Design of an immersive peripheral for object grasping

Jean-Rémy Chardonnet, Jean-Claude Léon

To cite this version:

Jean-Rémy Chardonnet, Jean-Claude Léon. Design of an immersive peripheral for object grasping. DETC2010 - ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Aug 2010, Montréal, Canada. pp.1335-1344, 10.1115/DETC2010-28416 . inria-00503126

HAL Id: inria-00503126

https://hal.inria.fr/inria-00503126

Submitted on 16 Aug 2010

Proceedings of the ASME 2010 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2010
August 15-18, 2010, Montreal, Quebec, Canada

DETC2010-28416

DESIGN OF AN IMMERSIVE PERIPHERAL FOR OBJECT GRASPING

Jean-Rémy Chardonnet
Jean Kuntzmann Laboratory, INRIA Rhône-Alpes,
655 avenue de l'Europe - Montbonnot
38334 Saint-Ismier Cedex France
Tel: (33) 04 76 61 55 52
jean-remy.chardonnet@inrialpes.fr

Jean-Claude Léon
G-Scop Laboratory, Grenoble INP,
Domaine Universitaire BP 53
38041 Grenoble Cedex 9 France
Tel: (33) 04 76 82 51 27
jean-claude.leon@grenoble-inp.fr

ABSTRACT

During product development processes, simulations involving user's grasping operations are of increasing interest to incorporate more quantitative information in DFA (Design For Assembly) or immersive simulations. We present several prototypes of an immersive peripheral device for controlling a virtual hand with fine dexterity. These prototypes are derived from the analysis of a grasping action to define the structure and main features of this device. The prototypes, as easy to manipulate as a computer mouse, enable the simultaneous control of a large number of degrees of freedom (dofs). The design issues, where physical phenomena, physiological behavior and device structure are all tightly combined and significantly influence the overall interaction, are reviewed. These issues include the generation of dofs, monitoring kinematics, force reduction during virtual hand and finger movements, and the influence of device design, sensor types and their placement on the interaction and on the range of configurations that can be achieved for grasping tasks, dexterity, and performance. Examples of grasping tasks show the effect of these immersive devices to reach user-friendly and efficient interactions with objects bringing new insight to the interaction with virtual products.

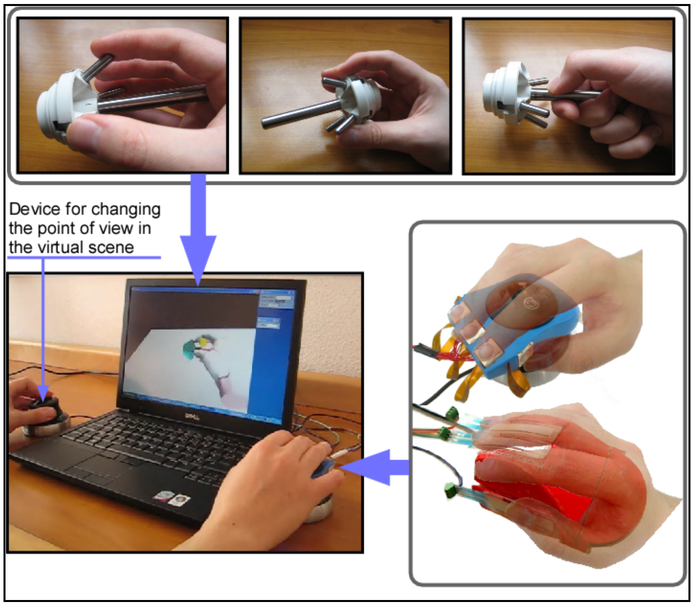


Figure 1: Example of manipulation tasks that can be achieved with the HandNavigator, an immersive peripheral device for grasping virtual objects. Several prototypes using different sensors' technologies are presented.

1. INTRODUCTION

Object manipulation, and more specifically grasping, is an everyday life's task that we perform unconsciously using our hands. However imitating grasping in virtual worlds using artificial tools still remains a great challenge. Multiple solutions exist to interact with virtual worlds, that could be used to control a virtual hand or even a robotic hand. Controlling hand tasks with these devices faces the critical question of user's dexterity and user's comfort. This paper extends and generalizes the HandNavigator, a device specifically designed for hands-on interaction in virtual environments [1]. To overcome the weaknesses of the original device, we set up a comparative study of several novel prototypes, where the physical phenomena used in sensors, the kinematic structure of the device and its shape evolved in order to improve ergonomics, usability, dexterity and user's performance.

1.1 Previous work

Grasping objects is a difficult task to analyze since there are many ways to perform it. Indeed, depending on the properties of the objects, e.g., shapes, size, material, the way we move our hand and fingers for grasping differ. But for the same object, there can be several possible configurations (see Fig. 1

and Fig. 12). Imitating grasping movements in virtual worlds is hence not trivial as there are a large number of possible configurations. However, this diversity is currently not covered by the existing tools. One of the applications that need grasping tasks in virtual reality is DFA [2]. Nowadays, DFA software allows a user to structure a grasping operation but they only set qualitative parameters. Indeed, every hand movement or every parameter cannot be controlled in virtual worlds as it highly increases the complexity of interfaces to be designed, but natural configurations of the hand can be achieved if the interface gives enough immersive sensations. However, this achievement is tightly linked to the design of an appropriate peripheral. Relying on a hand movement peripheral is a mean to evolve from qualitative assessment of grasping to a quantitative one.

Many solutions were proposed to achieve general manipulation tasks in virtual environments, transforming real hand movements into virtual ones [3]. The corresponding solutions are often based on optical motion capture systems (for example through cameras) [4, 5], or on mechanical motion capture systems (for example data gloves where sensors are associated to an exoskeleton touching the hand) [6]. These solutions produce a large panel of hand configurations, enabling many possibilities in moving the hand around objects and, subsequently, when grasping them. However, they have several important drawbacks. Calibration must be performed at the beginning of each use, requiring a good knowledge of the device and a significant setup time. Also, they can generate a wide diversity of movements, which increases their mechanical complexity and their cost. As stated just earlier: not only their calibration but also their ease of use can be difficult. Thus, these systems are not ready-to-use at a moment’s notice. Vision-based systems face occlusion problems. Indeed, there is often a point that is not visible by a camera, for example if the user's fingers or hand are hiding other fingers, or if the fingers are hidden because the hand is closed. To solve these problems, several cameras can be added. However, the cost and space required for the setup greatly increase, while there are still configurations for which occlusion problems remain. From an ergonomic point of view, an important issue is that long tasks with a "raised hand" as required for motion capture quickly lead to a muscular tiredness.

Some solutions additionally integrate haptic feedback [7, 8]. This feedback can also be found in larger devices including force feedback, such as haptic devices that permit interaction with physical simulations (see, for example, [9, 10]). These solutions enhance the level of information returned to the user and thus better take advantage of the brain sensorial capabilities. Haptic feedback is perceived only when a user is colliding with a virtual object but lacks entirely or so when moving in the air without any contact [11, 12]. Active haptic systems get more complex if the number of returned force components increases (many devices provide just two, three, or six returned force components). These systems allow a user to better perceive objects in virtual worlds, but are rarely used by ordinary people. The main reasons are the high cost of such systems and, as such, they are only found in companies or academic labs. One explanation is that these systems have a technological complexity that requires a good knowledge of the device and its limitations since tuning of parameters is necessary to implement a reasonable sensation of physical models, and such a tuning is generally subjective.

A last solution is to offer some kind of passive feedback. This feedback can be perceived as an improvement compared to systems without feedback [13], and can fool the proprioceptive senses of the user [14] when using a proxy such as a sponge, a small ball [12, 15]. The benefit of such solutions is their low cost and ability to be integrated in classical computer devices.

1.2 Overview

In contrast with previous works, our solutions rely on passive feedback and are devoted to hands-on interaction. We extend the HandNavigator device presented in [1]. Our focus is on the design of improved solutions, and the dependencies between the model of interaction, the device structure, sensors, ergonomics, and dexterity, which were not targeted in [1]. Validation tests involving users are a subsequent problem which we leave for future work.

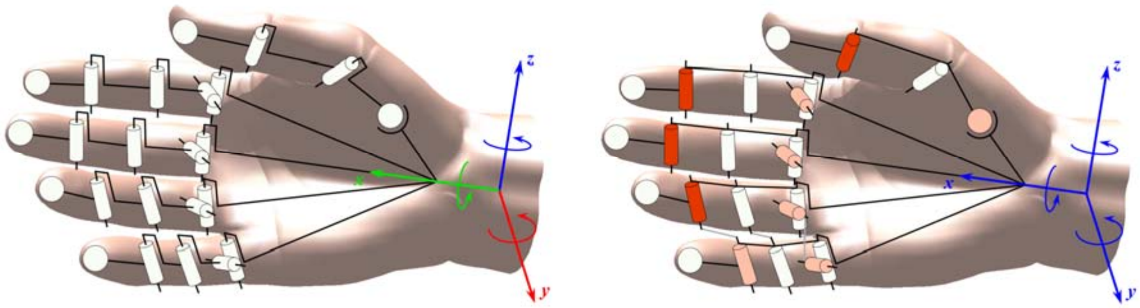
The paper is structured as follows. In the next section, we characterize the diversity of hand configurations leading to dexterity and analyze grasping actions. The problematics appearing for such a device associated with its kinematic structure, and more specifically connection between user’s interactions and virtual hand kinematics will be explained in Section 3. Then, interactions between shapes and sensors for the design of several prototypes will be addressed in Section 4. We will show some results and applications in Section 5 before concluding.

2. GRASPING ACTIONS ANALYSIS

The kinematic structure of our hand allows us a large diversity of configurations, among which we choose the best one to grasp objects. This ability to adapt our hand configurations to objects increases the dexterity we can achieve. Dexterity can be defined here as the coordination of the hands and fingers with the eyes, implying the ability of using the hands and fingers to perform rather precise activities. Therefore, tactile and visual feedbacks are two modalities to achieve dexterity [16]. More precisely, the concept of dexterity is reduced here to a set of tasks we want to be able to perform with our device. They can be summarized as:

- the independent or group movements of each virtual finger in the air, as naturally as possible;
- grasping rigid objects as easily as possible.

We want to address kinematic issues to control independently the position and orientation of the hand, and the motion of the fingers. This simultaneous mobility of hands and fingers are needed to achieve a large panel of tasks when the user interacts, i.e., applies forces, on the device.



(a) Degrees of freedom of the virtual hand: 6 for the wrist and 4 for each finger (2 for the first phalanx and 2 revolute joints for the two other phalanxes). The thumb has 5 degrees of freedom (3 for the first phalanx). Note that the kinematics of the virtual hand does not exactly correspond to the one of the real hand.

(b) Degrees of freedom controlled by the device. In blue, the degrees controlled by a navigation device. In dark orange, the degrees controlled by the sensors. In light orange, the degrees that can be controlled. The strokes in light gray between the ring and the pinky fingers mean that it is possible to couple these two fingers.

Figure 2: Kinematics of the virtual hand

When analyzing grasping, we propose to decompose it into three steps:

1) user's hand and fingers move freely in 3D space to approach the object to grasp. During this phase, the user's already adapts his hand's configuration so that it fits the object's shape;
2) user's hand and fingers touches the object to be grasped. These contacts may occur sequentially (the hand moves and then the fingers touches simultaneously or sequentially the object) or progressively (the hand is moving while closing the fingers);
3) user applies pressure on the object to tighten it. This pressure depends on the friction between the object and the user's hand and fingers. Typically, low friction on objects such as oiled objects, requires very strong pressures so that these objects do not slip. During this phase, fingers configurations change, generating different hand postures.

Performing these steps requires many dofs. In virtual worlds, some of these dofs can be configured using inverse kinematics (see Section 3) to cover a range of postures and while reducing the number of dofs to monitor on the device. Monitoring simultaneously a large number of dofs is needed to perform dexterous tasks. For instance, with existing peripherals like mouse or space mouse, placing and configuring virtual manikins are difficult to achieve because the user has access to a small number of dofs compared to the number of dofs of a whole manikin. Often, the user ends up with iterative, hence tedious adjustments to reach realistic and natural postures.

Considering these issues, the peripheral device we want to develop should take into account the following aspects: (i) enable the simultaneous control of a large number of dofs, (ii) allow a user to perform complex motions, (iii) address ergonomic issues to avoid generating any muscular pain or tiredness that dramatically reduces motion dexterity and does not enable intensive use (typically ranging from several minutes to hours), and (iv) be cheap, easy and ready to use, and calibration free to ease integration with standard computer devices.

3. PERIPHERAL STRUCTURE AND INTERACTION ANALYSIS

When designing a peripheral device that takes into account the above aspects, two important problems must be addressed:

- on the one hand, the way dofs are generated to control the virtual hand and its fingers;
- on the other hand, the analysis of the user's interaction with the device and its consequences on the motion control of both the hand and the virtual fingers to avoid cross effects between each other. This is needed to increase the range of tasks performed by the virtual hand while preserving a precise kinematic control to improve the user's dexterity.

Furthermore, the type of peripheral devices must be considered as their choice is critical and closely linked to its objectives. A peripheral device can be classified into the following categories [17]:

- isotonic: the motion of the effector is free and can be achieved with a null or nearly null resistance;
- isometric: the motion of the effector is constrained and the force applied on the effector is measured;
- elastic: the effector is not fixed and the resistance on the effector increases with the displacement.

However, this classification must be used with care, as it depends on the point of view from which the device is considered. Indeed, a classification linked with the user's perception could be interpreted as purely subjective, whereas a classification linked with physical (or mechanical) properties or physiological concepts is more objective. Here, we will always seek which category of peripheral device best fits the objectives, in terms of performance and desired tasks.

Figure 3: The SpaceNavigator and its six dofs.

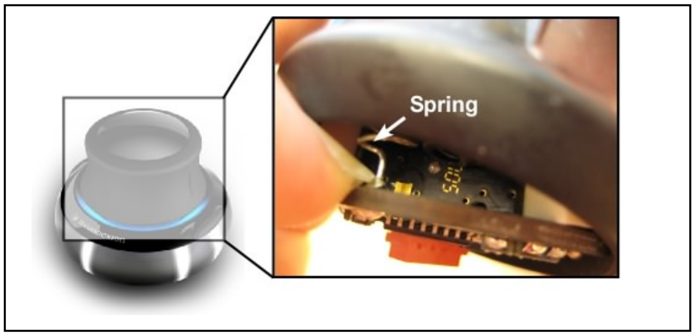


Figure 4: Inside the SpaceNavigator.

3.1 Generating the degrees of freedom

The virtual hand setup has 27 dofs (see Fig. 2(a)). This implies the treatment of a large amount of data. In our case, we will constrain some dofs to simplify the device in terms of data flow acquisition. More precisely, some phalanxes can be constrained by the kinematics of the virtual hand, so that only the end part of the fingers will be controlled (see Fig. 2(b)). Note that in our case, we ensure the uniqueness of finger configurations to avoid unexpected virtual motions of the fingers using an inverse kinematics algorithm with joint limit constraints, which is not necessarily the case for other approaches like finger motion of data gloves.

The position and orientation control of the virtual hand corresponding to the motion of the wrist can be achieved through a navigation device. In a first place, we use for this purpose the SpaceNavigator from 3dConnexion as it is a cheap and widely commercialized device, compared to other devices such as accelerometers, and meets some of our needs in terms of calibration and integration in desktop environments. This device is a velocity-controlled device consisting in two main parts (see Fig. 3):

- a heavy base avoiding the user to move the device while using it;
- a moving part mounted on the base with which the user interacts to generate movements in accordance to the dofs.

It allows the user to control the six dofs of an object position in 3D space. Springs located inside the device are deformed when a user acts on its moving part (see Fig. 4), the corresponding strains are measured and then converted into a velocity along the 6 dofs. If we consider again the classification of [17], the SpaceNavigator has an elastic behavior because of its mechanical properties.

Most of the fingers joints are modeled by revolute joints, i.e., by a 1-dof joint. Consequently, to control these dofs, we can use elementary sensors that give only one physical value. We will give a brief description of these sensors in Section 4.

The interest of having such a number of dofs is not trivial. For very simple grasping tasks such as basic pick-and-place, at least one grasping point is enough and, in this case, a simple computer mouse can be sufficient to achieve this task. Conversely, if the user wants to generate more complex tasks such as assembling several parts, as a human does with his real hand, e.g., twisting an object grabbed with two fingers needs at least two or three contact points and cannot be achieved easily with classical interfaces because the user cannot control precisely these points, whereas with our device we ensure the user to achieve such dexterous manipulation tasks, as if it was for real.

3.2 Kinematic monitoring

Combining a navigation device with sensors can create interferences between each other. Indeed, we want to control the position and orientation of the hand and the motion of the fingers without side effects between of each other, i.e., the hand moves while the user wants only a movement of some fingers as needed for dexterous motion. Because the SpaceNavigator is elastic, any force/moment applied to it by the user produces a hand displacement in virtual space. Also, to move fingers, the user has to apply forces to the corresponding sensors. These forces, even small, may interact with those needed for the hand movement, which explains the origin of perturbation between hand and finger movements. As a typical example, if the hand stands still, no force is applied to the SpaceNavigator, and if one finger has to be moved, the force applied to the sensor will move also the wrist, i.e., it perturbates the wrist motion. In this case, the force needed to move the finger has to be compensated in some ways to avoid perturbations.

It has to be noticed that these forces take place in all three steps of a grasping action (see Section 2). Particularly in step 1, forces have to be kept as low as possible to reduce user’s tiredness and help distinguishing step 1 from steps 2 and 3.

The structure of the device separates dofs because it is structured into two independent elements: the navigation part on the one side and the sensors for the fingers on the other side (see Fig. 5). In other words, the way the user interacts with the sensors as well as their technology can influence the behavior of the navigation device. To be able to achieve independent hand and finger movements depends on:

- the shape of the peripheral device, which will also improve the user's comfort;
- the types of sensors, especially their mechanical properties since forces are key parameters of the device behavior.

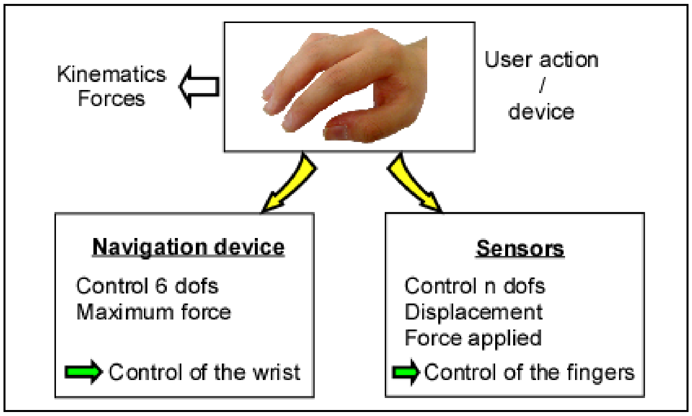


Figure 5: Control of the hand.

Consequently, the way forces act on the SpaceNavigator should stay independent of the forces generated to monitor the position of the fingers to reach a wider range of hand configurations and increase the dexterity. Also, resulting force configurations should be as intuitive as possible with respect to desired hand postures. In other words, one of the modality to achieve good decoupling is the intensity of forces the user has to apply on the device. Indeed, if a user has to apply on the sensors a force larger than the one of the threshold on the navigation device, it will result in an undesired motion of the hand. The consequences in terms of physiology are a contraction at the user's muscles, i.e. to avoid this motion bias the user has to compensate the unbalanced force with phalanxes or palm contacts. Hence, difficulties appear to move easily the real hand as well as a substantial fatigue for long tasks. In Fig. 6, we show cases of force compensation with one or two fingers supposed to act over pressure sensors. If sensors are uniformly distributed around the SpaceNavigator and all fingers are moving in the same manner (all fingers closing or opening), finger forces F_{fc} may compensate each other and thus no perturbation will be generated on the moving part of the SpaceNavigator (see Fig. 6a.). As a result, the position of the virtual hand can stand still while moving the fingers, which conforms to the desired behavior and stays intuitive for the user. However, if sensors are located on top of the SpaceNavigator, if a user wants to close the fingers while moving the hand frontward, he will have to compensate the vertical forces on the sensors, e.g., with lateral forces high enough and adherence phenomenon, to keep the virtual hand moving frontward without going downward, which requires muscular strength either from the palm or other phalanxes to achieve the compensation (see Fig. 6b.).

The smaller the compensation forces, the higher the motion dexterity. We can distinguish two types of dexterity: dexterity in reaching desired virtual hand configurations (step 1 in Section 2) and dexterity when touching an object (step 2 in Section 2). The first dexterity derives from the previous paragraph. The second one has a stronger link to visual and tactile feedbacks because the user will use his proprioceptive senses to perform manipulation tasks, as mentioned earlier.

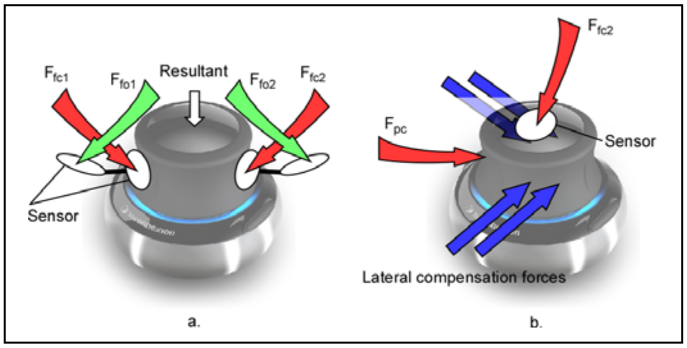


Figure 6: Force compensation (here with two fingers). In red: closing pressure (F_{fc} for the fingers and F_{pc} for the palm), in green: opening pressure (F_{fo} for the fingers), in blue: compensation forces. a: opposite distribution of forces when opening/closing simultaneously two fingers; no compensation needed. b: if the sensor is on the top of the device, it is necessary to apply strong lateral forces to compensate F_{fc2} and maintain the desired wrist configuration.

Therefore, the application software should include helpers such as shadows and markers.

The dexterity we are looking for should allow a user to perform tasks that are rather simple, such as grasping and manipulating rigid objects, but also more complex tasks, such as shape deformations, in virtual hand configurations that are as close as possible to those of a real hand. Traditional devices such as a mouse or simple buttons cannot cope with these tasks.

Note that the compensation mechanism can be achieved through software, however it needs parameter tuning and more sophisticated algorithms whereas the current work is a first level of prototype design.

4. DESIGN IMPROVEMENTS USING PROTOTYPES

4.1 Existing prototype

Here we address improvements of the device presented in [1]. This device consists in the SpaceNavigator presented earlier toe position and orient a virtual hand, enhanced with sensors to control the virtual fingers. As shown in Fig. 7, these sensors are fixed on lightweight metallic petals with a low stiffness (a few N.mm^{-1}). Note that this stiffness acts as a passive feedback. However, if we consider that proprioceptive effects are not critical, this stiffness is not of any interest for the device in terms of sensing. Therefore, another interest of this stiffness is to help the user compensating the forces applied on the SpaceNavigator when pressing the sensors. Each finger is controlled separately in velocity: the velocity of the fingers open-close motion is a function of the pressure applied on the sensors. Thus, when the user does not press any sensor, the virtual fingers do not move. We can see one major benefit of this device compared to other solutions such as data gloves: the user can interrupt a task anytime to start another task such as modifying scene parameters of an application or comparing

Figure 7: First versions of the prototypes.

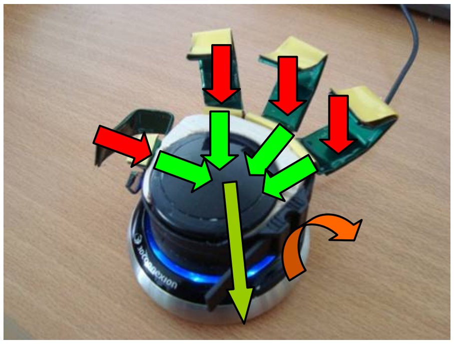
with a variant of scenario, and resume the virtual task without losing any information.

The main drawback of this device is its poor ergonomics. Indeed, the user must bend his wrist with a high angle, leading to an uncomfortable posture of the arm and the hand (the arm is in a raised position) and hence, to motions of the hand and the fingers that are difficult to perform because of a substantial contraction of the wrist. Moreover, the radial distribution of the petals is not natural for a real hand, which adds tiredness and difficulty to move the fingers. The advantage of this distribution holds in a rather homogeneous force distribution on the SpaceNavigator since the petals are spread uniformly around the moving part and the configuration of applied forces is mostly centripetal. This can produce a low resulting force and thus, generates low interaction with the forces applied to the SpaceNavigator.

4.2 Shape design

The observations made on the device presented in [1] show that it is critical to generate user's hand postures where wrist, hand and fingers contraction is kept low so that the user can monitor the device more accurately. To improve user's comfort, dexterity and performance for grasping tasks, shape design must be considered.

From the shape of the first versions, metallic petals can be redistributed so that it follows a natural distribution of the fingers (see Fig. 8(a)). Thus, it is no longer necessary to bend the wrist as it can be hold easily, avoiding any muscular contraction. However, when the user presses the sensors, this can generate a momentum high enough to create a non-negligible perturbation on the SpaceNavigator that must be canceled, hence generating a hardly avoidable and undesired motion (see Fig. 8(a)). Indeed, because of the sensors' low sensitivity, the user has to apply forces that are much higher than that needed to generate a motion of the navigation device. It is typically the case where separating the dofs of the hand from those of the fingers. Consequently, we did not stick to this shape anymore.



(a) Natural distribution of the petals. The red arrows are the forces the user applies on the sensors. It generates a torque (in orange). The four light green arrows are the forces applied on the device, generating a resulting force (in green) that is hard to compensate

(b) Computer mouse shape. The arrows show where the user can apply forces to help force compensation.

Figure 8: Shape design.

To handle the problems generated by the previous shape, we considered the shape of a computer mouse: it solves the problems of the wrist postures and those of interaction between the fingers and the nails (see Fig. 8(b)). This shape is also well known by most people and thus can be easily approached. It is possible to easily compensate the forces generated when pressing the sensors by using the palm on the backside of the device (see Fig. 8(b)). We kept this shape for the future prototypes.

We clearly see that the position of the sensor is important in the design of the HandNavigator. Especially, with the first versions, motion control difficulties can be faced when performing the transition between opening and closing motions of a finger as it is linked to two different sensors whereas a natural finger motion is achieved through a change of motion orientation.

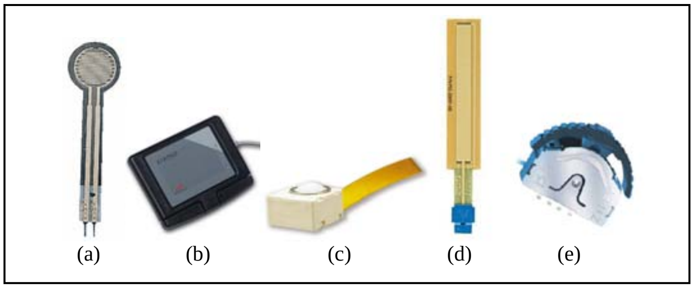


Figure 9: Different sensors' technologies. From left to right: pressure sensors, tactile pad, trackball, scrollpad, single-axis switch.

4.3 Sensor's technology

Sensor's technology can also highly influence the overall dexterity as the forces to be applied by the user on the sensor are not neglectible. We tested several technologies of existing sensors that can meet our requirements:

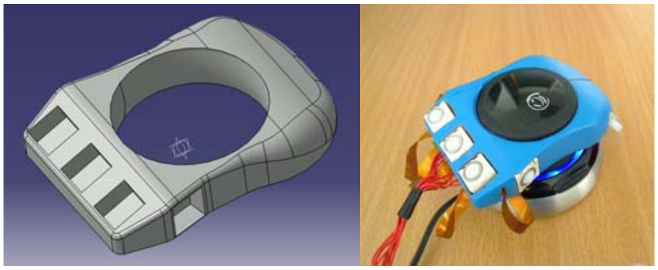
- the user must not feel any pain or tiredness to move the virtual fingers;
- interference with the navigation device must be as low as possible;
- the sensor must be small enough, lightweight and sensitive to avoid generating undesired motions;
- passive feedback can be integrated to help the user achieve more dexterous motions.

Considering these constraints, we tested the sensors depicted in Fig. 9.

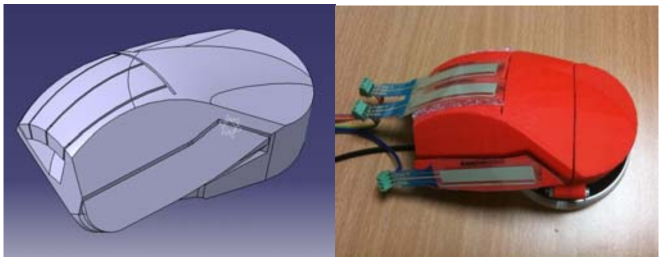
As mentioned in Section 3.2, dexterity can be achieved if the user can get desired configurations of the virtual hand, which implies to apply forces on sensors that are smaller than those on the navigation device. It is obvious that whatever technology we use for controlling the virtual fingers, forces have to be applied. The aim here is to find one or several technologies of sensors that minimize the force intensity to apply.

In the first versions, pressure sensors were used. Because of their mechanical design, they are adapted to velocity-based control, enabling to interrupt and resume anytime a task without losing the virtual hand's configuration, which is, as mentioned earlier, one of our objectives. However, to activate these sensors, a force of more than 1N is necessary, which is high compared to the minimal force applied to the SpaceNavigator to move the virtual hand (0.4N), implying user generated undesired motions of the virtual hand and causing substantial fatigue at the user's wrist.

Compared to pressure sensors, touch-pads and scroll-pads (see Fig. 9(b) and (d)) offer better characteristics in terms of force threshold. Indeed, a light touch (0.6N for scroll-pads) generates a signal, reducing the interferences as stated in Section 3.2 as well as tiredness. Note that the touch-pads can take two dofs, whereas scroll-pads are designed for one dof. However, these sensors do not integrate any passive feedback, unlike trackballs and single-axis switches (see Fig. 9(c) and (e)).



(a) The V3trackball prototype integrates trackballs.



(b) The V3scrollpad prototype integrates scrollpads.

Figure 10: Versions V3trackball and V3scrollpad of the device with CAD designs.

Trackballs are interesting with only 0.35N to activate them, meaning that the motion of the virtual fingers can be achieved without disturbing the one of the hand.

Through this analysis, we see that the dexterity the user will be able to achieve in the virtual environment depends on the sensor technology. Especially, the sensor sensitivity and its capability to return relevant information to the user are important. Among all the technologies, tactile pad, trackball, scroll-pad, single-axis switch can achieve opening and closing motions through reverse motion on sensor, which is a metaphor similar to the finger movement and hence more natural for the user.

4.4 New peripherial device

Based on these ergonomic and technological issues, we designed several prototypes using various types of sensors and called V3pressure (see Fig. 8(b)), V3trackball, and V3scrollpad (see Fig. 10), respectively. We chose a mouse-like shape for each prototype to get modularity with different sensor technologies as well as an easy integration into several applications.

Similarly to the initial prototypes, the V3pressure prototype integrates pressure sensors. Interaction with the sensors is much easier than with the initial prototypes. A rod helps the user compensate the forces applied to pressure sensors together with the user's palm, as mentioned in Section 4.2. However, latency problems of transition between the finger's open/close motions still hold because of two different sensors per finger. This version operates with a velocity-based control of the fingers, which meets the initial requirements of the HandNavigator.

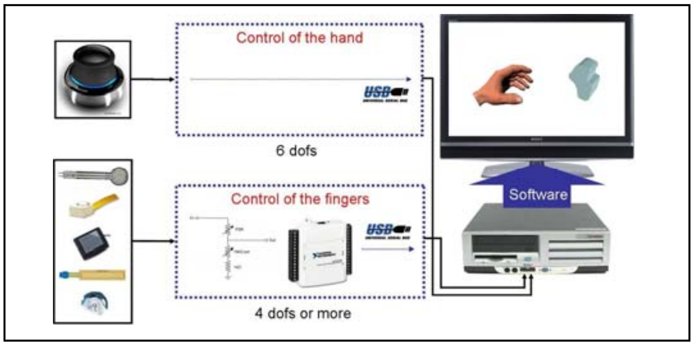


Figure 11: General outline of the Hand Navigator.

The V3trackball prototype integrates trackballs. The small size of these devices leads to an overall small size of the prototype and, as for the initial prototypes, the pinky and the ring fingers are kinematically linked.

The V3scrollpad prototype incorporates scroll-pads. Thanks to the scroll-pads' sensitivity, the user does not need to apply high intensity forces on the device to decouple the hand and fingers's movements. Moreover, the trajectory of the finger on the sensor corresponds to the natural trajectory of the finger when bending, i.e., a near rectilinear trajectory. Note that the shape of this version is different from the one of the V3trackball version to allow the user a more natural posture of his hand and fingers on the device, as well as being more suited to different hand sizes, thanks to the length of the sensors (50mm).

For these two versions, we tried to produce very simple and easy-to-use devices, requiring low forces to move the virtual fingers because the sensors are sensitive. Thus, the SpaceNavigator is better decoupled from the HandNavigator, which improves the dexterity of the virtual hand and fingers. Moreover, it is possible to incorporate either a position-based or a velocity-based control. With a position-based control, it is still possible to interrupt a task because the position returned by the sensor and the finger position stands still right away when idle. Using only one sensor for both opening and closing motions enables a smooth transition producing a better dexterity in object grasping tasks and finger movements in the air, as desired in the objectives.

Each prototype version has an isometric behavior getting close to an isotonic one because the user applies very low forces and can perform large movements with his real fingers, i.e., around 15mm, in order to generate large virtual movements. This property of the device to be at the boundary between isometric and isotonic categories seems to best meet the objectives because it produces a good kinematic decoupling of the SpaceNavigator, which enables dexterous movements.

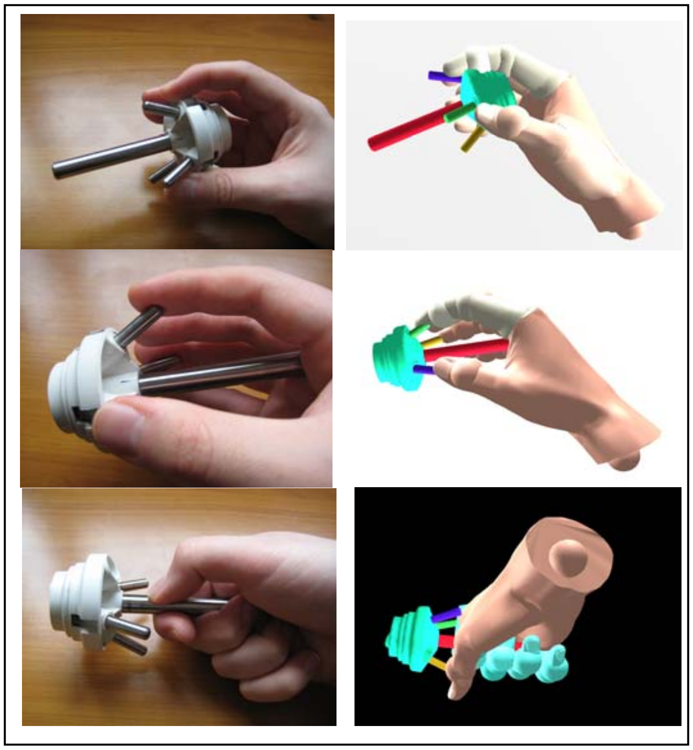


Figure 12: Examples of virtual grasping configurations corresponding to those in real situations.

Finally, the V3trackball was manufactured with rapid prototyping processes of type 'stratoconception'. This technology allows us to get rigid components to test the real use of the device but sets constraints on the thickness of shell areas because of cutter forces applied during the material removal process. This manufacturing process is also able to process lightweight material, which is also important to minimize the mass of the prototype and keep it within a couple of tens of grams so that it does not influence the Space Navigator. In contrast, the V3scrollpad was manufactured using a 3D printer. This technology allows us to get complex shaped components that cannot be easily produced using the cutter diameter constraint of the 'stratoconception'. Material density is harder to monitor and care must be taken about the mass of the prototype through shape and thickness to meet a goal of nearly 50 grams. In both cases, the centre of gravity must be kept close to the axis of the Space Navigator to avoid moments interfering with the Space Navigator.

The general outline of the HandNavigator is depicted in Fig. 11.

5. EXAMPLES OF GRASPING ACTIONS

Here, we show different examples of grasping configurations using the device shown in the previous section.

We developed a C++ library to allow a user interfacing quickly the HandNavigator with various applications and without any extra effort, using pre-defined functions returning the desired data. The integration of the HandNavigator in this application is a first level toward a more global evaluation of the device efficiency covering entirely the range of an interaction. Note that in our library the virtual camera can be either attached to the scene or attached to the hand, depending on the user's requirements. For our examples here, the camera is attached to the scene to show a global view of the grasping configuration.

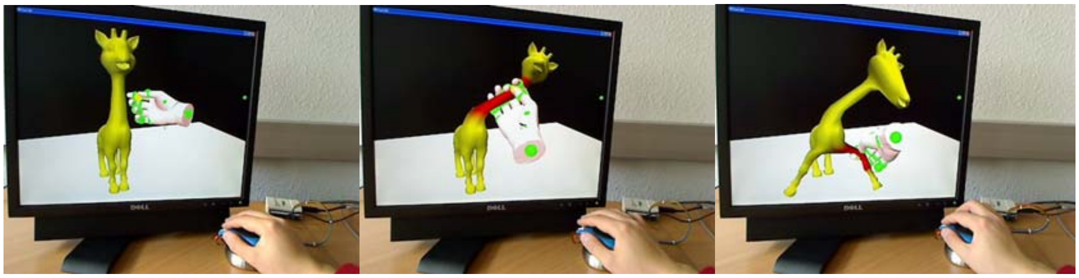


Figure 13: Playing with a deformable giraffe.

In Fig. 12, the left side shows three ways to grasp the same part coming from a food processor. Using our device we get the results as in the right side of Fig. 12. The virtual hand is handling the component in configurations very close to real ones. These grasping tasks were performed in about 15s without specific practice. Note that the virtual fingers are slightly penetrating the object (we measured about 4mm penetration), but this is not a problem as the real hand deforms when touching an object, which is not included. We can also mention that for the last configuration, the thumb in Fig. 12 is not matching the real configuration. This is because only the last phalanx is controlled, thus we cannot bend the thumb as done in the real situation. Compared to software applications with manikins manipulated with a computer mouse where fingers must be manipulated on by one, the proposed device significantly shorten the configuration time using the simultaneous dofs available on the prototypes.

Another example includes interaction with virtual deformable objects (see Fig. 13). In this application, we use the software of Rohmer *et al.* that computes constant volume deformation [18] to get an object deformation that is visually realistic. When the virtual hand gets close to the object and the user bends the first three fingers (the thumb, the index and the middle), the area of the object close to the virtual hand changes color, meaning the object is grasped in this area. Here, we use a very simple distance algorithm to detect whether the virtual hand is close to a part of the object or not.

More complex scenarios can be considered such as the placement of virtual avatars (including the position and orientation of each body) in virtual scenes or manipulation of elements for assembly processes, that are nowadays highly difficult to perform with traditional devices. This difficulty comes from the fact that traditional devices have a limited number of dofs and the dofs of the virtual hand must be operated independently of each other.

We clearly experienced difficulties with the first prototype versions because we could not easily open and close the fingers as naturally as needed. As mentioned in the previous section, the fact that prototypes V3trackball and V3scrollpad are using only one sensor per finger to move the fingers highly helps the user in performing the desired tasks whereas with prototypes having two different sensors per finger, we observed that the user often needs to look at his real hand to verify that his fingers are placed correctly on the device. With the V3trackball and V3scrollpad prototypes, the user has only to concentrate on on the screen to perform the desired tasks since his interaction with the sensor follows the natural scheme of the finger movements.

Figure 14: User testing one prototype of the HandNavigator in a public demonstration fair.

These prototypes were tested by several users (about 60 participants both male and female, children and adults) in several public fairs on the scenario of Fig. 13, mostly common people unfamiliar with such devices and specifically with the SpaceNavigator (see Fig. 14). Most of these users had some difficulties to get used to the SpaceNavigator but after 5 minutes, people could manipulate the virtual hand and grasp the giraffe. As for the sensors, users could easily act on them and perform desired motions. We are performing more precise tests with several users to get feedbacks in terms of controllability, usability, and performance and assess other issues about hand and finger sizes, left-handed or right-handed effects as well as human perception factors. First results show that users' performance increases from the first version to V3scrollpad, as with V3scrollpad, users achieve the scenario of Fig. 13 in less than 5s after only 3 tries while with the first version, the average time remains about 20s. As an overall users' evaluation, taking in consideration comfort of use and controllability, the first version gets a mark of 1.8/4 and the V3scrollpad 2.9/4. Further results will be presented and discussed in future works.

6. CONCLUSION AND PERSPECTIVES

We presented several versions of an immersive peripheral device allowing a user to control a virtual hand in position, orientation and gesture. Successive studies and comparisons led to the design of new prototypes to meet the objectives of accurate control of hand postures and fine dexterity for grabbing objects. Our specific focus is the selection of device shape and sensor technology to better monitor kinematics with an underlying SpaceNavigator. The comparative study among sensor technologies has led to scientific issues where physical phenomena, physiological behaviors and device structure significantly influence an overall interaction between real and virtual worlds. Thanks to a deep analysis of the ergonomics as well as different sensors' technologies, we were able to propose a range of prototypes to evaluate their influence on grasping dexterity. The analysis of such a range of prototypes shows that we can reach the user's needs without any difficulty and at low cost. Our device is cheaper and easier to integrate in office environments than other proposed solutions, calibration free compared to data gloves.

These prototypes were tested by several users that were not necessarily familiar with such kind of devices, so that we can improve them in terms of shapes, ergonomics, sensors, and propose a device suited for a range of users and a set of tasks.

Until now, we use the SpaceNavigator to control the position and orientation of the hand but it would be interesting to reconsider the way to generate the six dofs of the hand. In the future, we will study other devices based on wireless mouses or accelerometers to evaluate the influence of this technology over the device behavior.

The HandNavigator has been designed to be integrated in several applications fields, such as physical simulation, industrial processes or teleoperation. Future work will address these fields to produce new interaction capabilities.

ACKNOWLEDGMENTS

This work has been supported by GRAVIT. The authors would like to thank Idenea, an ergonomics consulting company, for their precious help in ergonomics matters, and the students of Grenoble Institute of Technology - ENSE3 engineer school for their participation in developing different versions.

REFERENCES

[1] P. G. Kry, A. Pihuit, A. Bernhardt, and M.-P. Cani, “HandNavigator: Hands-on interaction for desktop virtual reality”, in ACM symposium on Virtual Reality Software and Technology, 2008.

[2] G. Boothroyd, L. Alting, “Design for assembly and disassembly”, Annals of the CIRP, vol. 41/2/, 1992

[3] D. A. Bowman, E. Kruijff, J. J. Laviola, and I. Poupyrev, “3D User Interfaces: Theory and Pratice”, Addison-Wesley Educational Publishers Inc., 2004.

[4] G. Dewaele, F. Devernay, and R. P. Horaud, “Hand motion from 3d point trajectories and a smooth surface model”, in European Conference on Computer Vision, pp. 495–507, 2004.

[5] M. Schlattman and R. Klein, “Simultaneous 4 gestures 6 dof real-time two-hand tracking without any markers”, in ACM symposium on Virtual Reality Software and Technology, pp. 39–42, 2007.

[6] D. J. Sturman, D. Zeltzer, and S. Pieper, “Hands-on interaction with virtual environments”, in ACM SIGGRAPH symposium on User interface software and technology, pp. 19–24, New-York, NY, 1989.

[7] S. Kim, M. Ishii, Y. Koike, and M. Sato, “Development of tension based haptic interface and possibility of its application to virtual reality”, in ACM symposium on Virtual Reality Software and Technology, pp. 199–205, 2000.

[8] M. Bouzit, G. Burdea, G. Popescu, and R. Boian, “The rutgers master II-new design force-feedback glove”, IEEE/ASME Transactions on Mechatronics, vol. 7, June 2002.

[9] J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni, “Sofa – an open source framework for medical simulation”, in Medecine Meets Virtual Reality, pp. 13–18, 2007.

[10] J.-R. Chardonnet. “Real-time dynamic model for animation of poly-articulated objects in constrained environments with contact with friction and local deformations: application to humanoids and virtual avatars”, Ph. D. thesis, Université de Montpellier II, 2009.

[11] R. S. Johansson. “Sensory input and control of grip”, in Novartis Foundation Symposium, pp. 45–59, 1998.

[12] S. Zhai, P. Milgram, and W. Buxton, “The influence of muscle groups on performance of multiple degree-of-freedom input”, in CHI, pp. 308–315, 1996.

[13] B. E. Insko, “Passive haptics significantly enhances virtual environments”, Technical report, University of North Carolina, 2001.

[14] A. Lecuyer, S. Coquillart, A. Kheddar, P. Richard, and P. Coiffet, “Pseudo-haptic feedback: can isometric input devices simulate force feedback?”, in IEEE Virtual Reality, pp. 83–90, New Brunswick, NJ, March 18–22 2000.

[15] D. K. Pai, E. W. Vanderloo, S. Sadhukhan, and P. G. Kry, “The tango: A tangible tangoreceptive whole-hand human interface”, in Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 141–147, 2005.

[16] A. Talati, F. J. Valero-Cuevas, and J. Hirsch, “Visual and tactile guidance of dexterous manipulation tasks: an FMRI study”, Perceptual and Motor Skills, vol. 101, 2005.

[17] C. Chaillou and G. Casiez, “Périphérique d’entrée hybride isotonique/élastique”, patent number EP1720092A1, 5 2005.

[18] D. Rohmer, S. Hahmann, and M.-P. Cani, “Exact volume preserving skinning with shape control”, in Eurographics/ACM SIGGRAPH Symposium on Computer Animation, New Orleans, USA, August 2009.