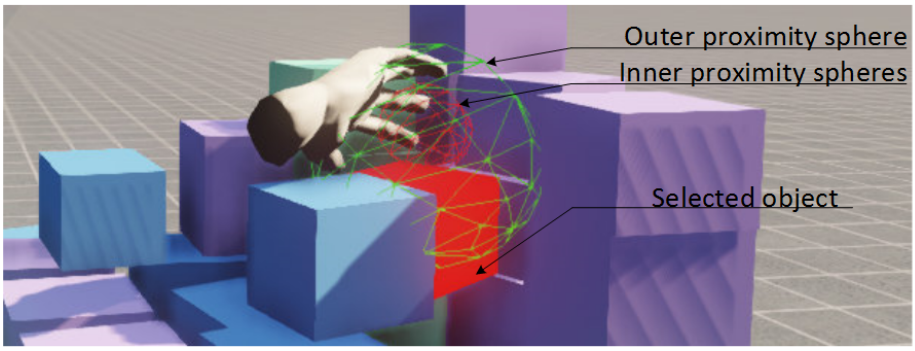

Figure 4.3: Progressive object selection in cluttered environments

joins the center of grasp and the selected object. The beam is used to indicate the direction in which the users need to move in order to approach or depart the hand model from the object currently selected. We will use the term center of grasp to refer to the point located at an approximate distance of 4 cm from the center of the middle finger's middle phalange along the perpendicular to this phalange (figure 4.4). The location of this reference point has been established empirically.

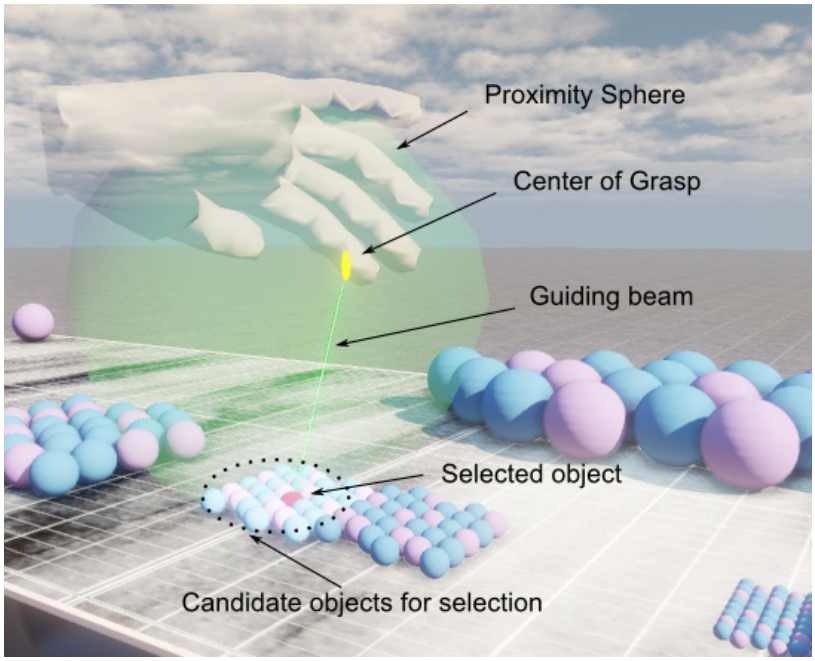

Figure 4.4: The graphical feedback generated during the selection process.

4.3.1 Selection Disambiguation

In order to be able to select in cluttered environments using natural hand gestures, we developed a seamless selection disambiguation mechanism that does not require the user to leave the current environmental context during the selection procedure. The proposed method selects the virtual object for which the following function is maximized:

$$iS(m,e) = t_l \cdot a_p(m,e) + a_{eff}(m,e) \tag{4.1}$$

$$a_{eff}(m,e) = \frac{1}{d_S(m,e)} \tag{4.2}$$

where iS - represents the strength of intent, m - the movement of the user's hand, e - the evaluated object, a_p - the action persistence and a_{eff} - the action efficiency , d_S - the distance to satisfaction, t_l - the tolerance/lock tuning factor.

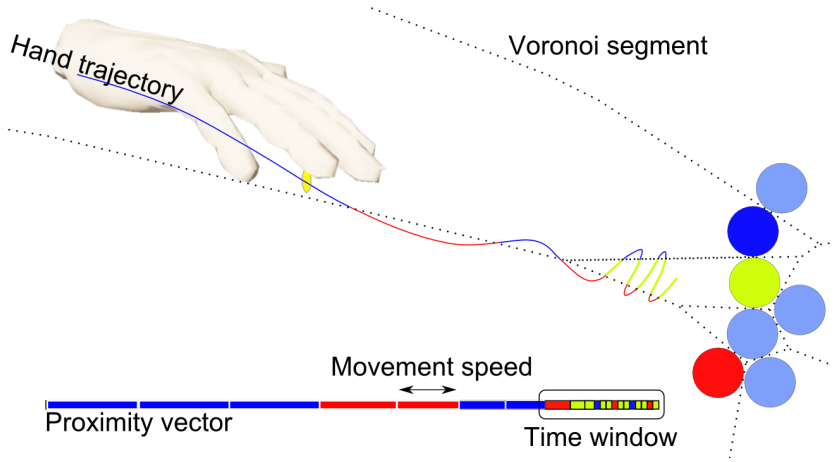

Figure 4.5: Evaluating the action persistence behavioral cue. The identity of each object is marked by its color, and the green disk represents the object that is targeted during selection.

The action persistence parameter captures the number of times in which the center of grasp lies in the Voronoi region of object 'e' during a specific time interval (figure

4.5). In other words, the a_p parameter estimates the number of user's attempts to approach object 'e'. The time window that we have used has a span of approximately 0.7 s while the position of the center of grasp is sampled approximately every 0.033 s. In order to evaluate the action persistence parameter, we use a proximity vector to store the identity of the object whose Voronoi region includes the center of grasp at the moment of sampling (see figure 4.5). The a_p parameter represents the number of times in which the object e was stored in the proximity vector during the past 0.7 seconds. The length of the time window that was used has been established empirically.

The d_S term represents the distance between the center of grasp and the surface of object 'e'. Therefore the action efficiency parameter will assume low values for distant objects that are difficult to select, and high values for objects that are close to the center of grasp. We use basic distance queries to evaluate the d_S parameter, as well as the membership of the center of grasp to Voronoi regions.

In this manner, for those users who show hand jitter, hesitation or lower hand control, the a_p behavioral cue will assume high values with respect to the object around which the user's virtual hand oscillates. As a result, our selection method identifies the target object at an early stage, and tolerates such hand placement or tracking faults. The tolerance is proportional to the value of the persistence behavioral cue as well as the size of the proximity sphere. In the case in which users show higher dexterity levels, our method will select the closest object to the users' hand.

The t_l factor is used to adjust the balance between the hand placement fault tolerance offered by our method, and the selection locking or sticking effect caused by large a_p values. Our experience shows that a good balance is achieved for $t_l = 2$.

4.4 Empirical Evaluation

In all studies described below the users stood approximately 2m in front of the projection screen described in section 4.1. A Kinect camera was placed in front of the screen to track the users movement. The study participants wore on their right hand a data glove equipped with 5 flex sensors that measure the approximate flexion of their fingers. Observe that the technique proposed in this paper is completely independent of the use of data gloves as long as the flexion of the fingers can be estimated. Different virtual environments were rendered on the screen for different tests, as discussed below.

4.4.1 Evaluating the Behavioral Cues

In what follows we test the main effects of the action efficiency and persistence behavioral cues on the performances of the proposed selection method. Specifically, we are interested in finding out the approximate size of the smallest object that can be practically selected when our disambiguation method is based solely on the action efficiency cue. Then we evaluate the hypothesis which states that by relying on the action persistence behavioral cue our method allows users to select their targets faster and more efficiently. Furthermore, we are investigating the influence of the users' number of selection trials using the IDS^2 method, and their previous experience with 3D virtual environments on the speed with which they manage to select their targets.

Test Population

Thirty participants were recruited to take part in this study. Their ages range from 20 to 52, with a median age 26, including 15 female participants and 3 left handed. Twelve have declared that they do not play video games or work with 3D CAD

software packages and virtual environments, seven are sometimes using such 3D environments, and 11 use them frequently. The test lasted approximately 30 minutes and the participants were compensated 10$ for their time.

Procedure

Before taking part in the actual tests, the participants witnessed a brief (less than 20s) demonstration of the capabilities of the interface. Then, they were allowed to experiment by themselves with the elements of the interface for no more than 5 minutes. On the screen the virtual environment shown in figure 4.6 was displayed. The diameters of the spheres assigned as target objects for selection were: target one 4.5 cm, target two 2.25 cm, target three 1.12 cm, target four 0.6 cm.

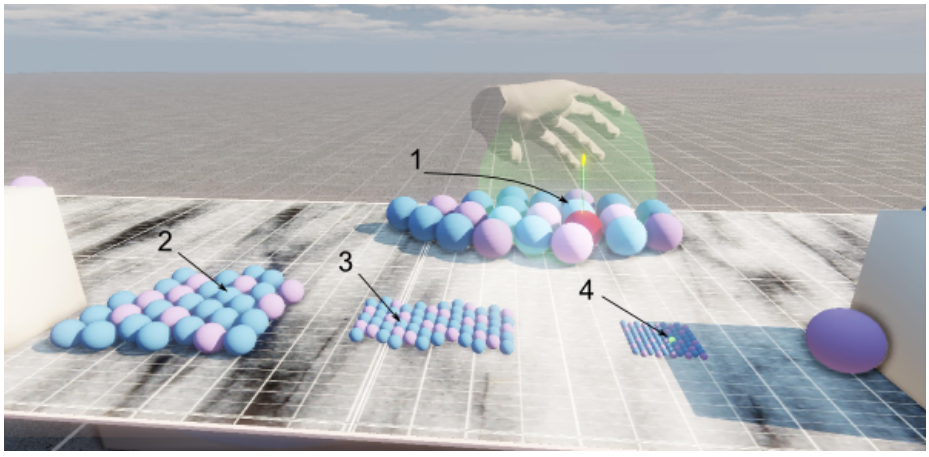

Figure 4.6: The selection test. Target number 4 is marked as the current selection target

In order to evaluate the efficiency of our selection method we considered the following performance parameters: 1) *Time efficiency*, which is the amount of time spent by the user while attempting to select designated objects, and 2) *Perceived effort*, which is the amount of effort spent by the user while performing the selection. To assess the effect of the action persistence behavioral cue on the efficiency of the IDS^2 method, we compared two versions of the IDS^2, namely with and without the

a_p behavior cue. The most efficient selection method is considered to be the one that minimizes both performance parameters identified above.

To evaluate the above parameters, we asked the participants to select as fast as they can the objects marked as targets in figure 4.6. Only one object was designated for selection at a time. Selecting an object different than the designated object triggers a distinctive sound. In order to avoid potential confusion, a different sound is played once a new object is assigned for selection. At the same time, the target object starts blinking bright green (figure 4.6 target 4) until it becomes the subject of a *stable selection*. A selection is considered to be stable if the target object remains selected for a period of 2s, and no other object becomes selected during this period. The selected objects are colored red, while the candidate objects for selection are marked by the glowing effect shown in figure 4.4.

Once the user performs a stable selection, the system assigns a new target object. The time passed between the moment the target object is assigned and the moment the user completes a stable selection is recorded and used for measuring *the time efficiency parameter*. Following the procedure above, the system guides the user through all selection cases shown in figure 4.6. After iterating once through all cases, the user is notified that the selection method is switched to the other selection method. Then, the same procedure is followed while using the other selection method.

In order to minimize the influence of chance on the test outcomes, this process is repeated 10 times for each selection method and each user. As expected, the first selection trials are the slowest for each participant. To avoid biasing the data, the starting selection method is changed with each user. Therefore the tests are counterbalanced, and the 2 selection methods are evaluated in identical conditions. To avoid biasing the user's opinion, during the test we referred to the IDS^2 method that does not use the a_p behavior cue as *'the blue method'* while the other one as *'the green method'*. The color of the guiding beam was changed according to the names

used. At the end of the test each user was asked to evaluate the following statement:

"The green selection method requires less effort than the blue selection method."

☐ strongly agree ☐ agree ☐ neutral ☐ disagree ☐ strongly disagree

The above Likert scale is used to evaluate the *perceived effort parameter*

We evaluate the size of the smallest spheres that can be practically selected when our selection disambiguation procedure relies only on the action efficiency cue by measuring the time spent by users while selecting, and the number of successful selection attempts on spheres of specific sizes.

Result Analysis

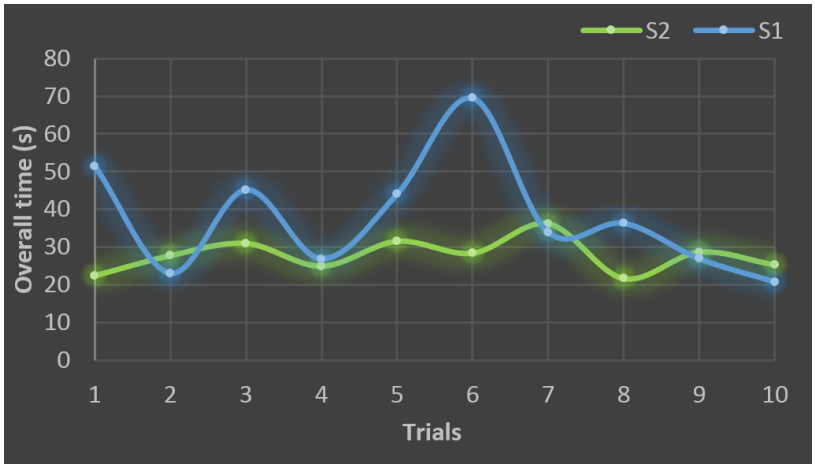

Figure 4.7: The evolution of the overall selection time of a typical user. S2 - the IDS^2 method employing the a_p behavioral cue, S1 - IDS^2 without a_p cue

The data shown in figure 4.7 suggests that the action persistence behavioral cue helps users achieve lower and less variable selection times. In order to obtain time efficiency parameters that are representative for the entire population, we average all data collected for each particular selection case. The results summarized in figure 4.8

indicate that by using the a_p behavioral cue the IDS^2 method becomes 5.4 % slower on target T1 (R = 2.25 cm) , 5.1 % faster on T2 (R = 1.12 cm) , 30.8 % faster on T3 (R = 0.56 cm) , 105.6 % faster on T4 (R = 0.28 cm) respectively 45.5 % faster overall.

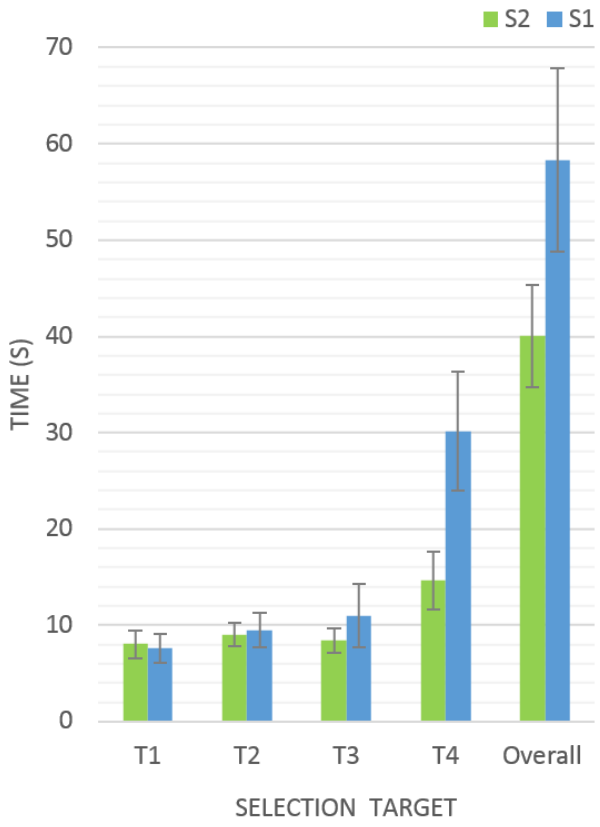

Figure 4.8: The selection time parameters averaged over the entire test population. S2 - the IDS^2 method employing the a_p behavioral cue, S1 - IDS^2 without a_p cue. The error bars represent the standard deviation of the average performance of each user.

We run repeated measures one way ANOVA tests to verify if the collected data provides significant evidences to support the above observations. The variances of the data collected for the 2 methods are stabilized by applying a natural logarithm transformation on the timing data.

The results show that, when augmented by the a_p behavioral cue, the IDS^2 method becomes significantly faster in terms of the time spent to select all 4 targets ($F_{1,29} = 83.7, p < 0.001, \eta^2 = 0.74$), as well as to select target 4 ($F_{1,29} = 129.9, p < 0.001, \eta^2 = 0.81$), and target 3 ($F_{1,29} = 13.6, p < 0.035, \eta^2 = 0.32$). On the other hand, the data does not show a significant difference between the methods when tested on target 2 ($F_{1,29} = 0.004, p > 0.1, \eta^2 = 0$) or target 1 ($F_{1,29} = 1.61, p > 0.1, \eta^2 = 0.05$). Also, as shown in figure 4.9, 23.3 % of the participants strongly agree that the 'green' selection method requires less effort than the 'blue' one, while 46.6 % agree, 13.3 % are neutral, and 16.6 % disagree. In consequence, we can conclude that by employing the action persistence behavioral cue our selection method allows users to select their targets faster and more efficiently, especially during difficult selection cases.

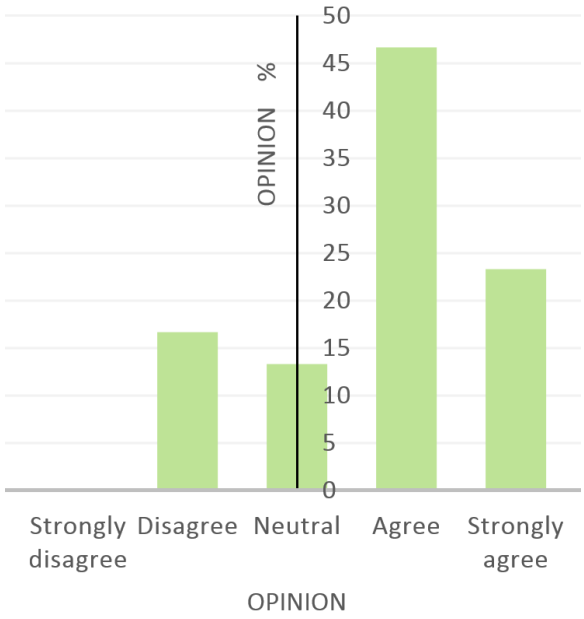

Figure 4.9: The user's perception of the selection effort reduction caused by involving the action persistence cue into the selection method.

Out of the 300 selection trials that have been performed on target 4, 1% ended in abandon when the 'blue' selection method was used. We used a time measurement equal to 3 minutes for every task abandon. The mean and the standard deviation

of the time spent by users during selection can be seen in figure 4.8. No abandon was encountered during the rest of the 2100 selection trials. Based on this data, we conclude that the smallest sphere that can be repeatedly selected while relying on the a_{eff} cue alone has a diameter of 0.6 cm. Furthermore during all 1200 selection trials no task abandon was encountered while using both behavioral cues in our selection disambiguation procedure.

Next, we look at the influence of users' experience with 3D virtual environments on their ability to perform quick selection tasks using the IDS^2 method. The tests that follow are performed with the complete IDS^2 method containing all behavioral cues presented in section 3. On the collected timing data we run a series of repeated measures ANOVA tests in which the declared user experience is treated as a 3 level factor. The results show that the users previous experience with 3D virtual environments does not significantly affect their performance in any of the 4 selection cases: T4 ($F_{2,27} = 2.42, p > 0.1, \eta^2 = 0.15$), T3 ($F_{2,27} = 1.08, p > 0.1, \eta^2 = 0.07$), T2 ($F_{2,27} = 1.75, p > 0.1, \eta^2 = 0.11$) and T1 ($F_{2,27} = 2.05, p > 0.1, \eta^2 = 0.13$).

Similarly, we test the influence of the number of selection trials on the users selection speed. This factor proves to be significant while tested on T4 ($F_{9,258} = 2.16, p < 0.05, \eta^2 = 0.07$), T3 ($F_{9,258} = 3.22, p < 0.001, \eta^2 = 0.1$), T2 ($F_{9,258} = 3.34, p < 0.001, \eta^2 = 0.1$) and T1 ($F_{9,258} = 7.85, p < 0.001, \eta^2 = 0.21$). Due to the fact that most of our users took a significantly longer time during their first selection trial than during the remaining 9 trials, we run the same ANOVA test without considering their first trial on each of the 4 targets. Interestingly, in this case the number of selection trials becomes insignificant when tested on T4 ($F_{8,229} = 1.04, p > 0.05, \eta^2 = 0.03$), as well as on T3 ($F_{8,229} = 1.71, p > 0.05, \eta^2 = 0.05$), and T2 ($F_{8,229} = 1.18, p > 0.05, \eta^2 = 0.03$), but not on T1 ($F_{8,229} = 6.02, p < 0.001, \eta^2 = 0.17$). These results show that after the first selection trial on each target the users stop learning how to use the IDS^2 method except in the case of target T1. This

surprising exception could be explained by the fact that the tests on both selection methods start with T1. Therefore the users have at least the experience of selecting one T1 target before they attempt to select the other targets. The difference in the results obtained while considering the first selection trial, and the ones obtained while neglecting the first trial indicates that the IDS^2 method requires very little experience or training.

4.4.2 The Performance of The IDS^1 Selection Method

Here we briefly perform a direct comparison between the efficiency shown by our IDS^1 method and the Virtual Hand Selection (VHS) method. As explained in section 3.3, the IDS^2 method incorporates and extends the strengths of the IDS^1 method and therefore the results obtained in this test represents an approximate lower bound of the capabilities of the IDS^2 method as well.

The test procedure is identical with what was presented in the previous test, except for the following aspects: eight volunteers took part in this study, including two female participants, and one left handed. Their ages range from 20 to 27, with a median age of 25. None of the participants had previous experience with manipulating virtual objects in 3D using natural gestures.

In order to evaluate the two performance parameters described in section 4.1.2, our participates were asked to select the targets shown in figure 4.10 five times with each selection method. The size of the virtual blocks (i.e., the *large* objects that we refer to below) used in our selection tests were $9 \times 10 \times 12$cm and the virtual spheres (the *small* objects) had a radius of 2.25cm. Each of the selection cases shown in figure 4.10 is designed to raise different selection challenges as we explain below:

- Case a): *selecting a large and mostly occluded object.* This task evaluates the selection efficiency in one of the common cases when users show a significantly increased hand placement imprecision. In such a case, the evaluation results

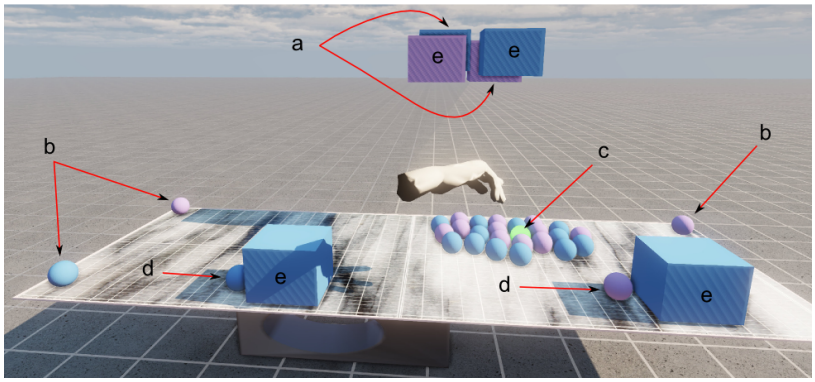

Figure 4.10: The selection cases: Selecting a) Large Occluded Objects, b) Under Tracking Noise Conditions, c) Small Objects in Cluttered Environment, d) Small Objects Close to Large Objects, e) Large Unoccluded Objects

will be dominated by the effects of *hand placement imprecision*. The motivation for this selection setup comes from the fact that users can observe only a small part of the target object and, therefore, they do not precisely know where the object boundaries are. Furthermore, because the target objects are occluded by large objects, the users' virtual hand will become partially occluded when the users approach their target. In consequence, participants will have difficulties in understanding the relative position between their hand and the target object, which decreases their ability to precisely position their hand model in the virtual space.

- Case b): *selecting a small object under tracking noise conditions*. As figure 4.10 shows, the target objects for this case are placed close to the lateral sides of the table model. The table top is positioned such that its side edges are in close proximity to the limits of the field of view (FOV) of the tracking camera. When the users try to get near these limits, parts of their body might leave the FOV of the camera or might get occluded by other body parts. In consequence, the image processing and tracking algorithms cannot collect sufficient information about the position of users body parts in order to produce reliable output. This

fact translates into an increased frequency of tracking noise occurrence.

- Case c) *selecting a small object in a cluttered environment.*
- Case d) *selecting a small object positioned in close vicinity to large objects.*
- Case e) *selecting large and unoccluded objects.*

Result Analysis

During the test each participant performed with each selection method 5 selection trials on each of the 12 target objects marked in figure 4.10 to produce a total of 60 measurements for each participant and selection method. The timing data summarized in figure 4.11 suggests that on average the IDS^1 helps users select 76% faster in case a) in which users show hand placement imprecision, 26% faster while they are selecting under increased tracking noise conditions (case b), 15% slower when selecting small and cluttered targets (case c), 9% slower when selecting small objects that are in close proximity of large objects (case d) and 17% slower when selecting large and unoccluded objects (case e).

We use again one way repeated measures ANOVA tests to analyze the significance of the observations made above, as described in section 4.1.3. The results show that by using the IDS^1 method users select their targets significantly faster when facing the selection case a) ($F_{1,7} = 6.5, p < 0.05, \eta^2 = 0.48$) as well as in case b) ($F_{1,7} = 13.1, p < 0.01, \eta^2 = 0.65$). On the other hand the data does not show a significant difference between the methods in selection case c) ($F_{1,7} = 3, p > 0.1, \eta^2 = 0.3$) or d) ($F_{1,7} = 0.67, p > 0.1, \eta^2 = 0.08$). Surprisingly, the IDS^1 method turned to be 17% slower ($F_{1,7} = 26.1, p < 0.01, \eta^2 = 0.78$) than the VHS method in the case in which the users were asked to select large objects (case e) that do not require accurate hand placement or significant tolerance to tracking noise. This might be explained by the fact that while using the VHS method an object is selected once the

hand model intersects an object. Because this selection case requires a lower level of control over the hand placement, and the user can easily see where the intersection takes place, the VHS method proves to be facile and fast. At the same time, unlike the IDS^2 method, the IDS^1 method does not show the extent of the proximity spheres. Therefore, without previous experience, the users cannot immediately tell where exactly is the intersection taking place, as is the case with the VHS or IDS^2 methods.

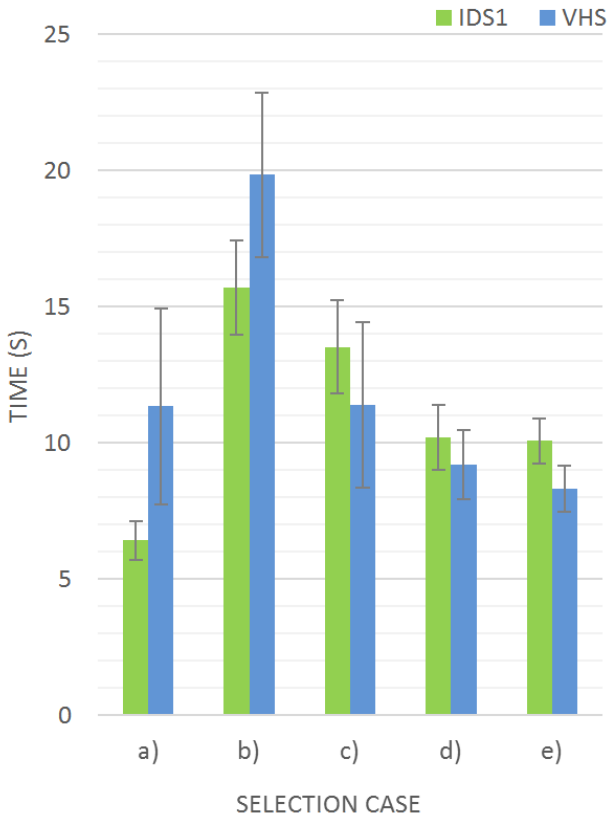

Figure 4.11: The selection time parameters averaged over the entire test population. The error bars represent the standard deviation of the average performance of each user.

While evaluating the perceived effort parameter 87.5% of the participants agreed that the IDS^1 method requires less effort for selection than the VHS method while

12.5% disagreed. These results indicate that the IDS^1 selection method allows users to select their targets faster and more efficiently than the VHS method, especially during challenging selection cases.

4.5 Conclusions

In this chapter we introduced a new virtual object selection method that facilitates the use of natural hand gestures to manipulate virtual objects in 3D. Our method does not rely on hand held devices or symbolic gestures and therefore, it does not restrict the manipulative capabilities of our natural hand gestures. Instead, this technique supports the use of 3D imaging methods for tracking the user's body, and compensates for the inherent tracking and hand placement faults.

When compared with the existent selection methods, our approach affords the use of natural hand gestures to select objects whose dimensions are smaller than the tracking resolution of the employed system. The proposed technique offers a seamless selection disambiguation mechanism, which does not require the user to leave the current manipulation context or use symbolic gestures and buttons.

We achieve these capabilities by identifying the objects that are targeted during the selection process based on a set of behavioral cues which have been documented into the neuropsychology literature. By means of user studies we have tested the relevance of 2 behavior cues with respect to the virtual object selection task. The results prove that the action persistence cue enables users to select objects 45% faster and more efficiently, especially during challenging selection tasks. At the same time the action efficiency behavior cue affords the selection of objects having their largest dimension as small as 0.6 cm even when these objects are located in environments in which the distance to neighboring objects is approximately 0.1 cm.

Furthermore, these behavioral cues enable us to estimate the user's need for hand

placement fault tolerance during the selection process. In consequence, our method is capable of automatically adapting to the user's subjective need for various levels of hand placement and tracking fault tolerance.

Chapter 5

Manipulating and Assembling Virtual Objects

5.1 General Concepts

An effective 3DUI that affords the use of hand gestures to manipulate virtual objects without employing wearable hardware can be developed by:

1. Compensating for the
    - Loss in the hand placement precision [8, 9, 7];
    - Gesture variability among different users and execution contexts;
    - Inherent tracking uncertainties and information delay;
2. Detecting the user's manipulation intention from natural hand gestures, and executing virtual object manipulation in real time.

We overcome these issues by extracting behavioral cues corresponding to manipulative gestures from the observed continuous natural hand gestures of the user. The

motivation behind our approach can be illustrated through the analogy with the human ability to understand someone's speech even when he is grammatically incorrect or accompanied by background noise. A similar observation can be made about our ability to recognize common human actions in the presence of incomplete information. For example, consider a person standing in front of a door with one hand near the door's key hole while the reaching arm shows subtle movements. To an outside observer such a situation suggests that it is very likely that the person standing by the door is trying to unlock it, and this is true although the key and the motion of the fingers are not seen by the observer. Similarly, for our context, we conjecture that it is possible to understand someone's manipulative intention even if his/her hands are inaccurately placed relative to the virtual object that is being manipulated or if our scene observations are temporarily affected by noise or uncertainties. In fact, the door manipulation example suggests that we can approach the task of intention inference by analyzing the activity contexts and the motion of large or salient body parts. The same idea is applied for reach to grasp gestures, but unlike the previous example, these gestures show high variability among different types of grasps, among gestures representing the same type of grasp while performed by different individuals, or even the same type of gestures performed by the same person but in different manipulation contexts [165, 166]. Analogous examples can be found for most of our natural gestures.

We argue that one can account for this large variability and build methods that are generalizable by relying on descriptive features that are not significantly influenced by tracking uncertainties, hand placement faults or the common variability of manipulative gestures. One such set of features can be found in the neuropsychology literature that describes the motion profiles shown by wrists and fingers during manipulative actions. By detecting those behavioral cues that are encountered in all

generic[1] human manipulative gestures we are able to infer the manipulative intentions of the user in the presence of gesture variability and uncertainties.

5.1.1 Behavioral Cues for Reach to Grasp Gestures

The studies summarized in [21] show that during a general reach to grasp movement the human hand shows early finger modulation that will morph into a hand grip posture which will show a maximum grip aperture in the time interval between 60% and 70% of the total reach to grasp movement time. This grip aperture was found to be linearly dependent on the object size. Furthermore during a general hand reaching movement the speed of the hand wrist shows a Gaussian like, or bell shaped profile [29, 21, 17, 18, 19, 20] as illustrated in figure 5.1. According to [17] the shape of the motion profiles can be observed at all movement speeds. Yet, due to the limitations of our body tracking technology we will not be able to observe high speed Gaussian profiles. However, the descending Gaussian branch shown in figure 5.1 corresponds to the movement phase in which users slow down their hands in order to reach the desired position. Since careful or meaningful movements usually characterize the object assembly procedures, we are interested in identifying motion cues that represent users' *intent to accurately position their reaching hand*. As you can see in figure 5.1, the end of the descending Gaussian profile represents such a behavioral cue. Namely, this cue is preceding the accurate placement of user's palm around the object to be grasped, respectively the careful placement of the grasped object. Such motion features are also present in the case in which the user attempts to employ general grasping gestures for manipulating virtual objects in a contact free virtual environment [165]. It is worth noting that if the users are required to position their hands with high precision in the above mentioned environments, the hand velocities will also exhibit a positioning correction phase [29] in addition to the

[1]By *generic* manipulative gestures we refer to all forms/instances of specific manipulative gestures, such as hand reaching or grasping, which commonly occur during physical manipulation.

large Gaussian profiles. During this phase the velocity profiles are less ample and regular than the Gaussian profiles. The length in time of the positioning correction phase can be interpreted as an indicator for the level of difficulties caused by the loss in the users' hand positioning precision for a certain manipulation task. As we will later show our 3DUI does not require the users to position their hands with high precision and therefore the correction phase in the hand reaching movement becomes negligible in our case.

Given the above gesture characteristics that remain invariant in all observed grasping gestures, we use them to build salient feature vectors that are representative for general forms of grasping gestures. Consequently, we are able to overcome the challenges posed by typical grasping variability by extracting behavioral cues from the continuous tracking data and classifying them into groups that represent grasping or other manipulative intentions.

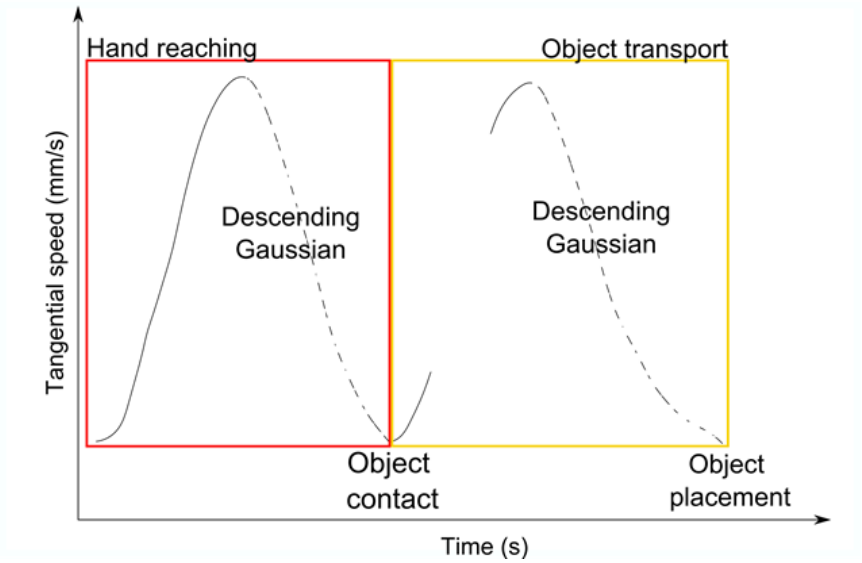

Figure 5.1: The bell-shaped profiles shown by the tangential speed of the wrist during reaching movements.

5.1.2 Human Action Segmentation

In order to develop efficient classifiers we need to *segment the continuous stream* of tracking data into time windows that represent a single action [148, 167].We accomplish this by using the behavioral cue shown during a general arm reaching movement as the segmentation feature that marks the time boundaries of a certain action. For reasons explained above we use the end of the descending Gaussian profile shown by the tangential speed of the wrist of the reaching arm as our segmentation feature. Given the fact that the arm reaching movement is a primitive motion present in all manipulative gestures that require a *precise hand positioning*, the proposed segmentation feature is generally applicable to all these types of gestures, including those involved in assembly tasks.

Considering that during an object assembly task the user's hand reaches the object to be manipulated, performs the manipulation task, and then reaches again towards the position and orientation where the object must be placed, the arm reaching movement appears to be a natural action segmentation feature for an assembly task. Note that a large part of our daily interactions involve arm reaching movements, so this segmentation feature is applicable to a wide range of human activities. Similar principles are used for other manipulative gestures as shown below.

5.1.3 Managing Tracking Uncertainties

In order to build intention inference methods that tolerate *tracking uncertainties* we need to identify those behavioral cues that have a low likelihood of being significantly affected by such uncertainties. Based on the time duration of these uncertainties, they can be classified as instantaneous or persistent tracking uncertainties. In the instantaneous class we group the tracking faults that have a time persistence of less than 60ms such as those caused by imaging noise, transient external light influences, specularities or short time occlusions. On the other hand, the group of persistent

uncertainties includes tracking faults caused by persistent occlusions, low imaging resolution and others. The time threshold that separates these 2 classes has been established experimentally as its exact value depends on the performance of specific body tracking methods.

One group of behavioral cues that are not sensitive to instantaneous uncertainties are those defined by tracking information having a long time span. However, the long time span will obviously force the processing methods into a long time wait before they can start processing the data stream. In order to limit this delay to values below 200ms, we are incorporating in our behavioral cues large time span tracking information by defining the cues in specific activity contexts such as the correlation between the finger modulation, and the hand reaching movement. Despite this relatively large time span that accounts for the time needed for the fingers to modulate plus the time spent during the hand reaching phase, these behavioral cues can be identified without a noticeable time delay due to the overlapping character of these actions. By detecting these specific activity contexts we can identify the tracking information that does not match the context, and treat this information as noise.

We handle persistent tracking uncertainties by using behavioral cues that include motion features of salient or large body parts, which are less likely to be affected by persistent tracking uncertainties. Namely, the motions of homologous finger joints of the same hand are highly correlated during the reach to grasp movement [168]. This fact helps improve the finger tracking robustness to occlusion or noise. As an example, if we are observing a Gaussian like speed profile for the forearm of the arm, and three of the fingers are showing the modulation in time that would describe a grasping intention, then we can still infer the grasping intention of the user even if the other two fingers are completely occluded or if the stochastic tracking algorithm indicates that they are moving in an unusual manner.

5.1.4 The Inherent Hand Positioning Imprecision and The Virtual Object Assembly Endeavor

In order to resolve the problems induced by *the imprecision in the hand/finger positioning*, our intention inference methods rely solely on behavioral cues that describe the hand/finger motion of the user relative to a previous hand/finger position. In other words, our intention inference methods do not depend on the relative position between the virtual model of the user's hand and the objects that are to be manipulated, therefore, we can robustly detect the user's intention despite the hand positioning imprecision problem. We identify the object which the user intends to select by relying on the intent driven selection (IDS) technique proposed in chapter 4. In consequence, our manipulation techniques benefit form the hand placement and body tracking fault tolerance embedded in the IDS method.

To infer the fashion in which our users intend to assemble virtual objects we will apply and extend the intention inference principles introduced in section 4.1. Namely we are decomposing the virtual object assembly task into a set of constraint selection actions that occur in parallel or in sequence. The constraint selection process happens seamlessly while our users employ their natural hand gestures to assemble virtual objects. To achieve this, out of the acquired tracking data we extract a set of behavioral cues that are representative for general goal directed actions. The cues are used to model users' behavior and infer their manipulation intentions by means of machine learning. Below we offer a conceptual description of the role played by each of the behavioral cues that we employ.

Using a metric for the efficiency of an action as an indicator for an intentional action is justified by *the principle of rational action* [14, 13]. This principle states that we, as rational beings, devise our actions such that we approach our goal in one of the most efficient manners, considering the constraints of the situation. It is therefore likely that the assembly constraint which the user intends to apply is among

those constraints that can be more easily enforced in the situation at hand.

The work in [15, 16] reveals that the quality of action persistence or the fact that an action repeatedly ends in a similar state represents significant evidence of an intentional action. Therefore, the persistence shown by the user in moving the manipulated object towards the location where a particular assembly constraint is satisfied represents an important clue about the constraint which the user intends to apply. Our hypothesis is that by using an action persistence metric to infer the targeted constraint we will significantly improve the tolerance of our assembling intention inference method to tracking and hand placement inaccuracies. More specifically, this cue helps us discriminate between accidental movements, body tracking faults and users' persistent trials to apply the set of assembly constraints they desire.

The action duration or the effort invested by our users in a specific action represents another indicator of the intentional character of the action [16]. The correlation between these behavioral cues and the intent of an action has been studied in the neuropsychology literature. However, the exact relationship between these behavioral cues and the strength of intent of the actor has not been studied before. In order to establish the relative importance between these cues and infer users' intentions, we employ a CRF probabilistic graphical model as described in section 5.6.2.

5.2 System Setup

The current hardware setup for our virtual environment consists of a stereoscopic projective display that offers a rendering resolution of 2480x1050 pixels. A LeapMotion camera is place in front of the screen to track a user's hand and fingers. Before we feed the acquired tracking data into our intention inference algorithms, we pass it through an acceleration low pass filter to eliminate data indicating tracking faults or unnatural body motion. The filter has an empirically established cut off value of 5.4

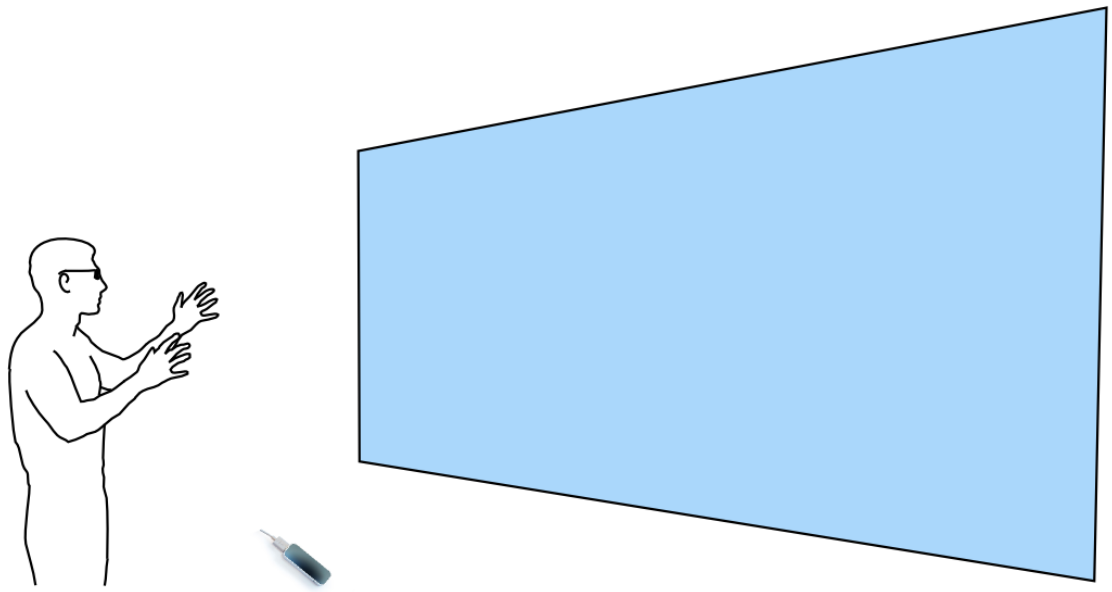

Figure 5.2: System setup

mm/s^2. Once filtered, the tracking data drives a virtual hand model used to simulate the manipulation of virtual objects. We evaluate the collisions between virtual objects and simulate their interactions in a physically plausible manner using the PhysX engine and perform scene rendering using UE4. The proper rendering of a virtual environment can significantly improve the user's spatial perception and further mitigate the challenges induced by the lack of precision in hand positioning. It is important to note that human depth perception does not solely rely on the principles of stereo vision, but also on shading cues [163], motion cues as well as texture [164].

5.3 Action Segmentation

Once an object is selected, we need to infer users' manipulation intentions in real time. The action segmentation strategy discussed in the General Concepts section plays a critical role in enabling the efficient classification of continuous hand motion

data into clusters that represent specific manipulative intentions.

We propose using the end of the descending branch of the Gaussian profile shown by the wrist's speed as the feature that marks the end of a manipulation activity. As explained in section 5.1.1 this motion feature is present in general arm reaching movements. In order to detect the motion feature, we store the speed of each wrist into a ring buffer, and then determine the variation of these velocities by differentiating the preprocessed tracking data. On top of the low pass filter mentioned in section 5.2 we apply an acceleration median filter defined over 5 data frames. To reduce the influence of hand trembling, or oscillations that might be caused by hand placement faults, we smooth the filtered data using 4 data frames. Therefore, the detection of the end of a descending Gaussian profile reduces to identifying a smooth descent in the speed profile followed by a relative hand stop. The velocity analysis is triggered once a virtual object is selected and the relative velocity between the hand and the object is lower than an empirically established threshold of 1 mm/s. All the design parameters that are empirically established are strongly dependent on the performance of the used body tracking method.

5.4 Object Grasping and Basic Manipulation

To detect users' grasping intention, our algorithm starts by searching for the end of the descending Gaussian profile. When the finger flexion pattern [21] is identified following the descending Gaussian profile, a grasp gesture is detected and the outer surface of the selected object is attached to the inner arch of the reaching palm. The implementation details can be seen in the process flow diagram sketched in figure 5.3.The IDS object selection method equips our grasping technique with a significant tolerance to hand positioning faults, while mitigating the effects of the tracking uncertainties as explained earlier.

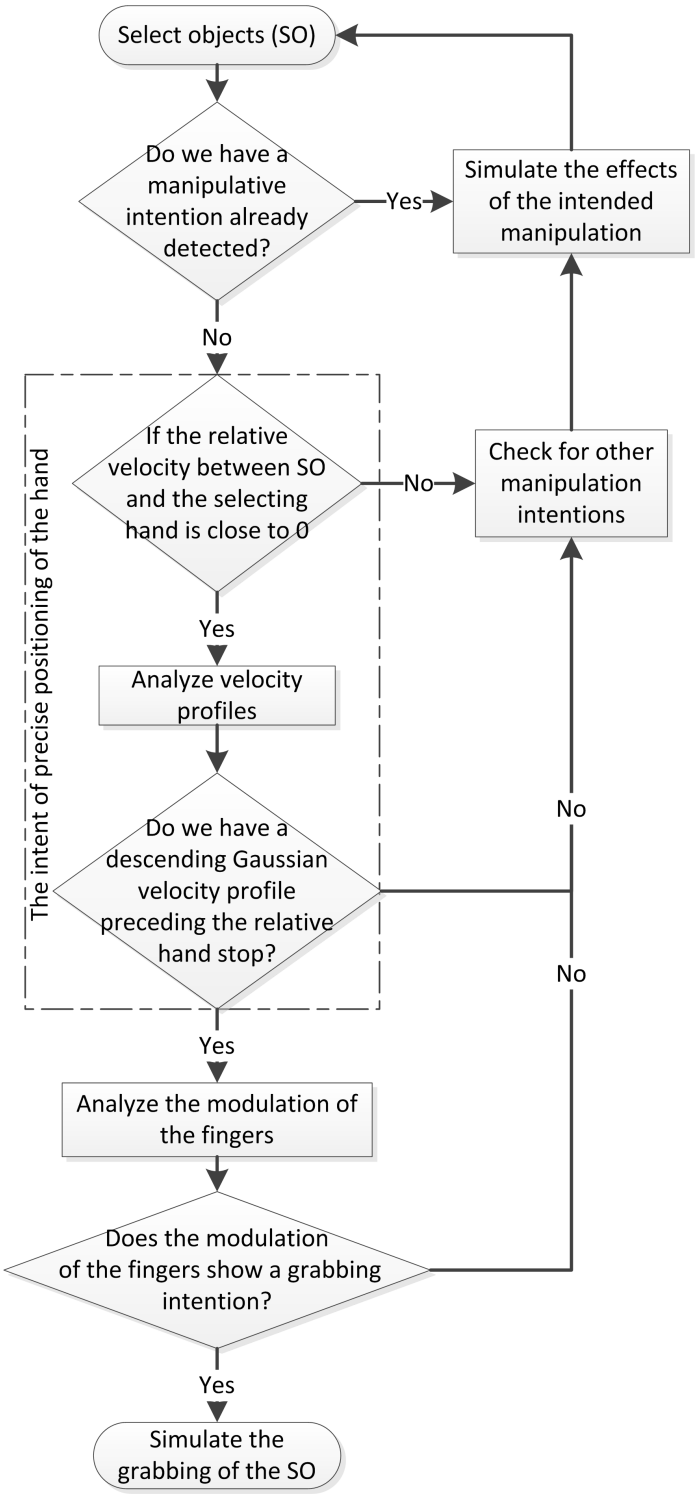

Figure 5.3: The process flow diagram of the grasping method

Once grasped, the selected object naturally follows the translation and the rotation of the virtual hand model. Finally, the grasp release is triggered when the sum of the angle openings in the joints of the grasping fingers is increased more than 10 degrees. After an object is released its dynamics are simulated in a physically plausible manner using the PhysX engine.

5.5 Guiding Push and Object Hitting Simulation

The guiding push of virtual objects is achieved by evaluating the collisions between the simplified hand model and the virtual environment. Specifically, if a collision with an object is detected and the hand continues to move with accelerations smaller than a prescribed threshold (1.7 mm/s^2), the system will infer a pushing intention and, in consequence, will attach the pushed object to the virtual hand model by mating the points of collision between the two bodies. So the pushed object will follow the motion of the hand. Once the action end feature is identified (see the General Concepts section), the object-hand attachment will be terminated and the pushing effect will end. In the following text we will use the term Intent Driven Push (IDP) to refer to this method.

It is apparent that our approach does not simulate the pushing effect as realistically as the physics dominant methods. However, by relying exclusively on physical simulations, any manipulation task in the presence of hand positioning and tracking uncertainties is bound to suffer from unintended instabilities of the physical simulations, which often make such a system unusable. On the other hand, our approach allows the use of natural gestures for pushing virtual objects by detecting the pushing intention and simulating the effect of such a push in an intuitive manner while offering tolerance to the hand positioning and tracking errors. We demonstrate in section 5.8 that these aspects lead to a significant increase of efficiency and versatility of our

methods over what the physics based methods can offer.

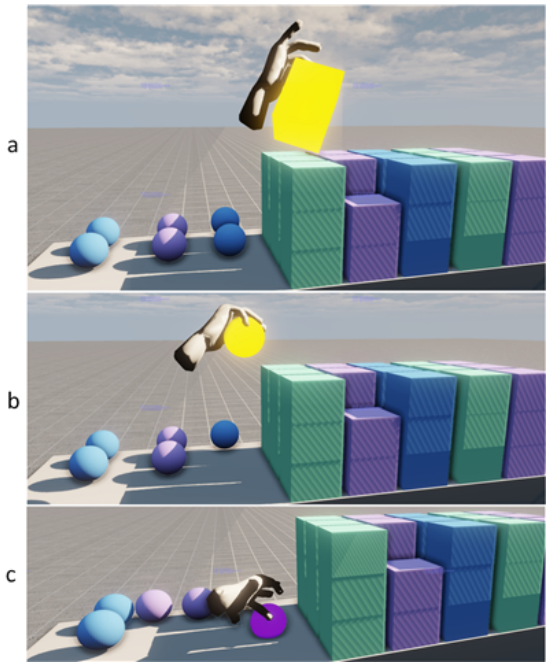

Figure 5.4: a) Precision grip, b) power grip, c) the guiding push of a sphere

The object hitting intention is a particular case of pushing and, consequently, is detected by the same algorithm. The difference between hitting and pushing gestures is that the former exhibits large accelerations of the wrist. The hitting intention is simulated by applying a force impulse on the object that is being hit along the direction of the hand movement and having a magnitude proportional to wrist speed before the impact. Examples of virtual object manipulations can be seen in Figure 5.4.

5.6 Inferring Users' Assembly Intention

The hand placement inaccuracies and body tracking faults become a serious impediment during virtual object assembly tasks. Since there are hundreds of different ways (figure 5.5) in which two simple brick models can be assembled, we have developed

a probabilistic graphical model to represent users' behavior during assembling tasks and automatically infer the fashion in which they intend to couple the assembled objects. In this manuscript we demonstrate our approach while inferring users' intent to apply the following assembly constraints: incidence and coincidence constraints between a vertex and another vertex, an edge, an axis or a planar face; parallelism and coincidence constraints between two edges or axes; and coplanar constraints between faces.

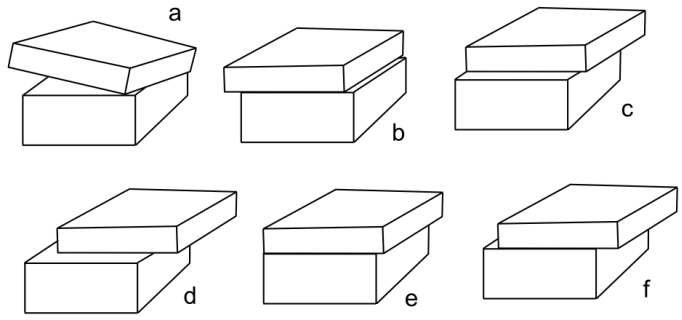

Figure 5.5: A few of the various ways in which two parallelepipeds can be coupled. a) The configuration in which the user might be able to position the objects by means of free hand movements. b-f) exemplify potentially desired assemblies.

5.6.1 Behavioral Cues Characterizing Specific Assembly Intentions

Our probabilistic graphical model is conditioned on the behavioral cues introduced in section 5.1.4. In the context of our application, these behavioral cues are defined to characterize users' intent to enforce various assembly constraints.

The *action efficiency* behavioral cue ($\boldsymbol{a_{eff}}$) estimates the effort required to enforce a specific assembly constraint. We will define this cue to be inversely proportional to the following distance to constraint satisfaction function ($\boldsymbol{d_S}$).

$$a_{eff} = \frac{1}{d_S} \qquad d_S = \frac{d}{[mm]} + \frac{a}{[deg]} \tag{5.1}$$

Where $\boldsymbol{d}$ represents a distance metric and $\boldsymbol{a}$ an angle measurement that characterizes each specific type of constraints that we evaluate. For example, in the case of a coplanar constraint between two faces, 'd' represents the distance between the evaluated faces while 'a' is the angle between them. In the case of a parallelism constraint between two edges 'a' represents the angle between the edges and 'd' is 0. While for the coincidence between edges 'd' is the distance between the edges and 'a' has the same significance as in the previous case. In the case of coincidence constraints between vertices or the incidence constraints between a vertex and an edge, axis or face 'd' is the distance between the vertex and the other coupling element, while 'a' is ignored.

The *action persistence* ($\boldsymbol{a_p}$) behavioral cue estimates the level of perseverance with which the user attempts to apply a particular assembly constraint. More specifically, this cue represents the frequency with which the user gets close to satisfying the evaluated constraint (see section 4.3 [169]). To measure the closeness to satisfying a constraint, we use the distance to constraint satisfaction metric described above. In this manner, for those users who show hand jitter, hesitation or lower hand control, the a_p behavioral cue will assume high values with respect to the constraint around which the user's virtual hand oscillates. As a result, our inference method will be able to identify the targeted assembly constraint at an early stage, and tolerate such hand placement or body tracking faults.

The *action duration* ($\boldsymbol{a_d}$) behavioral cue represents the number of steps with which the user advances towards satisfying a particular assembly constraint. The advancement is established based on the d_S function and each step is counted at approximately even time intervals (33 ms). In the case in which the d_S metric shows a decrease at the moment of users' advancement evaluation, the a_d behavioral clue will be incremented otherwise a_d will decrease by one unit. In figure 5.6 you can see the evolution of the values assumed by our behavioral cue metrics during several virtual object assembly tasks.

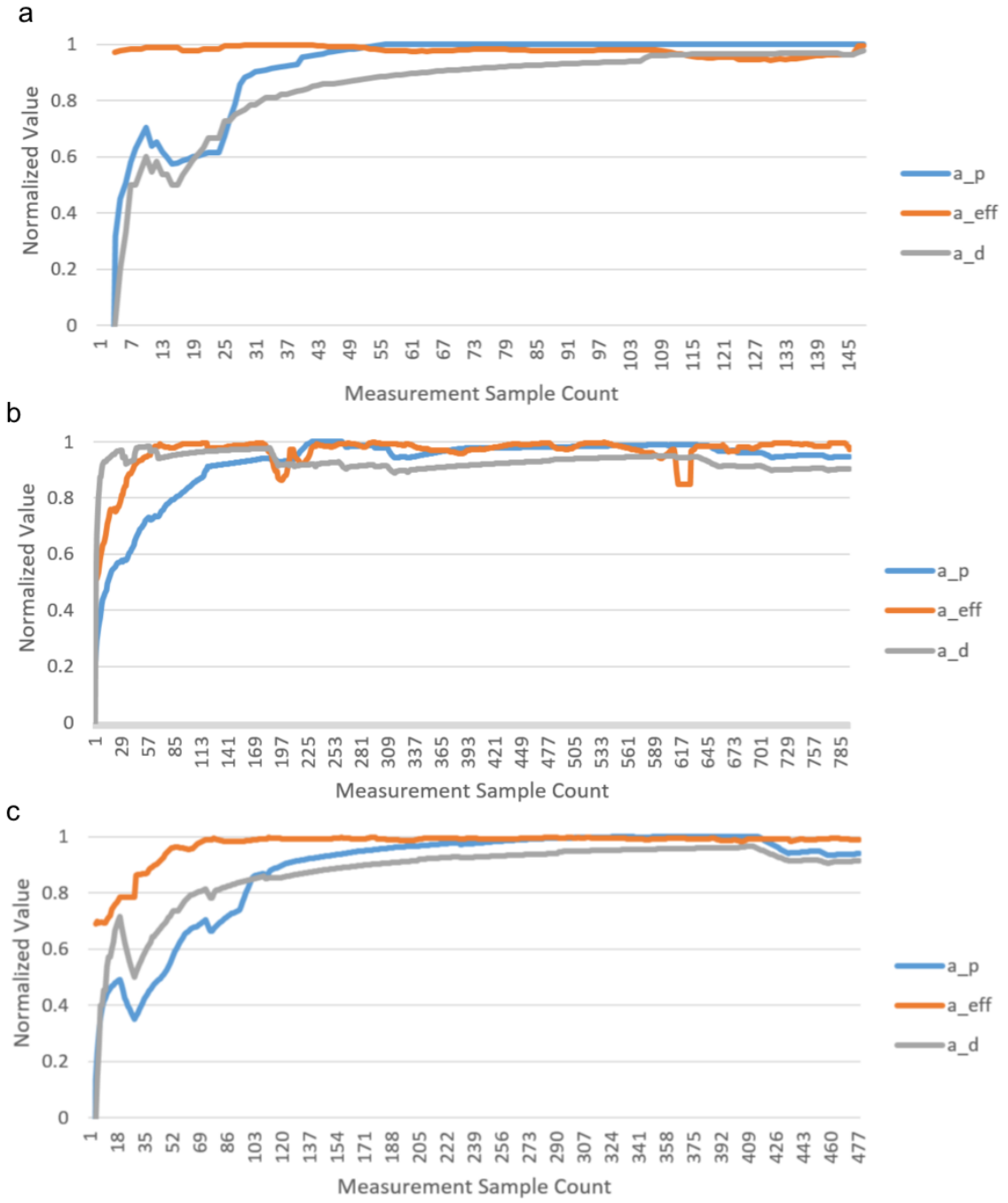

Figure 5.6: The evolution of our behavior cue metrics while the user attempts to: a) overlap two faces, b) apply an incidence constraint between a corner and an edge, c) apply a coincidence constraint between two corners.

The neuropsychology literature documents the correlation between such behavioral cues and intentional actions. However, the exact relationship between these cues and users' intention remains to be discovered. Therefore, we will rely on machine learning techniques to estimate this relationship.

5.6.2 Modeling Users' Behavior During Assembling Tasks

One of the most popular techniques used to represent an uncertain chain of events that describe an activity is the Hidden Markov Model (HMM) [170, 171]. Such models are capable of representing mutual dependencies between events or observations, and their parameters are relatively easy to interpret. However, they cannot represent complex contextual information in an efficient manner [172]. Also the HMM models require independent feature vectors [167], which fact becomes an impediment in our case, since all our behavioral cues depend on the distance to constraint satisfaction. Another common class of approaches are the Dynamic Bayesian Networks (BN) [173, 174]. Their intuitive structure makes them easy to design and their parameters are easy to interpret, but BN can only represent unidirectional dependencies between events or random variables. Conditional Random Fields (CRF) have been successfully used in [175, 176, 177]. Such models can be conditioned on heterogeneous contextual information and represent mutual dependencies among events. However, their parameters are less interpretable [172] and computationally expensive to estimate. Various other models have been proposed and most of them are reviewed in [167, 178].

We aim to identify the assembly constraints which the user intends to enforce while manipulating virtual objects. Since our main source of information about users' intentions are our uncertain observations about their behavior, which often includes accidental movements, we will treat the potential assembly constraints, or intentions, as random variables. Given the fact that the enforcement of an assembly constraint

can prevent or facilitate the enforcement of another constraint, our behavioral model must be able to represent the mutual dependencies between our random variables. Therefore, we choose to represent users' behavior using a CRF model in which the potential assembly constraints are random variables conditioned on behavioral cues. As shown in figure 5.7 our random variables are represented by the $\boldsymbol{C_{ij}^{l}}$ nodes while the $\boldsymbol{e_{cc}}$ edges of influence represent the mutual dependencies between the potential assembly constraints. In our notation the $\boldsymbol{l, p, q} \in [1, 2..Q]$ indices represents a particular constraint out of the total Q constraints that can be applied between object 'i' and object 'j' or 'k'. Therefore, our CRF model, represents users' potential intent to apply one or more constraints 'l', 'p' and/or 'q' between object 'i' and object 'j' in the presence of one or more objects 'k', where $\boldsymbol{i, j, k} \in [1, 2, ..M]$.

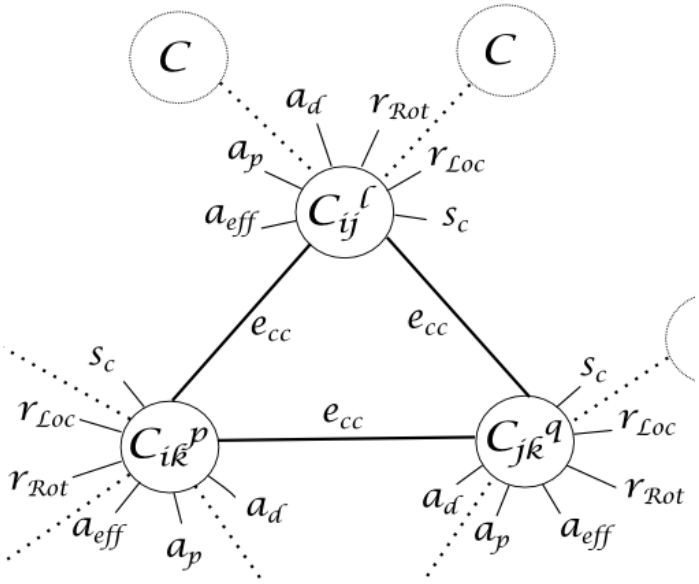

Figure 5.7: The Conditional Random Field Model representing users' potential intent to apply one or more constraints 'l', 'p' and/or 'q' between object 'i' and object 'j' in the presence of object 'k'.

The random variables in our CRF model are not only conditioned on behavioral cues (a_{eff}, a_p, a_d) but also on the constraints supported ($\boldsymbol{S_C}$) by the geometry of the manipulated objects as well as the relative location ($\boldsymbol{r_{Loc}}$) and orientation ($\boldsymbol{r_{Rot}}$)

between the assembled objects. The dashed edges in figure 5.7 represent the various other potential constraints which the user might intend to apply a specific moment in time. In the attempt to describe a general application case for the developed methods, in figure 5.7 we illustrate the form taken by our model when we have three or more potential constraints that can occur between three or more objects i,j,k. However, we can have multiple constraints occurring between only two objects. In that case the C_{jk}^{q} random variable is replaced by C_{ij}^{q} and the following model changes in a straight forward manner.

5.6.3 Parameterizing Our Model

We parametrize our CRF model using log linear edge and node potentials: In all the equations below, the $\boldsymbol{y}$ terms represent indicator functions used for classification, while $\underline{\boldsymbol{\phi}}$ represent feature vectors.

The node C_{ij}^{l} potential takes the form:

$$\Psi_{c}^{t}(ij) = \sum_{l \in K_{ij}} y_{ij}^{l} \left(\underline{w}_{c} \cdot \underline{\phi}_{c}^{t}(i,j) \right) \qquad y_{ij}^{l} = \begin{cases} 1 & \text{if } C_{ij} \text{ can take label 'l' at time 't'} \\ 0 & otherwise \end{cases} \tag{5.2}$$

Where K_{ij} represents the set of potential constraints or labels that node C_{ij} can assume at time t and $\underline{w}_{c}$ is a weight vector learned during the training process.

The potential associated with the e_{cc} edge which joins nodes C_{ij} and C_{ik}:

$$\Psi_{cc}(ij,ik) = \sum_{(l,p) \in (K_{ij} \times K_{ik})} y_{ij}^{l} y_{ik}^{p} \left(w_{cc} \cdot \phi_{cc}^{lp}(ij,ik) \right)$$
$$y_{ij}^{l} = \begin{cases} 1 & \text{if constraint } C_{ij} \text{was labeled 'l'} \\ 0 & otherwise \end{cases} \tag{5.3}$$

Where w_{cc} represents a weight vector learned during the training process such that $w_{cc} \cdot \phi_{cc}^{lp}(ij, ik)$ is maximized when the user attempts to simultaneously apply constraint 'l' between objects 'i' and 'j' and constraint 'p' between objects 'i' and 'k'.

The $\underline{\phi}_c$ feature vector is built out of the normalized a_{eff}, a_p and a_d behavioral cues while the ϕ_{cc}^{lp} features capture the relative evolution of the a_{eff} behavioral cue calculated for the 'l' constraint with respect to the same parameter evaluated for the 'p' constraint. More specifically, using the data captured for the a_{eff} parameters during the past 10 seconds we evaluate $\phi_{cc}^{lp}(ij, ik)$ as follows:

$$d_{SS}(\hat{C}_{ij}^{l}, \hat{C}_{ik}^{p}, t) = \left| a_{eff}\left(\hat{C}_{ij}^{l}, t\right) - a_{eff}\left(\hat{C}_{ik}^{p}, t\right) \right| + \frac{1}{a_{eff}\left(\hat{C}_{ij}^{l}, t\right)} + \frac{1}{a_{eff}\left(\hat{C}_{ik}^{p}, t\right)} \quad (5.4)$$

$$\phi_{cc}^{lp}(ij, ik) = \frac{1}{\min_{t}\left(d_{SS}\left(\hat{C}_{ij}^{l}, \hat{C}_{ik}^{p}, t\right)\right)} \qquad t \in (t_0 - 10s, t_0) \quad (5.5)$$

Here d_{SS} represents the distance to simultaneous satisfaction of the evaluated constraints 'l' and 'p'. The $\hat{C}_{ij}^{l}$ term represents the estimated label for the C_{ij} variables while t_0 is the current time value. The d_{SS} function is minimized when both constraints are close to being satisfied.

The compatibility of a particular label assignment is measured for each of our nodes using the following energy function:

$$E = \sum_{r \in V} \Psi_r^t + \sum_{(r,u) \in E_{cc}} \Psi_{ru} \quad (5.6)$$

$$E = \sum_{r \in V} \sum_{l \in K_r} y_r^l \left(\underline{w}_c \cdot \underline{\phi}_c^t(r)\right) + \sum_{(r,u) \in E_{cc}} \sum_{(l,p) \in (K_r \times K_u)} y_r^l y_u^p \left(w_{cc} \cdot \phi_{cc}^{lp}(r, u)\right) \quad (5.7)$$

For a simplified notation, above we used V to represent the set of vertices in our model while K_r and K_u represent the set of labels that can be assigned to each of our vertices. Similarly, E_{cc} is the set of edges included in our model. The energy function

E will be maximized when the feature vector $\underline{\phi}_c$ is likely to represent users' intent to apply a constraint 'l' between objects 'i' and 'j' while ϕ_{cc} is likely to represent users' intent to simultaneously apply multiple constraints.

5.6.4 Training and Employing the Model

While training the model, the weight vectors $\underline{w}$ are learned such that the energy function E is maximized given a labeled training data set: $(x_n, y_n), n \in [1, N]$. Here x_n represents an observation and y_n the label associated with it. To achieve this, we will minimize the average Hamming loss function, or training error defined as:

$$L\left(\bar{y}, \hat{y}\right) = \sum_{r \in V} \sum_{l \in K_r} \left| \bar{y}_r^l - \hat{y}_r^l \right| \tag{5.8}$$

Where $\bar{y}$ represents the ground truth labeling solution given as training data and $\hat{y}$ represent the labels inferred based on our current estimate for the weight vector $\underline{w}$.

In what follows $\underline{w}$ represents the weight vector $\underline{w}_c$ that has appended at its end the values of w_{cc}. Similarly, we build vector Ψ by merging the vectors $y_r^l \underline{\phi}_c^t(r)$ and $y_r^l y_u^p \phi_{cc}^{lp}(r, u)$ used in equation 5.7. As shown in [179, 172, 176], we can minimize our loss function by solving the following quadratic optimization problem :

$$\begin{aligned} &\min_{w, \xi \geq 0} \frac{1}{2} \underline{w}^T \underline{w} + C\xi \\ &s.t. \forall \bar{y}_1, ..., \bar{y}_N \in \{0, 0.5, 1\}^Q : \frac{1}{N} \underline{w}^T \sum_{i=1}^{N} \left[\Psi\left(x_i, y_i\right) - \Psi\left(x_i, \bar{y}_i\right) \right] \geq \frac{1}{N} \sum_{i=1}^{N} L\left(\bar{y}_i, y_i\right) - \xi \end{aligned} \tag{5.9}$$

Where C and ξ represent the constant and the slack variable specific to the SSVM optimization algorithm proposed in [179]. The efficiency of the algorithm depends on

the speed with which we evaluate the following expression:

$$\bar{y}_i = \underset{\hat{y}\in\{0,0.5,1\}^Q}{\text{argMax}} \left[L\left(y_i, \hat{y}\right) + \underline{w}^T \Psi\left(x_i, \hat{y}\right) \right] \tag{5.10}$$

As explained above the $\underline{w}^T\Psi$ term represents the energy function E. In equation 5.7 the indicator variables y belongs to the $\{0, 1\}$ domain. Since each vertex in our model can take only one label, the y variables are subject to the following constraint: $\forall r \in V : \sum_{l\in K_r} y_r^l = 1$. In order to resolve this maximization problem (equation 5.10) in an efficient manner we relax the solution space of our y variables to the $\{0, 0.5, 1\}$ domain and employ the GLPK [180] mixed-integer programming solver.

During the assembly intention inference procedure, the energy function E is maximized with respect to the indicator variables y, given the learned parameters $\underline{w}$ and the observed behavioral cues $\underline{\phi}$. This inference problem can be formulated as a particular case of equation 5.10 in which the loss function L is 0. Therefore, the inference can be resolved using the same mixed integer programing approach mentioned above. As a result, the assembly constraint recognition is achieved in real time (0.0012s) while using a 4 core CPU running at 2.5 GHz and 16 GB of RAM. The CRF model is learned offline, in approximately 34 minutes, when the training is done on 250 assembly examples.

5.7 Paving the Way towards User-Centered Interfaces

It is well-known that human skills, including dexterity, vary significantly among people. By identifying users' intent to perform manipulative tasks that involve high dexterity skills, our system is able to assist them in achieving the intended manipulation by increasing the resolution with which they control the virtual hand model.

Namely, once an assembly intention is detected, the control-display (CD) ratio can be dynamically varied [34] such that a large or coarse hand movement in the physical space can control fine hand model movements in the virtual environment. The same principles of intention detection are embedded in our IDS selection method [169]. Aided by this technique our virtual object manipulation methods are capable of adapting automatically to the users' subjective need for various levels of hand placement and tracking fault tolerance. Such need depends on factors like dexterity, visual acuity, and subjective preferences.

The level of hand placement fault tolerance offered by our system increases proportionally to the values assumed by the a_p and a_d behavioral cues as well as the opening of the user's hand (see section 4.1). Note that a person with inferior hand control will instinctively open the hand more in the attempt to perform a coarser object selection, or a power grip instead of a precision grip on the objects of interest. By identifying such behavioral cues our system automatically adapts to the user's subjective need for higher hand placement fault tolerance. A similar observation can be made for the case in which a user is uncertain about the relative position of his/her hands with respect to the virtual objects that the user intends to grab. Such uncertainty can be caused by the subjective ability of the user to perceive the elements of the virtual scene and/or their location in the virtual 3D space. Furthermore, for the cases in which the user shows hand jitter, hesitation or lower hand control, the a_p behavioral cue will assume high values with respect to the coupling element or object around which the user's virtual hand oscillates. As a result, our inference method is able to identify the targeted assembly constraint at an early stage, and tolerate such hand placement or body tracking faults.

In other words, the methods described in this paper contribute to the advancement towards user centered interfaces by adapting to the subjective abilities and needs of the user.

5.8 Empirical Evaluation

Different people took part to the different user studies described below. Before taking part in the actual tests, the participants witnessed a brief demonstration of the capabilities of the interface. Then, they were allowed to experiment by themselves with the elements of the interface for no more than 5 minutes. The participants were also informed that they could use their natural gestures, however they prefer in order to accomplish the tasks involved in the tests below.

5.8.1 Guiding Push

In the following study the users stood approximately 2m in front of the projective screen described in the System Setup section. A Kinect camera was placed in front of the screen to track the user's movement. This pilot study was done before we integrated the finger tracking camera from LeapMotion into our system. Therefore, in order to track the approximate flexion of our participants fingers we asked them to wear on their right hand a data glove equipped with 5 flex sensors.

In the test below we evaluate the performance differences between the proposed virtual object push method (IDP) and the physics based push method (PBP). While using the PBP method the object pushing is controlled by simulating the contact forces between the hand model and the virtual objects. We choose to compare our method with the PBP method due to the fact that the physics based manipulation methods are well established approaches, that are aimed at the goal at which our methods aim: enabling the manipulation of virtual objects by means of natural hand gestures. Therefore, we are comparing the efficiency of these two methods and the capabilities they afford for executing common but challenging pushing tasks. In order to do so we test the two pushing methods on bodies bounded by surfaces that facilitate their placement in a stable equilibrium position such as cubes and unstable bodies

such as spheres. These objects are pushed over flat and inclined surfaces. The pushed cubes have a 3cm side length, while the spheres have a radius of 2.25cm (figure 5.8).

During the test we evaluate the hypothesis stating that the IDP method is more efficient than the PBP method for pushing virtual objects using natural hand gestures.

Test Population

Thirty users took part in our experiment. Their age range between 20 and 35, median age 25, including 13 female participants and 4 left handed. Eleven have declared that they do not play video games or work with 3D CAD software packages and virtual environments. Eight are sometimes using such 3D environments and 11 use them frequently. The test lasted approximately 30 minutes and the participants were compensated 10$ for their participation.

Pushing on Horizontal Surfaces

In order to evaluate the efficiency of our pushing method we considered the following performance parameters: 1) *Time efficiency*, which is the amount of time spent by the user while completing a pushing task, and 2) *Perceived effort*, which is the amount of effort spent by the user while performing the task. The most efficient pushing method is considered to be the one which minimizes both of the above performance parameters.

To evaluate these parameters, the participants were asked to position a body on a designated target by pushing it. The participants are free to choose any body belonging to the group of cubes or spheres shown in figure 5.8. During these tasks the hand model wore a semitransparent mesh such that objects could not become occluded by it. Once a new placement target is assigned a distinctive sound is played and the designated target blinks purple (figure 5.8 case 1) until the user performs a *stable placement* of an object on it. An object placement is considered to be stable if

the object remains in contact with the red center of the placement target for at least 2s. During this time the placed object blinks bright green as seen in figure 5.8 case 1.

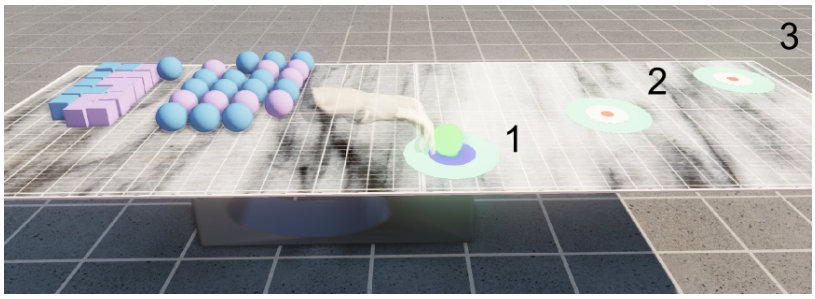

Figure 5.8: Pushing on horizontal plane surfaces: push case 1) over short distances, case 2) over medium distances, case 3) over large distances and increased noise conditions.

After the participant performed a stable placement the system assigned a new target. The time passed between the moment in which the target was assigned and the moment in which the user completed a stable placement was recorded and used for measuring *the time efficiency parameter*. Following the procedure above, the user positioned objects on all the targets shown in figure 5.8. After performing a stable placement on each target the participant was notified that the pushing method will be switched to the other push method. Then the exact same procedure was followed while using the other method of pushing.

In order to minimize the influence of chance on the test outcomes this process was repeated 3 times for each type of object, pushing method and user. Therefore, each participant tried to position 3 spheres and 3 cubes using each of the two pushing methods. As expected the first pushing trials are the slowest for each participant. To avoid biasing the data, the first pushing method with which the test process begins was changed with each user. Therefore, the tests are counterbalanced and the 2 pushing methods are evaluated in identical conditions. At the end of the test each user was asked to evaluate the following statements:

"1. The Intent Driven Push method requires less effort for pushing virtual objects

than the Physics Based Pushing method."

☐ strongly agree ☐ agree ☐ neutral ☐ disagree ☐ strongly disagree

"2. It was easier to push virtual objects while using the Physics Based Pushing method than with the Intent Driven Push method."

☐ strongly agree ☐ agree ☐ neutral ☐ disagree ☐ strongly disagree

The above Likert scale is used to evaluate the *effort efficiency* parameter. We use the second statement in order to identify and avoid wording related misunderstandings.

Results

As figure 5.9 shows the IDP method appears to be 150% faster than the PBP method while pushing stable objects. This means that the time required by the PBP method for achieving the aforementioned pushing task represents 250% of the time required by IDP.

We run repeated measures one-way ANOVA tests to verify if the collected data provides significant evidences to support the above observations. The variances of the data collected for the 2 methods are stabilized by applying a natural logarithm transformation on the timing data. The results show that the IDP method allows users to position virtual objects significantly faster than the physics based pushing alternative ($F_{1,29} = 117.2, p < 0.001, \eta^2 = 0.8$). Furthermore, while trying to place spheres by using the PBP method none of the participants managed to complete one test trial. Three participants managed to position a total of 8 spheres on 2 of the assigned targets before they abandoned the test. On the other hand, by using the IDP method all participants managed to place spheres on all 3 targets 3 times, and

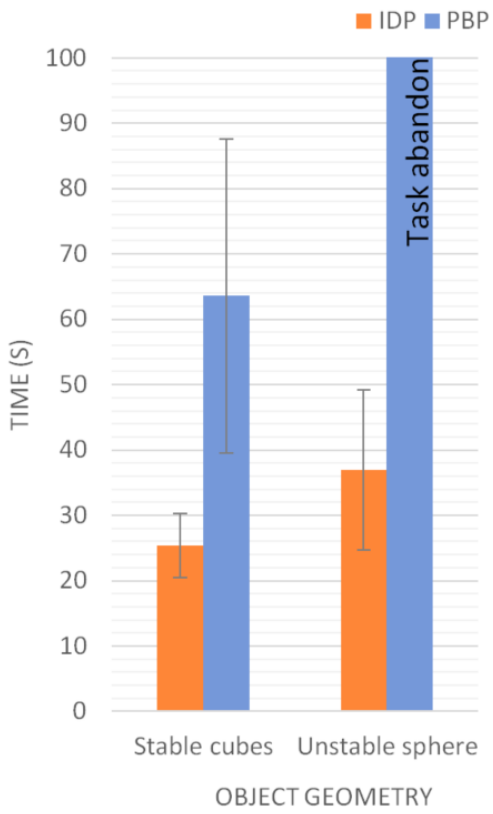

Figure 5.9: The total pushing time parameters averaged over the entire test population. The error bars represent the standard deviation of the performance of each user while pushing on horizontal surfaces

therefore complete the test.

To evaluate the interaction between the pushing distance and the pushing method as well as the effect of this interaction on the task completion time we run a two-way multivariate repeated measures ANOVA test in which the push method and the distance of push are the independent variables. The results show that there is no significant interaction between these two factors in terms of pushing times ($F_{1,29} = 3.1, p > 0.05, \eta^2 = 0.09$).

Pushing on Inclined Plane Surfaces

Here we test the efficiency and the capabilities offered by the two methods for pushing objects in the following common cases: 1) Pushing an object up an inclined surface. 2) Pushing an object while the object and the hand model become occluded during pushing. 3) Pushing under tracking noise conditions.

The participants that took part in the previous push test also are took part in the current test.

Procedure

Following a similar procedure to what was previously described for pushing, we tested the 2 methods on the pushing case illustrated in figure 5.10. The slope of the ramps on which the users had to push their objects is 35 degrees. The second pushing case occurred when users were pushing objects behind the first ramp in order to bring them close to the second placement target and the third case occurred while pushing objects on the ramp situated in the corner of the table. The table top was positioned such that its side edges are in close proximity to the limits of the field of view (FOV) of the tracking camera. When the users try to get close to these limits, parts of their body might leave the FOV of the camera or might get occluded by other body parts. In consequence, the image processing and tracking algorithms cannot collect sufficient information about the position of users' body parts in order to produce reliable outputs. This fact translates into an increased frequency of tracking noise occurrence. In order to save time, the users were allowed to try reaching each of the assigned targets for 3 minutes. If they did not manage to succeed or if they abandoned the task the time measurement was considered to be 3 minutes.

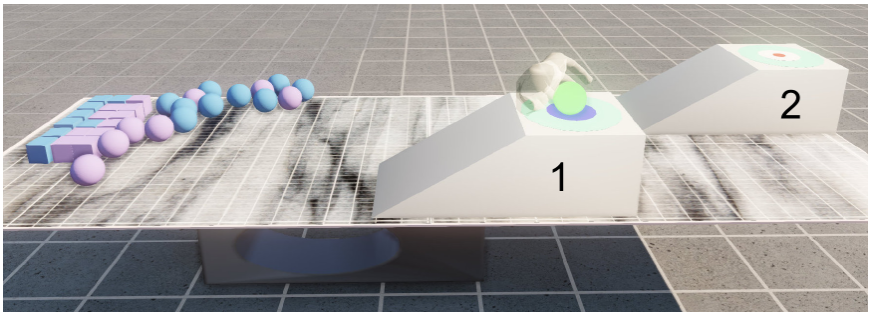

Figure 5.10: Pushing on inclined plane surfaces: push case 1) over short distances, case 2) over large distances and increased noise conditions.

Results

While employing the PBP method, none of the users managed to push a sphere onto the assigned targets. Out of the 180 trials of pushing cubes up onto the inclined plane using the PBP 26.6% resulted in task abandon. On the other hand, when using the IDP method the users managed to complete the test without a single abandon during the cube pushing trials as well as the sphere pushing trials. Furthermore, the IDP method showed to be more than 504% faster than the PBP method during the cube pushing task (figure 5.11). Following the procedure used in the previous pushing test we verify to see if the collected data provides enough evidence to support these observations. The one way repeated measures ANOVA test shows that the IDP method is significantly faster than the PBP method: $F_{1,29} = 274.3, p < 0.001, \eta^2 = 0.9$.

Interpreting the Pushing Test Results

In all the pushing tests described above the IDP method proved to be remarkably faster than the PBP. In addition, 83.3% of the participants strongly agreed that the IDP method requires less effort for selection than the PBP method while the rest agreed with the same statement. If we map these responses on a scale between 0 (Strongly Disagree) to 1 (Strongly Agree) the standard deviation of users' response

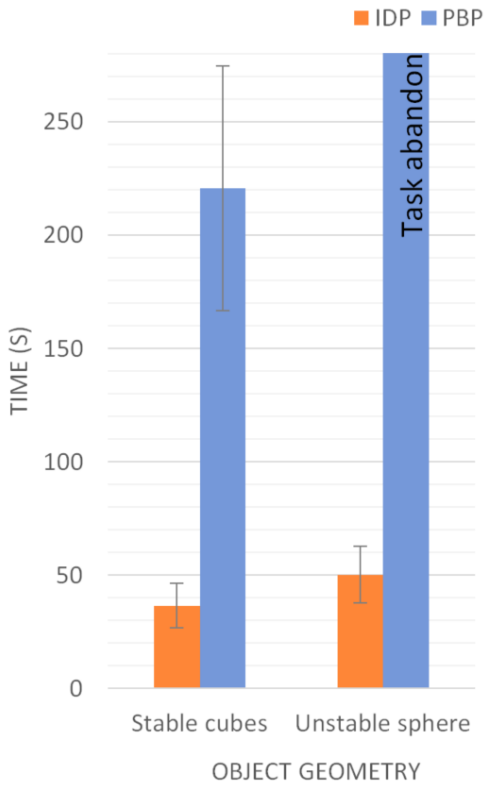

Figure 5.11: The total pushing time parameters averaged over the entire test population. The error bars represent the standard deviation of the performance of each user while pushing on inclined plane surfaces.

is 0.06. These results confirm the hypothesis which states that the IDP method is more efficient than the PBP pushing technique. While using the PBP method none of the users managed to complete a full test on pushing spheres. Furthermore, they gave up in 26% of the trials even while attempting to push cubes on inclined surfaces using the PBP method. The manipulation difficulties that led to such results were mainly caused by the hand placement imprecision shown by the users and the tracking uncertainties previously explained. In those cases in which the pushing brings into contact surfaces that do not offer a stable physical equilibrium, such as the spherical surfaces or the inclined planes, a single hand placement fault often leads

to a large uncontrolled displacement of the manipulated object. Often times, due to tracking uncertainties, the manipulated objects were struck by the hand model with uncontrolled force in an uncontrolled direction. As our data shows, the above problems are severely affecting the manipulation capabilities offered by the physics based pushing method. Because all physics based manipulation methods rely on the same physics principles we can conclude that all such methods will suffer similar limitations. These results come to confirm the arguments we used while comparing our methods with the physics based methods in the Related Work section: Due to the lack of tolerance to hand placement and tracking faults, the physics based methods become inefficient and even impractical under our work conditions. On the other hand, all test trials were completed by all users in a timely manner while using the Intent Driven Push method.

5.8.2 Manipulation Intention Recognition

Here we focus on analyzing the robustness of our classification methods to gesture variability. Note that the grasp release, object translation, and object rotation gestures have never been misclassified because they involve straightforward events that we can consistently detect and interpret correctly. Therefore we are testing the manipulation gestures that are occasionally misclassified: grasping, guiding push and hitting.

Test Population

Thirty participants were recruited to take part in this study. Their ages range from 22 to 35, with a median age of 25, including 14 female participants and 3 left handed. Twelve have declared that they do not play video games or work with 3D CAD software packages and virtual environments. Seven are sometimes using such 3D environments and 11 use them frequently. The test lasted approximately 20 minutes

and the participants were compensated 10$ for their participation.

Procedure

During the test the virtual environment shown in figure 5.12 was rendered. In order to evaluate the performance of our gesture classification methods the users were asked to perform 20 repeated trials of each of the 3 manipulation types: grasp, push and hit. The green objects shown in figure 5.12 have been assigned as target objects during the trials. Their size is $0.5 \times 3.3 \times 14$cm. These objects afford 15 different grip types, out of which 7 are illustrated in figure 5.13. The other grasp types can be seen in the grasp taxonomy presented in [11].

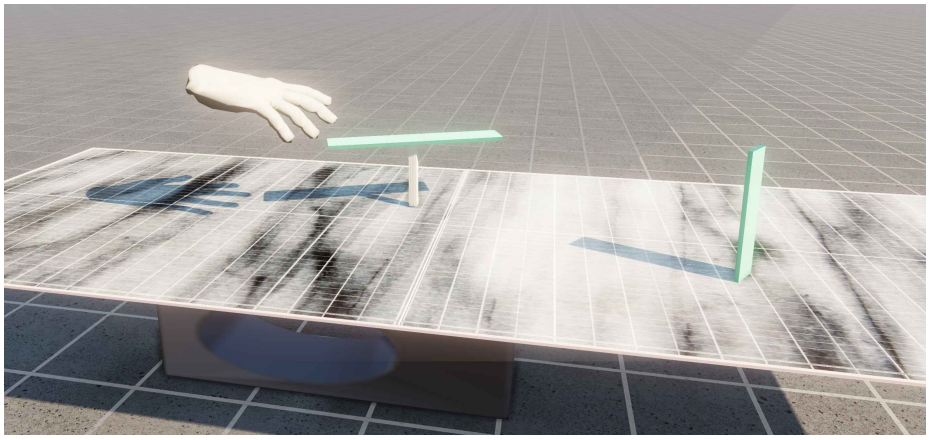

Figure 5.12: The intention recognition test

In the case in which the system classifies the currently observed actions to a grasping attempt the color of the object that is currently manipulated changes to yellow. Otherwise, if a pushing intent is inferred the color of the object at hand will change to purple, or to white if a hit is inferred. At the same time the effect of such intentions is simulated as previously described. The number of times in which the system inferred the wrong manipulative intentions was counted based on the observed graphical feedback as well as the user's feedback.

The guiding push gestures were executed using various parts of the hand as the

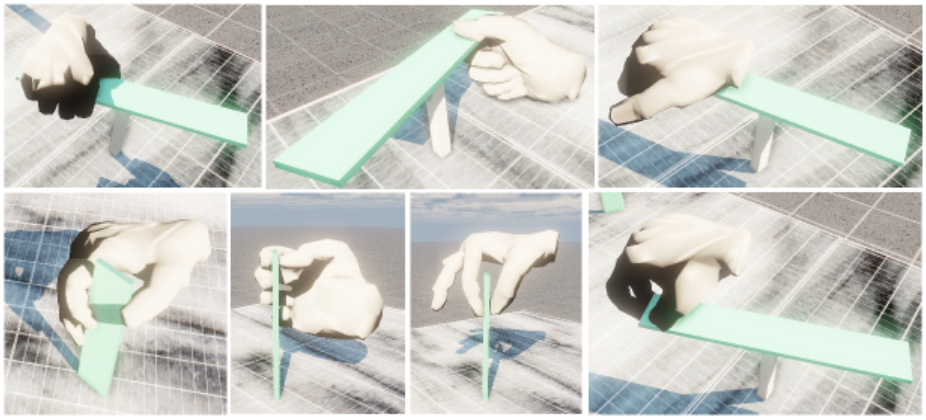

Figure 5.13: Several of the types of grips that are afforded by our experimental setup

contact surface: the finger tips, the outside and inside arch of the hand model, the lateral side of the hand that contains the little finger, etc. The recognition of various hitting gestures was evaluated in a similar manner as the guiding push gestures. The results of these tests are synthesized in the confusion matrix shown in figure 5.14.

True Intention	Classified Intention			
	Grasp	Guiding Push	Hit	Success Rate
Grasp	484	0	76	86.4%
Guiding Push	0	526	74	87.7%
Hit	3	14	583	97.2%

Figure 5.14: The confusion matrix summarizing the intention classification performance

Results

While testing the recognition of grasping intentions, the data acquired for two participants showed that our method has a classification success rate of more than two standard deviations away from the mean value of the test population, respectively 55% and 65%. One of the possible reason for these exceptional performances might be

the fact that both of these participants were tested at 7 PM after business hours and in consequence it is likely that they were less observant than the other participants. Also one of them was wearing a very dark shirt which made him more difficult to track using structured light imaging techniques, and therefore he faced more tracking noise than normal.

Without considering these two outliers the tests show an average success rate of 86.4% for grasping, 87.7% for pushing and 97.2% for hitting. The robustness of our methods against the hand gestures variability can be seen in the standard deviation of our success rates across the test population for: grasping 8.5%, pushing 7.4%, hitting 2.5%. The data shows that most of the misclassified gestures are confused with hitting gestures which correspond to highly accelerated hand motion. These inference errors are mainly caused by tracking uncertainties. By employing a larger number of tracking cameras that are properly distributed the amount of observation uncertainty as well as tracking uncertainties will be reduced. Another possible approach to improve the false hit classification is to involve more information into the decision making process. Machine learning techniques are often used to approach this type of problems.

5.8.3 Assembly Intention Inference

In the following user study, will compare the success rate of the proposed intent driven constraint recognition technique (IDCR) with the geometry dominant constraint recognition alternative. As explained in our Related Work section, the current state of the art constraint recognition techniques select a particular constraint based on geometric information alone. Namely, a constraint is selected if the output of a characteristic distance function is minimized or falls within a certain domain. Such distance functions are defined in terms of constraint specific angle and/or distance measurements [44, 45]. For example, the distance to constraint satisfaction function described in section 5.6 is one of such function, which is applicable to our general

assembly goal.

We tested the behavior of these techniques when applied to the general assembly case in which constraints can be applied between all the geometric primitives that define the manipulated objects. The assembled virtual objects are cubes where the face of each cube has two extra coupling elements, that represent sub regions of the face, as shown in figure 5.15. These surface sub regions have been added in order to test the performance of the evaluated methods for the case in which the user intends to position an edge or a corner on a particular sub region of the surface of a specific object. In this case the distance to constraint satisfaction is evaluated with respect to the center of the circular area that represents the sub region area. The size of the manipulated objects is $10 \times 10 \times 10$ cm while the radius of the circles that represent sub regions is 4 mm. Before employing the IDCR technique, we trained our CRF model using the data acquired while 10 different users attempt to execute various virtual object assemblies. None of the 10 users on which our inference model was trained participated in the tests described below.

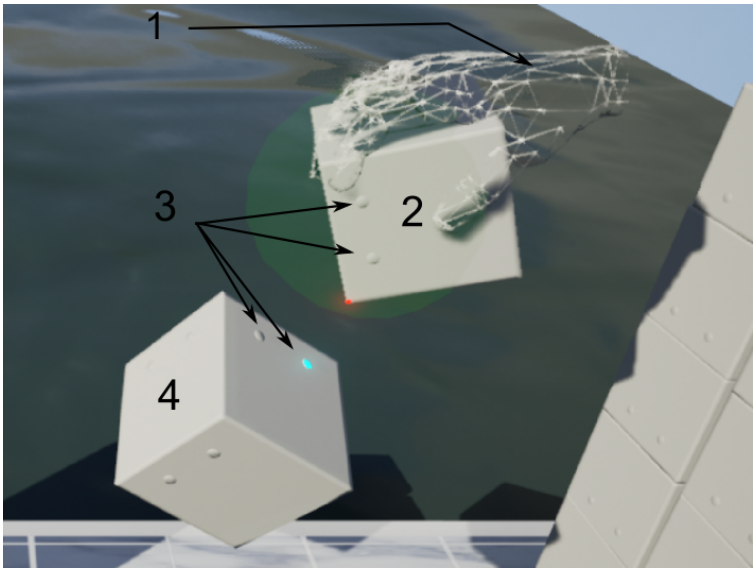

Figure 5.15: The Assembled virtual objects: 1) The virtual hand model, 2) The manipulated virtual object, 3) Examples of surface sub regions, 4) The virtual object on which the manipulated object is assembled.

Test Population

Thirty users took part in our experiment. Their age ranged between 21 and 57, median age 25, including 14 women. Nine have declared that they do not play video games or work with 3D CAD software packages and virtual environments. Twelve are sometimes using such 3D environments and 9 use them frequently. The test lasted approximately 75 minutes and the participants were compensated 10$ for their participation.

Apparatus

During this user study our test participants stood approximately 2m in front of the projective screen described in the System Setup section. In front of the screen a Leap Motion stereo camera was placed to track users' hands and fingers. On the screen, the virtual environment shown in figure 5.16 was displayed.

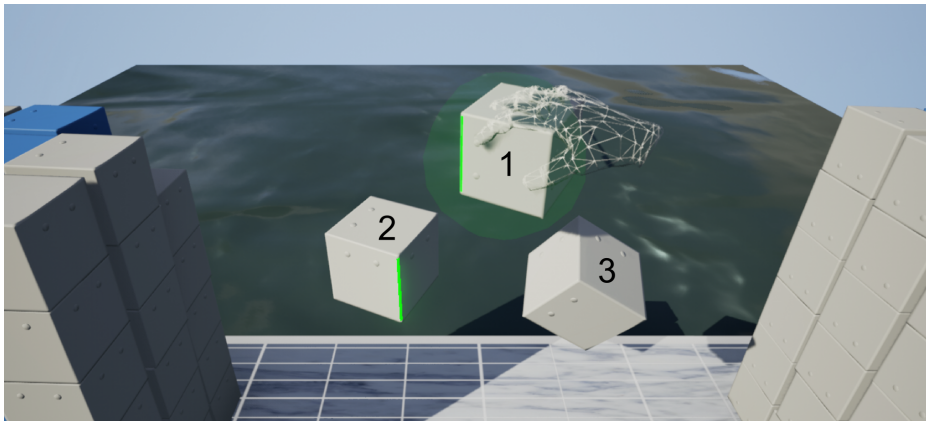

Figure 5.16: The environment displayed during the test: 1) The manipulated virtual object, 2) and 3) The passive virtual objects on which the manipulated object can be assembled.

Procedure

During the study, our participants were asked to grab and manipulate the cubical virtual objects shown in figure 5.16, in order to enforce a specified set of assembly constraints. For example, one of their tasks was to align two edges such that they are parallel or coincident. Before each trial we specified the type of geometric constraint that needed to be applied between the object which the user grasps and the suspended objects , respectively objects 2 and 3 in figure 5.16. In each assembly trial, the assigned constraint must be applied between well-defined geometric primitives. To avoid confusion, the targeted geometric primitives had their mesh highlighted in blinking bright colors, as can be seen in figure 5.16 on objects 1 and 2. Overall, each of our participants performed 25 trials in which they attempted to apply 7 types of constraints under different conditions. In total our test involved 750 assembly trials in which the users tried to apply the following constraint types:

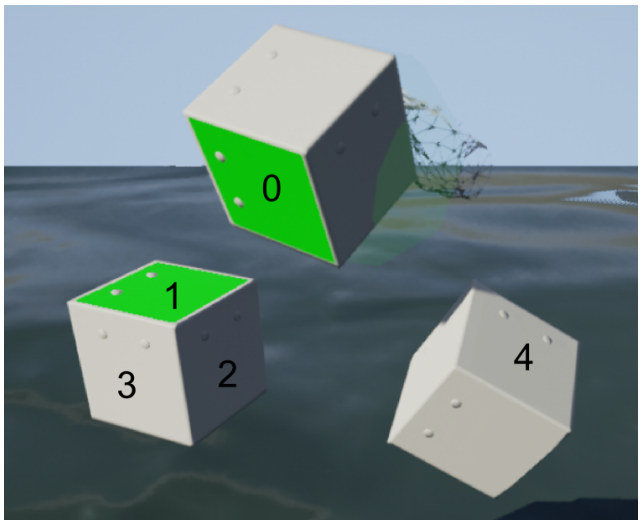

Figure 5.17: Applying coplanar constraints between faces: 0) The face of the manipulated object that was assigned to be constrained, 1) The face of the passive object that was assigned to be constrained, 2),3),4) The other faces of the two passive objects on which our participants applied coplanar constraints between faces during the other trials.

Case a) Coplanar constraints between faces: To test the recognition of this con-

straint, our participants were asked to overlap a designated face of the manipulated object on a specified face of another object such that the two faces were coplanar. Each of our participants tried to apply this constraint during 4 different situations in which different faces of different orientations were assigned to be constrained. The faces that were designated to overlap were highlighted in green as shown in figure 5.17.

Case b) Incidence constraints between points and faces: While testing the recognition of this constraint we asked our participants to position a corner of the manipulated object on the 3 different faces during 3 different trials. In each trial, the corner on which the constraint was applied was blinking bright red while the face subjected to this constraint blinked green as illustrated in figure 5.18.

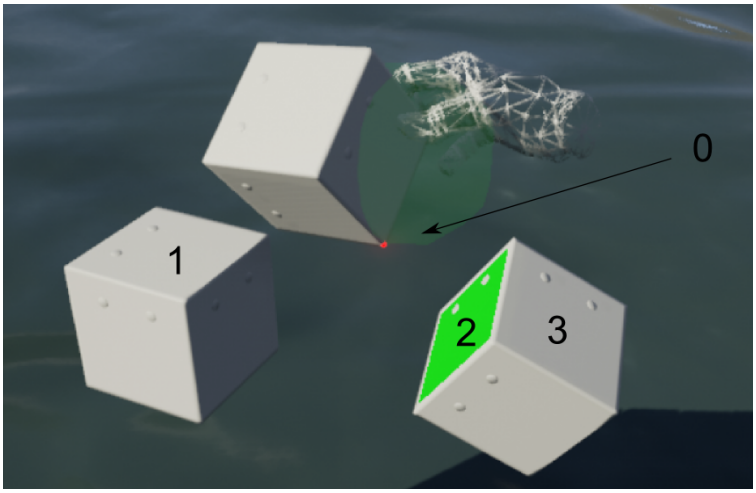

Figure 5.18: Applying incidence constraints between a point and a face: 0) The corner of the manipulated object that was assigned to be constrained, 2) The face of the passive object that was assigned to be constrained, 1),3) The other faces of the two passive objects on which our participants applied such constraints during the other trials.

Case c) Coincidence constraints between points: In this case, the constraint recognition is tested while our participants perform 5 trials to overlap several corners of the manipulated objects with 5 different corners of the passive objects. In each trial

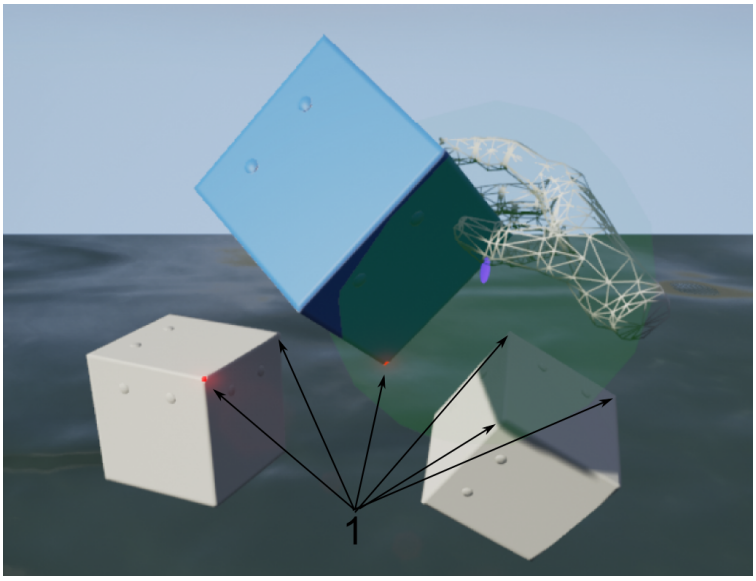

Figure 5.19: Applying coincident constraints between points: 1 marks the corners which were involved in our constraint recognition test. During each test trial the coincident constraint was applied between a corner of the manipulated object and one of the corners of the passive objects.

a pair of corners are assigned to be constrained and their mesh blinks bright red as shown in figure 5.19

Case d) Incidence constraint between edges and points: To test the recognition of this constraint type we asked our participants to bring into contact a corner of the manipulated object with one of the edges of the passive object. Each participant repeated this test three times for the three edges marked in figure 5.20. The corner that was assigned to be constrained was blinking red while the edge was blinking green as shown below.

Case e) Parallel constraints between edges: In order to test the ability to recognize this constraint we asked our participants to position a specified edge of the manipulated object such that it was parallel to a designated edge of the passive objects. The edges that were subjected to the constraint were blinking bright green. The users repeated this test trial on different edges having different orientation as shown in figure 5.21. Often during this test case our participants attempted to overlap the

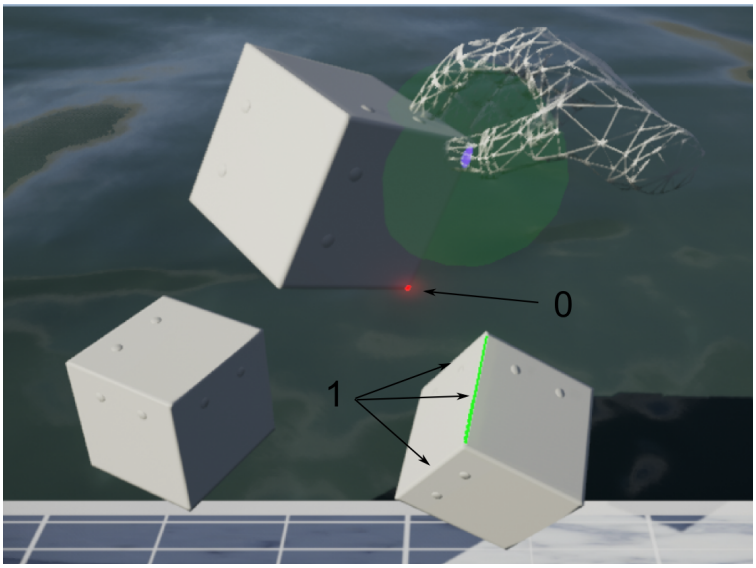

Figure 5.20: Applying coincident constraints between a point and an edge: 0) Marks the corner which was involved in our constraint recognition test. 1) Marks the edges of the passive objects on which the constraint was applied during the different trials.

designated edges due to the fact that when the edges overlap it is easier for them perceive the relative orientation of the edges that are constrained.

Case f) Intersection constraints between an edge and a sub-region of a surface: In order to test the recognition of this constraint type, we asked our participants to position a specific edge of the manipulated object such that it intersected a particular sub-region of one of the passive objects. During the test, each participant tried to apply this constraint on the sub-regions marked in figure 5.22. The edge of the manipulated object that was assigned to be constrained was blinking bright green during the test, while the sub region on which the constraint had to be enforced blinked bright blue.

Case g) Incidence constraint between points and sub-regions of surfaces: In this test trial we asked our participants to position a corner of the manipulated object on a particular sub region of the passive objects. Each participant attempted to apply this constraint on the different regions marked in figure 5.23. During the test the

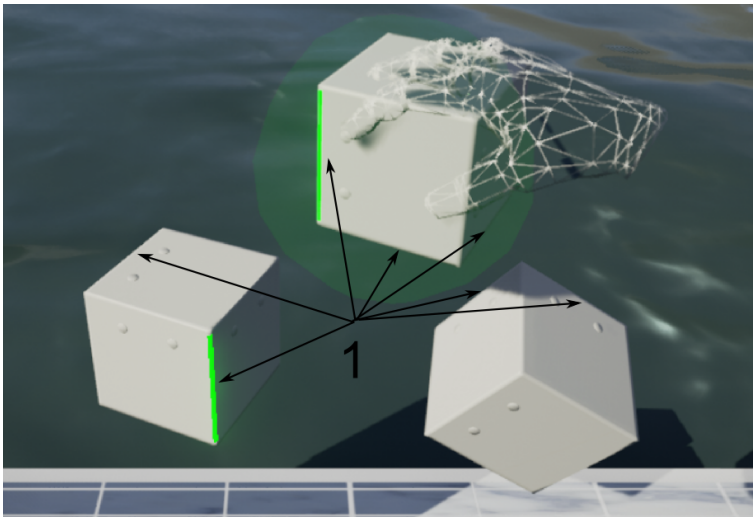

Figure 5.21: Applying parallel constraints between edges: 1 marks the edges which were involved in our constraint recognition test.

corner was blinking bright red while the sub-region on which the constraint had to be applied was blinking blue as shown below.

The tracking data that was acquired during this experiment was used to run the proposed constraint recognition technique as well as the alternative geometry dominant constraint recognition method. Therefore, the evaluated methods were tested in identical conditions, on identical input data. The output of these methods was an ordered list of the constraints considered to be the ones which the user intended to apply at a specific moment in time. The list is ordered in terms of the strength of the evidence that the user attempted to apply a specific constraint. In the case of the IDCR technique the user's strength of intent was estimated using the energy function defined in equation 5.7. While in the case of the geometric dominant technique the strength of evidence was inversely proportional to the d_S function.

Results

Often times when a specific constraint is applied other constraints are implicitly satisfied. For example, when we overlap two faces the corners that are adjacent to

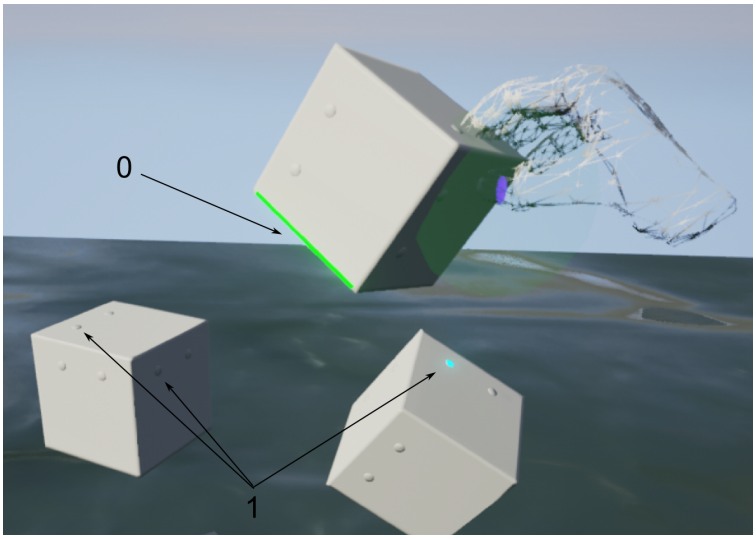

Figure 5.22: Applying intersection constraints between an edge and a sub-region of a surface: 1) marks the sub-regions which were involved in our constraint recognition test while 0) marks the edge assigned to be constrained.

one face can implicitly become incident to the other face. Therefore, it is important for our constraint recognition technique to be able to recognize an ordered list of the constraints which the user most likely intends to apply. Some of the constraints that are related to the targeted constraint are likely to become satisfied even before the targeted constraint is satisfied. Therefore, such constraints might be placed ahead of the targeted constraint on the hierarchy of the output list. However, the closer to the top of the hierarchy of the output list the targeted constraint becomes, the stronger is the intention disambiguation offered by the tested constraint recognition technique. We will assign to the top of the hierarchy an index 0 which increases as the hierarchical position of the targeted constraint decreases. In the following text we use the term negative score to refer to this hierarchical index. A constraint recognition failure occurs when the targeted constraint is not included in the output list or when untargeted constraints that are not implicitly satisfied along with the targeted constraint are wrongfully placed in the output list on a higher hierarchical position than the targeted constraint.

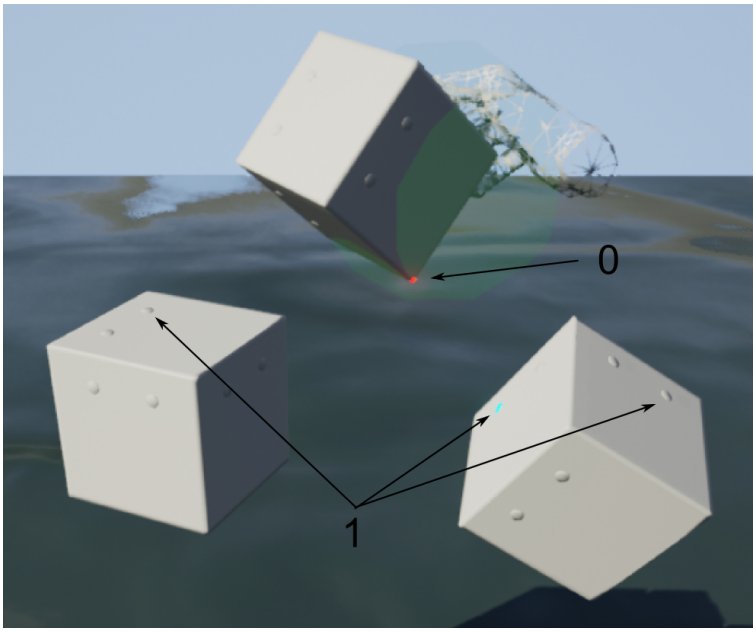

Figure 5.23: Applying incident constraints between a point and a sub-region of a surface: 1) marks the sub-regions which were involved in our constraint recognition test while 0) marks the corner assigned to be constrained.

Figure 5.24.a summarizes the results achieved across the entire test population in terms of the position of the targeted constraint in the hierarchy of the output list of the IDCR method. The median negative score achieved by our method is 3 while the standard deviation of this result is 4.54. Figure 5.24.b shows the same performance metric evaluated for the geometry dominant constraint recognition method. The median negative score achieved by this method is 8 while the standard deviation of this score is 7.58. This data shows that the proposed technique offers stronger assembly intention disambiguation than the geometry dominant constraint recognition alternative.

The majority of the negative scores larger than 19 represent unrecognized constraints. The graphs in figure 5.24 suggest a significant difference between the tested methods in terms of their rates of success. Across the 750 test trials the IDCR technique showed a rate of success of 89.14%. On the other hand, the alternative method

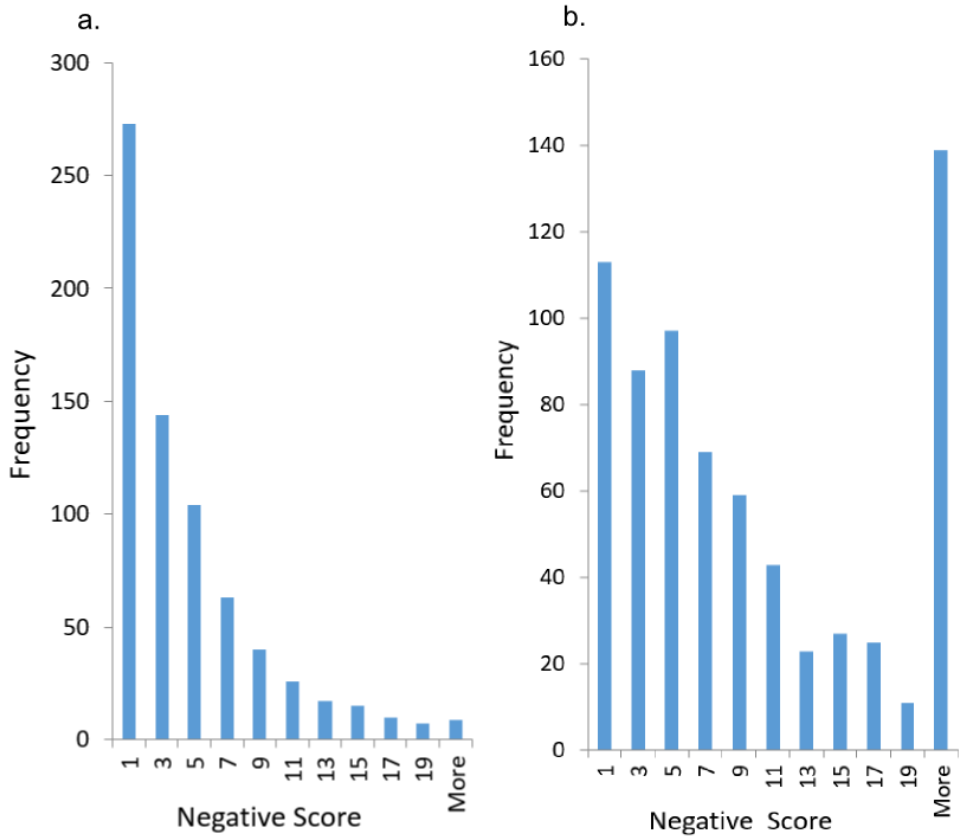

Figure 5.24: a)The negative scores achieved overall test population while using the proposed intent driven constraint recognition method when tested on the 25 assembly cases. b)The negative scores achieved by the geometry dominant constraint recognition alternative when tested on the same data.

proved to succeed in 63.36% of the trials. To verify if the collected data provides significant evidences to support the above observations we ran repeated measures one-way ANOVA tests. The result showed that the proposed technique has a significantly higher rate of success than the geometry dominant constraint recognition methods ($F_{1,29} = 116.7, p < 0.001, \eta^2 = 0.80$).

5.9 Current Limitations and Future Developments

One of the most important limitations of our intention inference algorithms is the maximum wrist speed shown during gestures that can be segmented by our system, which is imposed by the performance of our body tracking method. Specifically, we acquire tracking data at an approximate rate of 30Hz, and we need to use multiple samples[2] to filter the potential detection noise due to tracking uncertainties. Such noise is the cause behind most of the spikes shown by the hand speed profiles rendered in Figure 5.25. In the current set of motion samples, we require 3 frames to indicate that we are reaching the end tail of the descending Gaussian profile of the wrist speed, while the preceding 3 frames also indicate a descending trend of the speed. In other words, the fastest descending Gaussian profile that we identify cannot occur in less than 198ms (6 frames), which limitation is imposed by the current imaging hardware capabilities. In addition, the tracking uncertainties make an analytical evaluation of the lower bound on the speed of the traceable gestures difficult. Due to these reasons we can only provide an empirical lower bound of 1 cm/s obtained through 30 repeated trials.

In the current implementation the size of the smallest feature that can be practically selected is limited to 0.6cm [169] due to the measurement resolution featured by our 3D imaging camera. This hardware limitation could be improved by applying, during the selection procedure, the principles of intent driven motion scaling proposed above for assembly tasks. Thus, the user could use large hand movements to control the fine motion of the hand model in the virtual space. On the other hand, this scaling strategy could easily drive the user outside of his/her natural working space, which is a common problem in practically all human-computer interfaces with several potential solutions being proposed [25]. Another well-known challenge in virtual object manipulation is performing a selection of different geometric features whose

[2]In this work we use information from 4 frames for data filtering.

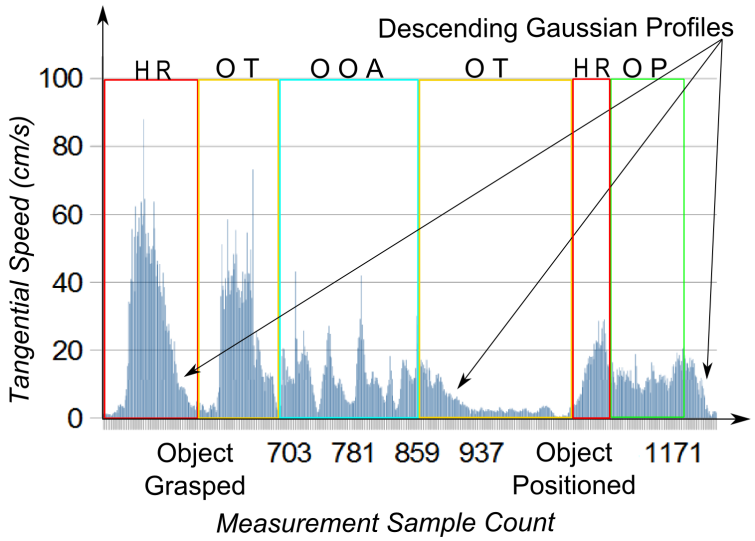

Figure 5.25: The hand speed profiles recorded during the following manipulation task: (HR) hand reaching, object grasping, (OT) object transport,(OOA) object orientation adjustment, object positioning, and finally pushing the object (OP) into the desired position and orientation.

sizes are orders of magnitude apart.

Currently our system allows the user to apply one hand grasp on one object at a time, even for objects that cannot be physically grasped with one hand. For example, we cannot physically grasp with one hand a rigid flat surface that is larger than the area covered by our extended hand. While there are many potential strategies for improving the physical realism of the grasping simulation, the exploration of such strategies is outside of the scope of this manuscript.

Discussion

The empirical data presented in section 5.8.2 shows that tracking faults represent one of the main classes of disturbing factors that affect our gesture recognition methods. These tracking uncertainties mainly occur for parts of the body that are affected by occlusion, imaging noise, low reflectivity, light glare, or body parts that cannot be distinguished from others due to the perceived similarities caused by low imaging resolution or other factors. Given the speed with which the Kinect camera gained popularity, as well as other commercial 3D imaging devices, I expect that in the next few years we will have available on the market 3D imaging systems that offer faster data acquisition rates and higher imaging resolution. Such sensing improvements will alleviate part of the tracking uncertainties we see today. However, regardless of the performance of the employed imaging sensor we will continue to encounter the tracking challenges caused by unavoidable body part occlusions and similarities between body parts. By employing a larger number of imaging sensors that are properly distributed in space we can resolve part of these occlusions. Yet, the self-occlusions that can affect the fingers of our users will continue to pose a tracking challenge. To predict finger motion in the context of partially missing tracking information we can rely on machine learning techniques. As shown in section 5.6, by means of machine learning we can infer a user's intended actions based on behavioral cues and contextual information. Although we achieved promising results, the developed methods serve as a proof of concept and are not aimed to offer an all-inclusive solution to vir-

tual object manipulation problem. In particular, I have not extended these inference techniques to simple manipulation procedures like translating and rotating virtual objects. Therefore, under the influence of tracking uncertainties our system sometimes infers undesired hitting intentions. Another factor that challenges the proposed intention inference techniques is the large variability of our natural hand gestures. This is another reason why the highest rate of confusion of our gesture recognition methods occurs for grasping gestures (86.4% success rate). However, the 25.78% success rate improvement shown by our machine learning technique with respect to the performance of the current state of the art constraint recognition methods indicates that the rate of success of our inference methods increases with the amount of relevant information involved into the inference process. I make this observation based on the outcome of our experimental tests presented in section 5.8.3. Therefore, by embedding additional relevant information into our intention inference process we can further improve the success rate of our gesture recognition techniques.

In order to illustrate the potential of the proposed methods, these intention inference techniques are applied to a case of significant complexity. Namely, while assembling the virtual objects described in section 5.8.3, on each object, we have defined 38 coupling elements that can join in 937 different ways. Inferring the particular manner in which the user intends to assemble these object objects becomes a non trivial challenge in the context of the aforementioned hand placement, body tracking faults and gesture variability (for reasons explained in section 1.1). By adjusting the number and the density of the geometric primitives, or coupling elements, that define the assembled virtual objects the assembly intention inference problem can be morphed in an arbitrary complexity problem. To identify the particular manner in which the user intends to assemble these objects, our probabilistic graphical model (figure 5.7) is conditioned on behavioral cues, geometric parameters and other factors. Note that our CRF model can be used to approximate the relationship between any

quantifiable factor and a user's intent. As shown in [181] such graphical models can be conditioned on complex heterogeneous information. However, their parameters are computationally expensive to estimate and therefore we must choose carefully the associated feature vectors. Part of the used behavioral cues are characterizing general goal directed actions. Therefore, such cues can be applied to automatically identify common types of general intentional actions. Due to the general character of the proposed behavioral cues and the versatile nature of our CRF model, the work presented here lays the foundation of a general intention inference framework.

Conclusions

In this thesis I introduce several new virtual object manipulation methods based on natural hand gestures that do not require the attachment of any hardware on the user's body. Compared to the existing physics dominant methods for virtual object manipulation our techniques show higher robustness to body tracking and hand positioning faults. In addition, our methods manage to handle a wider range of types and variations of natural hand gestures than the symbolic manipulation methods do.

The novelty of our approach stems from the use of characteristic behavioral cues that have been documented in the neuropsychology literature for specific manipulative intentions and gestures. I have shown that these behavioral cues can be successfully used to develop metrics for robustly classifying the observed hand movement into motion primitives corresponding to manipulative gestures. The resulting intent driven manipulation methods are tolerant to hand gesture variability, hand placement imprecision and body tracking uncertainties. In consequence these techniques offer a user friendly framework in which the operator can use natural gestures to perform physically plausible object manipulations. The user studies that have been conducted show that when compared to the physics based interaction methods, our object pushing method affords a faster and more efficient virtual object manipulation. In fact, our methods afford common types of object manipulations which are impractical or at least difficult to achieve while using the physics based manipulation methods.

This thesis introduces a new virtual object selection technique. If compared with the existent selection methods, our approach affords the use of natural hand gestures to select objects whose dimensions are smaller than the tracking resolution of the employed system. The proposed technique offers a seamless selection disambiguation mechanism, which does not require the user to leave the current manipulation context or use symbolic gestures and buttons.

Additionally, the manner in which users intend to assemble virtual objects is estimated by relying on machine learning techniques to evaluate the correlation between the proposed behavioral cues and a user's intent. The resulting technique enables natural virtual object assembly procedures in which every geometric primitive of the manipulated object can be coupled with any other geometric primitive. The performance of the resulting constraint recognition method had been tested by means of user studies. The collected data shows that proposed intent driven constraint recognition method offers a stronger assembly intention disambiguation and a 25.78% higher success rate than the alternative constraint recognition techniques. Furthermore, we have shown that by means of intention inference we can develop interfaces that automatically adapt to the preferences or subjective needs of their users.

However the methods presented in this manuscript serve as a proof of concept and do not offer a comprehensive solution to the virtual object manipulation problem. More specifically, I have not applied our inference techniques to simple manipulation procedures like translating and rotating virtual objects. Therefore, under the influence of the aforementioned tracking uncertainties, sometimes our system infers undesired hitting intentions. Also, the virtual object grasping simulation could be represented in a more realistic fashion. You can find in section 5.9 a discussion on several of the many other problems that remain unsolved.

The work presented here advances the state of the art in 3DUIs towards more user-friendly or even person centered user interfaces by developing user adaptable interfaces

driven by intention inference. This can dramatically shorten the time required by a novice user to start performing efficient virtual object manipulations.

Bibliography

[1] Grigore C Burdea and Philippe Coiffet. *Virtual reality technology*, volume 1. John Wiley & Sons, 2003.

[2] V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):677–695, 1997.

[3] M. Chu and S. Kita. The nature of gestures' beneficial role in spatial problem solving. *Journal of Experimental Psychology: General*, 140(1):102, 2011.

[4] Michelangelo Buonarroti. Creation of Adam. Sistine Chapel, Rome.

[5] E Varga, I Horváth, Z Rusák, JJ Broek, et al. Hand motion processing in applications: A concise survey and analysis of technologies. In *DS 32: Proceedings of DESIGN 2004, the 8th International Design Conference, Dubrovnik, Croatia*, 2004.

[6] A. Erol, G. Bebis, M. Nicolescu, R.D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1-2):52–73, 2007.

[7] N. Osawa and K. Asai. Adjustment and control methods for precise rotation and positioning of virtual object by hand. In *Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry*, pages 131–138. ACM, 2010.

[8] S. Frees. Context-driven interaction in immersive virtual environments. *Virtual reality*, 14(4):277–290, 2010.

[9] R. Kopper, F. Bacim, and D.A. Bowman. Rapid and accurate 3D selection by progressive refinement. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 67–74. IEEE, 2011.

[10] J. Jacobs, M. Stengel, and B. Froehlich. A generalized god-object method for plausible finger-based interactions in virtual environments. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pages 43–51. IEEE, 2012.

[11] Thomas Feix, Roland Pawlik, H Schmiedmayer, Javier Romero, and Danica Kragic. A comprehensive grasp taxonomy. In *Robotics, Science and Systems: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation*, pages 2–3, 2009.

[12] Malinda Carpenter, Nameera Akhtar, and Michael Tomasello. Fourteen-through 18-month-old infants differentially imitate intentional and accidental actions. *Infant Behavior and Development*, 21(2):315–330, 1998.

[13] Elisheva Bonchek-Dokow and Gal A Kaminka. Towards computational models of intention detection and intention prediction. *Cognitive Systems Research*, 28:44–79, 2014.

[14] György Gergely and Gergely Csibra. Teleological reasoning in infancy: The naıve theory of rational action. *Trends in cognitive sciences*, 7(7):287–292, 2003.

[15] Andrew N Meltzoff, Alison Gopnik, and Betty M Repacholi. Toddlers' understanding of intentions, desires and emotions: Explorations of the dark ages. 1999.

[16] Chi-Tai Huang, Cecilia Heyes, and Tony Charman. Infants' behavioral reenactment of" failed attempts": exploring the roles of emulation learning, stimulus enhancement, and understanding of intentions. *Developmental psychology*, 38(5):840, 2002.

[17] T. Flash and B. Hochner. Motor primitives in vertebrates and invertebrates. *Current opinion in neurobiology*, 15(6):660–666, 2005.

[18] M. Santello, M. Flanders, and J.F. Soechting. Patterns of hand motion during grasping and the influence of sensory guidance. *The Journal of Neuroscience*, 22(4):1426–1435, 2002.

[19] S.A. Winges, D.J. Weber, and M. Santello. The role of vision on hand preshaping during reach to grasp. *Experimental Brain Research*, 152(4):489–498, 2003.

[20] S. Glover. Separate visual representations in the planning and control of action. *Behavioral and Brain Sciences*, 27(01):3–24, 2004.

[21] A.P. Sangole and M.F. Levin. Palmar arch dynamics during reach-to-grasp tasks. *Experimental Brain Research*, 190(4):443–452, 2008.

[22] Ferran Argelaguet and Carlos Andujar. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics*, 2012.

[23] D. Holz, S. Ullrich, M. Wolter, T. Kuhlen, and J. Herder. Multi-contact grasp interaction for virtual environments. *Journal of Virtual Reality and Broadcasting*, 5(7):1860–2037, 2008.

[24] M. Moehring and B. Froehlich. Natural interaction metaphors for functional validations of virtual car models. *Visualization and Computer Graphics, IEEE Transactions on*, 17(9):1195–1208, 2011.

[25] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80. ACM, 1996.

[26] M. Moehring and B. Froehlich. Effective manipulation of virtual objects within arm's reach. In *Virtual Reality Conference (VR), 2011 IEEE*, pages 131–138. IEEE, 2011.

[27] Lode Vanacken, Tovi Grossman, and Karin Coninx. Exploring the effects of environment density and target visibility on object selection in 3D virtual environments. In *3D User Interfaces, 2007. 3DUI'07. IEEE Symposium on*. IEEE, 2007.

[28] Anthony Steed. Towards a general model for selection in virtual environments. In *3D User Interfaces, 2006. 3DUI 2006. IEEE Symposium on*, pages 103–110. IEEE, 2006.

[29] Karin Nieuwenhuizen, Lei Liu, Robert van Liere, and Jean-Bernard Martens. Insights from dividing 3d goal-directed movements into meaningful phases. *IEEE computer graphics and applications*, 29(6):44–53, 2009.

[30] Gang Ren and Eamonn O'Neill. 3d selection with freehand gesture. *Computers & Graphics*, 37(3):101–120, 2013.

[31] Jeffrey Cashion, Chadwick Wingrave, and Joseph J LaViola. Dense and dynamic 3d selection for game-based virtual environments. *Visualization and Computer Graphics, IEEE Transactions on*, 18(4):634–642, 2012.

[32] G. De Haan, M. Koutek, and F.H. Post. Intenselect: Using dynamic object rating for assisting 3D object selection. *IPT/EGVE*, pages 201–209, 2005.

[33] Jonathan Wonner, Jérôme Grosjean, Antonio Capobianco, and Dominique Bechmann. Starfish: a selection technique for dense virtual environments. In *Proceedings of the 18th ACM symposium on Virtual reality software and technology*, pages 101–104. ACM, 2012.

[34] S. Frees, G.D. Kessler, and E. Kay. Prism interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer Human Interaction*, 14(1), 2007.

[35] C.W. Borst and A.P. Indugula. A spring model for whole-hand virtual grasping. *Presence: Teleoperators & Virtual Environments*, 15(1):47–61, 2006.

[36] G. Heumer, H.B. Amor, M. Weber, and B. Jung. Grasp recognition with uncalibrated data gloves-a comparison of classification methods. In *Virtual Reality Conference, 2007. VR'07. IEEE*, pages 19–26. IEEE, 2007.

[37] G. Lu, L.K. Shark, G. Hall, and U. Zeshan. Immersive manipulation of virtual objects through glove-based hand gesture interaction. *Virtual Reality*, pages 1–10, 2011.

[38] Microsoft Research. Kinect for windows sdk v1.7, March 2013.

[39] Keyan Liu, Xuyue Yin, Xiumin Fan, and Qichang He. Virtual assembly with physical information: a review. *Assembly Automation*, 35(3):206–220, 2015.

[40] Abhishek Seth, Judy M Vance, and James H Oliver. Virtual reality for assembly methods prototyping: a review. *Virtual reality*, 15(1):5–20, 2011.

[41] Run-dang Yang, Dian-liang Wu, Xiu-min Fan, and Jun-qi YAN. Research on constraint-based virtual assembly technologies. *COMPUTER INTEGRATED MANUFACTURING SYSTEMS-BEIJING-*, 12(3):413, 2006.

[42] Rafael Radkowski and Christian Stritzke. Interactive hand gesture-based assembly for augmented reality applications. In *The Fifth International Conference on Advances in Computer-Human Interactions*, pages 303–308. Citeseer, 2012.

[43] Luis Marcelino, Norman Murray, and Terrence Fernando. A constraint manager to support virtual maintainability. *Computers & Graphics*, 27(1):19–26, 2003.

[44] ZB Wang, SK Ong, and AYC Nee. Augmented reality aided interactive manual assembly design. *The International Journal of Advanced Manufacturing Technology*, 69(5-8):1311–1321, 2013.

[45] Q Yang, DL Wu, HM Zhu, JS Bao, and ZH Wei. Assembly operation process planning by mapping a virtual assembly simulation to real operation. *Computers in Industry*, 64(7):869–879, 2013.

[46] SpaceControl GmbH. http://www.3d-mouse-for-cad.com/, June 2011.

[47] 3dconnexion. http://www.3dconnexion.com/, June 2011.

[48] HCILogitech. http://www.logitech.com/en-us/support-downloads/downloads/design-controllerJune 2011.

[49] Axsotic. http://www.axsotic.com/, June 2011.

[50] Novint. http://home.novint.com/index.php/novintfalcon, June 2011.

[51] Sensable. http://www.sensable.com/products-haptic-devices.htm, June 2011.

[52] Haption. http://www.haption.com/site/eng/html/materiel.php?item=0, June 2011.

[53] 5dt Inc. http://www.5dt.com/hardware.html, June 2011.

[54] CyberGraspSystem User Guide v2.0. http://www.cyberglovesystems.com/sites/default/files/c June 2011.

[55] Measurand. http://www.finger-motion-capture.com/index.html, June 2011.

[56] PixelTech. http://www.pixeltech.fr/anglais/gants_nodnaa.php, June 2011.

[57] Cypress. http://www.cypress.com/?id=1938, June 2011.

[58] Ellipticlabs. http://www.ellipticlabs.com/products/, June 2011.

[59] IDENT Technology AG. http://www.gesture-cube.com/, June 2011.

[60] Henry Holtzman Ramesh Raskar Matthew Hirsch, Douglas Lanman. Bidi screen:a thin,depth-sensing lcd for 3d interaction using lights fields. *ACM Trans. Graph.*

[61] Evoluce. http://www.evoluce.com/de/multitouch/multitouch_lcd.php, June 2011.

[62] Oblong G-Speack. http://oblong.com/, June 2011.

[63] SoftKinetic iisu Product Datasheet V3.5.1. http://www.softkinetic.com, January 2013.

[64] Microsoft Kinect for Windows SDK. http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/, June 2011.

[65] GestPointMaestro3D Toolkit Brochure. http://www.gesturetek.com/pdfs/printer_friendly/ma June 2011.

[66] Omekinteractive. http://www.omekinteractive.com/products.html, June 2011.

[67] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. Kinectfusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *UIST '11*.

[68] Mgestyk. http://www.mgestyk.com/technology.html, June 2011.

[69] The PrimeSensor Reference Design 1.0.8. http://www.primesense.com/files/.

[70] Baumer TZG01 Users Guide for Digital 3D Camera. http://ftp.elvitec.fr/baumer/manuels/usersguide_tzg.pdf, June 2011.

[71] SR4000 Usermanual V2.0 MESA Imaging. http://www.mesa-imaging.ch/dlm.php?fname=customer/customer_cd/sr4000_manual.pdf, June 2011.

[72] PMD vision CamCube 3.0 Datasheet. http://www.pmdtec.com/fileadmin/pmdtec/downloads/ June 2011.

[73] PDFD-Imager Speci

cations Panasonic. http://panasonic-electric-works.net/d-imager/pdf/ekl3104specificationssheet101201.pdf, June 2011.

[74] ZC 1000 TOF Camera datasheet Optex. http://www.optex.co.jp/e/product/pdf/zc-1000.pdf, June 2011.

[75] DS410 Datasheet Softkinetic. http://www.softkinetic.com/portals/0/download/ds410_datashe June 2011.

[76] Leap Motion. https://www.leapmotion.com, January 2015.

[77] B. Curless and S. Seitz. 3D photography. *Course Notes for SIGGRAPH 2000*, 2000.

[78] F. Blais. Review of 20 years of range sensor development. *Journal of Electronic Imaging*, 13(1), 2004.

[79] Bianca Hagebeuker Stephan Hussmann, Thorsten Ringbeck. A performance review of 3d tof vision systems in comparison to stereo vision systems. *Stereo Vision,I-Tech, Vienna,ISBN 978-953-7619-22-0*, page 372, 2008.

[80] A. Nüchter. *3D robotic mapping: the simultaneous localization and mapping problem with six degrees of freedom*, volume 52 of *Springer Tracts in Advanced Robotics*. Springer Verlag, 2009.

[81] G. Yahav, GJ Iddan, and D. Mandelboum. 3D imaging camera for gaming application. In *Consumer Electronics, 2007. ICCE 2007. Digest of Technical Papers. International Conference on*, pages 1–2. IEEE, 2007.

[82] A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-flight cameras in computer graphics. In *Computer Graphics Forum*, volume 29, pages 141–159. Wiley Online Library, 2010.

[83] S. Hussmann, T. Ringbeck, and B. Hagebeuker. A performance review of 3D TOF vision systems in comparison to stereo vision systems. *Stereo Vision*, pages 103–120.

[84] G. Sansoni, M. Trebeschi, and F. Docchio. State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors*, 9(1):568–601, 2009.

[85] J. Salvi, J. Pages, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849, 2004.

[86] N. Karpinsky and S. Zhang. High-resolution, real-time 3d imaging with fringe analysis. *Journal of Real-Time Image Processing*, pages 1–12, 2010.

[87] H. Kawasaki, R. Furukawa, R. Sagawa, and Y. Yagi. Dynamic scene shape reconstruction using a single structured light pattern. 2008.

[88] B. FREEDMAN, A. SHPUNT, M. MACHLINE, and Y. ARIELI. Depth mapping using projected patterns, October 9 2008. WO Patent WO/2008/120,217.

[89] Douglas Lanman and Gabriel Taubin. Build your own 3D scanner: 3D photograhy for beginners. In *SIGGRAPH '09: ACM SIGGRAPH 2009 courses*, pages 1–87, New York, NY, USA, 2009. ACM.

[90] M. Koch and M. Kaehler. Combining 3d laser-scanning and close-range photogrammetry-an approach to exploit the strength of both methods. In *Making History Interactive. Computer Applications and Quantitative Methods in Archeology Conference*, pages 22–26, 2009.

[91] Point Grey Research. Stereo accuracy and error modeling http://www.ptgrey.com/support/kb/data/kbstereoaccuracyshort.pdf, April 2004.

[92] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. 2006.

[93] Christopher Zach, Mario Sormann, and Konrad Karner. High-performance multi-view reconstruction. *3D Data Processing Visualization and Transmission, International Symposium on*, pages 113–120, 2006.

[94] SM Gibson, PA Coe, A. Mitra, DF Howell, and RB Nickerson. Coordinate measurement in 2-d and 3-d geometries using frequency scanning interferometry. *Optics and lasers in engineering*, 43(7):815–831, 2005.

[95] H. Liang, B. Peric, M. Hughes, A.G. Podoleanu, M. Spring, and S. Roehrs. Optical coherence tomography in archaeological and conservation science–a new emerging field. In *Proc. of SPIE Vol*, volume 7139, pages 713915–1.

[96] L. Tian, N. Loomis, J.A. Domínguez-Caballero, and G. Barbastathis. Quantitative measurement of size and three-dimensional position of fast-moving bubbles in air-water mixture flows using digital holography. *Applied optics*, 49(9):1549–1554, 2010.

[97] A. Pelagotti, M. Paturzo, A. Geltrude, M. Locatelli, R. Meucci, P. Poggi, and P. Ferraro. Digital holography for 3d imaging and display in the ir range: challenges and opportunities. *3D Research*, 1(4):1–10, 2010.

[98] M.S. Hrebesh, Y. Watanabe, and M. Sato. Profilometry with compact single-shot low-coherence time-domain interferometry. *Optics Communications*, 281(18):4566–4571, 2008.

[99] U.P. Kumar, B. Bhaduri, MP Kothiyal, and N.K. Mohan. Two-wavelength micro-interferometry for 3-d surface profiling. *Optics and Lasers in Engineering*, 47(2):223–229, 2009.

[100] A.G. Podoleanu. Optical coherence tomography. *British journal of radiology*, 78(935):976, 2005.

[101] A. Bradu, L. Neagu, and A. Podoleanu. Extra long imaging range swept source optical coherence tomography using re-circulation loops. *Optics Express*, 18(24):25361–25370, 2010.

[102] J. Rosen, B. Katz, and G. Brooker. Review of three-dimensional holographic imaging by fresnel incoherent correlation holograms. *3D Research*, 1(1):28–35, 2010.

[103] RM Costa, RA Braga, BS Oliveira, E. Silva, T. Yanagi, and JT Lima. Sensitivity of the moiré technique for measuring biological surfaces. *Biosystems Engineering*, 100(3):321–328, 2008.

[104] W.J. Ryu, Y.J. Kang, S.H. Baik, and S.J. Kang. A study on the 3-d measurement by using digital projection moiré method. *Optik-International Journal for Light and Electron Optics*, 119(10):453–458, 2008.

[105] F. Mohammadi, K. Madanipour, and A.H. Rezaie. Application of digital phase shift moiré to reconstruction of human face. In *UKSim Fourth European Modelling Symposium on Computer Modelling and Simulation*, pages 306–309. IEEE, 2010.

[106] Mayssa Karray Jun-chang Li Jean Michel Desse Pascal Picart Patrice Tankam, Qinghe Song. Real-time three-sensitivity measurements based on three-color digital fresnel holographic interferometry incoherent correlation holograms. *OPTICS LETTERS*, 35(12), 2010.

[107] G. Vogiatzis and C. Hernández. Practical 3d reconstruction based on photometric stereo. *Computer Vision*, pages 313–345, 2010.

[108] Emmanuel Prados and Olivier Faugeras. Shape from shading. In Y. Chen N. Paragios and O. Faugeras, editors, *Handbook of Mathematical Models in Computer Vision*, chapter 23, pages 375–388. Springer, 2006.

[109] G. Lippmann. Épreuves réversibles donnant la sensation du relief. *Journal de Physique Théorique et Appliquée*, 7(1):821–825, 1908.

[110] Fumio Okano, Haruo Hoshino, Jun Arai, and Ichiro Yuyama. Real-time pickup method for a three-dimensional image based on integral photography. *Appl. Opt.*, 36(7):1598–1603, Mar 1997.

[111] B. Tavakoli, M. Daneshpanah, B. Javidi, and E. Watson. Performance of 3d integral imaging with position uncertainty. *Optics Express*, 15(19):11889–11902, 2007.

[112] D. Chaikalis, NP Sgouros, and D. Maroulis. A real-time fpga architecture for 3d reconstruction from integral images. *Journal of Visual Communication and Image Representation*, 21(1):9–16, 2010.

[113] R.R. Sahay and AN Rajagopalan. Dealing with parallax in shape-from-focus. *Image Processing, IEEE Transactions on*, 20(2):558–569, 2011.

[114] R. Minhas, A. Mohammed, Q. Wu, and M. Sid-Ahmed. 3d shape from focus and depth map computation using steerable filters. *Image Analysis and Recognition*, pages 573–583, 2009.

[115] M.B. Ahmad. Focus measure operator using 3d gradient. In *Machine Vision, 2007. ICMV 2007. International Conference on*, pages 18–22. IEEE.

[116] Y. An, G. Kang, I.J. Kim, H.S. Chung, and J. Park. Shape from focus through laplacian using 3d window. In *2008 Second International Conference on Future Generation Communication and Networking*, pages 46–50. IEEE, 2008.

[117] A. Thelen, S. Frey, S. Hirsch, and P. Hering. Improvements in shape-from-focus for holographic reconstructions with regard to focus operators, neighborhood-size, and height value interpolation. *Image Processing, IEEE Transactions on*, 18(1):151–157, 2009.

[118] R.R. Sahay and AN Rajagopalan. Dealing with parallax in shape-from-focus. *Image Processing, IEEE Transactions on*, 20(2):558–569, 2011.

[119] P. Favaro, S. Soatto, M. Burger, and S.J. Osher. Shape from defocus via diffusion. *IEEE transactions on pattern analysis and machine intelligence*, pages 518–531, 2008.

[120] F. Galasso and J. Lasenby. Shape from texture via fourier analysis. *Advances in Visual Computing*, pages 803–814, 2008.

[121] J.T. Todd and L. Thaler. The perception of 3d shape from texture based on directional width gradients. *Journal of vision*, 10(5), 2010.

[122] A. Lobay and DA Forsyth. Shape from texture without boundaries. *International Journal of Computer Vision*, 67(1):71–91, 2006.

[123] A.M. Loh and R. Hartley. Shape from non-homogeneous, non-stationary, anisotropic, perspective texture. In *Proc. of the BMVC*, pages 69–78. Citeseer, 2005.

[124] Y. Sheikh, N. Haering, and M. Shah. Shape from dynamic texture for planes. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2285–2292. IEEE, 2006.

[125] G. S. Cheok, K. S. Saidi, M. Franaszek, J. J. Filliben, and N. Scott. Characterization of the range performance of a 3D imaging system. Technical Report NIST TN - 1695, NIST, 2011.

[126] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. 2006.

[127] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado. A state of the art in structured light patterns for surface profilometry. *Pattern recognition*, 43(8):2666–2680, 2010.

[128] M. Schaffer, M. Grosse, and R. Kowarschik. High-speed pattern projection for three-dimensional shape measurement using laser speckles. *Applied optics*, 49(18):3622–3629, 2010.

[129] S. Foix, G. Alenya, and C. Torras. Lock-in time-of-flight (TOF) cameras: A survey. *Sensors Journal, IEEE*, (99):1–1, 2011.

[130] Yan Cui, Sebastian Schuon, Derek Chan, Sebastian Thrun, and Christian Theobalt. 3D shape scanning with a time-of-flight camera. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1173–1180, 2010.

[131] R. LANGE, P. SEITZ, A. BIBER, and R. SCHWARTE. Time-of-flight range imaging with a custom solid-state image sensor. In *Proceedings of SPIE, the International Society for Optical Engineering*, volume 3823, pages 180–191. Society of Photo-Optical Instrumentation Engineers, 1999.

[132] B. Büttgen, T. Oggier, M. Lehmann, R. Kaufmann, and F. Lustenberger. CCD/CMOS lock-in pixel for range imaging: Challenges, limitations and state-of-the-art. In *Proceedings of the 1st Range Imaging Research Day*, pages 21–32, 2005.

[133] M. Schaffer, M. Grosse, B. Harendt, and R. Kowarschik. High-speed three-dimensional shape measurements of objects with laser speckles and acousto-optical deflection. *Optics Letters*, 36(16):3097–3099, 2011.

[134] X. Su and Q. Zhang. Dynamic 3-d shape measurement method: A review. *Optics and Lasers in Engineering*, 48(2):191–204, 2010.

[135] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, R. Schmidt, and T. Graf. High accuracy stereo vision system for far distance obstacle detection. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 292–297. IEEE, 2004.

[136] A. Criminisi, A. Blake, C. Rother, J. Shotton, and P.H.S. Torr. Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming. *International Journal of Computer Vision*, 71(1):89–110, 2007.

[137] T. Latychevskaia and H.W. Fink. Solution to the twin image problem in holography. *Physical Review Letters*, 98(23):233901, 2007.

[138] L.H. Bieman and K.G. Harding. 3D imaging using a unique refractive optic design to combine moiré and stereo. In *Proceedings of SPIE*, volume 3204, page 2, 1997.

[139] S. Hussmann, T. Ringbeck, and B. Hagebeuker. *Stereo Vision*, chapter A performance review of 3D TOF vision systems in comparison to stereo vision systems, pages 103–120. I-Tech, November 2008. ISBN 978-953-7619-22-0.

[140] Mohit Gupta, Amit Agrawal, Ashok Veeraraghavan, and Srinivasa G. Narasimhan. Structured light 3d scanning in the presence of global illumination. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 713 –720, june 2011.

[141] J. Geng. Structured-light 3d surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2):128–160, 2011.

[142] I. Popov, S. Onuh, and K. Dotchev. Dimensional error analysis in point cloud-based inspection using a non-contact method for data acquisition. *Measurement Science and Technology*, 21:075303, 2010.

[143] J Valença, ENBS Julio, and HJ Araújo. Applications of photogrammetry to structural assessment. *Experimental Techniques*, 36(5):71–81, 2012.

[144] Joris JJ Dirckx, Jan AN Buytaert, and Sam AM Van der Jeught. Implementation of phase-shifting moiré profilometry on a low-cost commercial data projector. *Optics and Lasers in Engineering*, 48(2):244–250, 2010.

[145] R. Lange. *3D time-of-flight Distance Measurement with Custom Solid-state Image Sensors in CMOS/CCD-Technology*. PhD thesis, University of Siegen, 2000.

[146] L.C. Chen, S.L. Yeh, A.M. Tapilouw, and J.C. Chang. 3-d surface profilometry using simultaneous phase-shifting interferometry. *Optics Communications*, 283(18):3376–3382, 2010.

[147] HS Jung, Z. Lu, JS Won, MP Poland, and A. Miklius. Mapping three-dimensional surface deformation by combining multiple-aperture interferometry and conventional interferometry: Application to the june 2007 eruption of kilauea volcano, hawaii. *Geoscience and Remote Sensing Letters, IEEE*, 8(1):34–38, 2011.

[148] S. Mitra and T. Acharya. Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3):311–324, 2007.

[149] Y. Wu and T. Huang. Vision-based gesture recognition: A review. *Gesture-based communication in human-computer interaction*, pages 103–115, 1999.

[150] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 2010.

[151] X. Zabulis, H. Baltzakis, and A. Argyros. Vision-based hand gesture recognition for human-computer interaction. *The Universal Access Handbook, Human Factors and Ergonomics. Lawrence Erlbaum Associates, Inc.(LEA).*

[152] R. Hassanpour, S. Wong, and A. Shahbahrami. Visionbased hand gesture recognition for human computer interaction: A review. In *IADIS International Conference Interfaces and Human Computer Interaction*, page 125. Citeseer, 2008.

[153] P. Garg, N. Aggarwal, and S. Sofat. Vision based hand gesture recognition. *World Academy of Science, Engineering and Technology*, 49:972–977, 2009.

[154] GRS Murthy and RS Jadon. A review of vision based hand gestures recognition. *International Journal of Information Technology*, 2(2):405–410, 2009.

[155] M.S. Del Rose and C.C. Wagner. Survey on classifying human actions through visual sensors. *Artificial Intelligence Review*, pages 1–11, 2011.

[156] H.I. STERN, J.P. WACHS, and Y. EDAN. Designing hand gesture vocabularies for natural interaction by combining psycho-physiological and recognition factors. *Int. J of Semantic Computing. Special Issue on Gesture in Multimodal Systems*, 2008.

[157] J. LaViola. A survey of hand posture and gesture recognition techniques and technology. *Brown University, Providence, RI*, 1999.

[158] M. Zobl, M. Geiger, K. Bengler, and M. Lang. A usability study on hand gesture controlled operation of in-car devices. *Abridged Proceedings, HCI*, pages 5–10, 2001.

[159] F. Althoff, R. Lindl, L. Walchshausl, and S. Hoch. Robust multimodal hand- and head gesture recognition for controlling automotive infotainment systems. *VDI BERICHTE*, 1919:187, 2005.

[160] Sreeram Sreedharan, Edmund S. Zurita, and Beryl Plimmer. 3D input for 3D worlds. In *Proceedings of the 19th Australasian conference on Computer-Human Interaction: Entertaining User Interfaces*, OZCHI '07, pages 227–230, New York, NY, USA, 2007. ACM.

[161] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. IEEE, 2011.

[162] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. *BMVC, Aug*, 2, 2011.

[163] G. Vogiatzis and C. Hernández. Practical 3D reconstruction based on photometric stereo. *Computer Vision*, pages 313–345, 2010.

[164] A.M. Loh and R. Hartley. Shape from non-homogeneous, non-stationary, anisotropic, perspective texture. In *Proc. of the BMVC*, pages 69–78. Citeseer, 2005.

[165] J. Maycock, B. Bläsing, T. Bockemühl, H. Ritter, and T. Schack. Motor synergies and object representations in virtual and real grasping. In *1st International Conference on Applied Bionics and Biomechanics (ICABB)*, 2010.

[166] Kenneth F Valyear, Craig S Chapman, Jason P Gallivan, Robert S Mark, and Jody C Culham. To use or to move: goal-set modulates priming when grasping real tools. *Experimental brain research*, 212(1):125–142, 2011.

[167] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

[168] A.P. Sangole and M.F. Levin. A new perspective in the understanding of hand dysfunction following neurological injury. *Topics in stroke rehabilitation*, 14(3):80–94, 2007.

[169] F. Periverzov and H.T. Ilieş. IDS: The intent driven selection method for natural user interfaces. In *3D User Interfaces (3DUI), 2015 IEEE Symposium on*, pages –. IEEE, 2015.

[170] Brian D Ziebart, Andrew L Maas, Anind K Dey, and J Andrew Bagnell. Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 322–331. ACM, 2008.

[171] Dizan Vasquez, Thierry Fraichard, Olivier Aycard, and Christian Laugier. Intentional motion on-line learning and prediction. *Machine Vision and Applications*, 19(5-6):411–425, 2008.

[172] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[173] Lin Liao, Donald J Patterson, Dieter Fox, and Henry Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5):311–331, 2007.

[174] Dirk Gehrig, Peter Krauthausen, Lukas Rybok, Hildegard Kuehne, Uwe D Hanebeck, Tanja Schultz, and Rainer Stiefelhagen. Combined intention, activity, and motion recognition for a humanoid household robot. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4819–4825. IEEE, 2011.

[175] Sungyoung Lee, Hung Xuan Le, Hung Quoc Ngo, Hyoung Il Kim, Manhyung Han, Young-Koo Lee, et al. Semi-markov conditional random fields for accelerometer-based activity recognition. *Applied Intelligence*, 35(2):226–241, 2011.

[176] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

[177] L Morency, Ariadna Quattoni, and Trevor Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[178] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *Communications Surveys & Tutorials, IEEE*, 15(3):1192–1209, 2013.

[179] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

[180] Andrew Makhorin. Gnu linear programming kit, July 2016.

[181] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.