A Virtual Reality Application for 3D Sketching in Conceptual Design

Marcello Lorusso[1], Marco Rossoni[2], Marina Carulli[3], Monica Bordegoni[4] and Giorgio Colombo[5]

[1]Politecnico di Milano, marcello.lorusso@polimi.it
[2]Politecnico di Milano, marco.rossoni@polimi.it
[3]Politecnico di Milano, marina.carulli@polimi.it
[4]Politecnico di Milano, monica.bordegoni@polimi.it
[5]Politecnico di Milano, giorgio.colombo@polimi.it

Corresponding author: Marcello Lorusso, marcello.lorusso@polimi.it

Abstract. The development of Virtual Reality in a wide range of field, including engineering related applications, has pushed towards the investigation of novel solutions that are able to take advantage of such new possibilities, while possibly trying to seamlessly integrate them within currently established workflows. Regarding conceptual sketching, which commonly represents one of the first activities taking place across Product Design development workflows, there are examples of applications that allow to shift from the 2D layout of traditional drawing to a fully immersive 3D environment where the user is able to produce strokes in space by means of a set of natural gestures. Despite sounding extremely intuitive, this kind of approach also comes with potential issues: the lack of a supportive surface onto which the user can rely on to produce strokes with a high degree of precision while not feeling tired after prolonged sessions can be problematic. Based on these premises, a new hybrid approach is proposed: the user is still immersed in the Virtual Environment, but is able to make use of a traditional tablet device which lays on a physical desk in order to produce visible strokes in Virtual Reality, while having the possibility to simultaneously manipulate the position and the orientation of the scene thanks to a hand tracking device to break into the third dimension. As designed, the application supports the generation of simple line strokes and few basic commands, but a thorough testing session is still needed to validate the solution and investigate on the necessary improvements.

Keywords: conceptual design, virtual reality, product design
DOI: https://doi.org/10.14733/cadaps.2022.256-268

1 INTRODUCTION

The matter of Virtual Reality (VR) as a revolutionary technology with regards to a wide range of fields, from entertainment to more professional applications, has been highly debated in recent years. If the focus is shifted on the design and engineering domains, it's easy to recognize a vast collection of research studies, prototype implementations and more recently even fully fledged commercial software that clearly show how far the development has come to enable these novel approaches to be integrated into established workflows. More specifically, both hardware and software tools have become widely available and affordable, and they're decently optimized to be implemented straight away without ending up tackling the steep learning curve that oftentimes is associated with such breakthrough technologies.

On the downside, the simple fact is that this transition still doesn't fully unfold, especially when considering industrial design and the complex shape generation methods that are usually associated with the development of everyday objects as well as automotive related products. In other words, the industry seems quite reluctant to embrace these new possibilities, choosing to stick with traditional approaches that are deemed as more reliable and consistent. There might be several reasons behind this mismatch: some of the core advantages of the more analog methods have not been offset yet by the simulations that have been developed so far for the digital world to replicate them: an example is the fact that as long as the sense of touch isn't integrated seamlessly within a VR environment, it's hard to imagine any design studio doing without the activity of physical clay modeling for the prototyping stage [6]. Despite this, many efforts are going in this direction and the scope for improvements is great; overall, the importance of VR is still widely recognized, in the sense that even though only few concepts so far have made it through to turn into commercial successes, it is also true that any investigation, study, and development in general contributed to pave the way to the present and for the future. Similarly, also Augmented Reality (AR) has recently proved itself as an effective set of tools towards this domain: the work proposed by Cascini et. al. [2] describes a new methodology for products and packaging exploiting the digital mapping of colors and textures upon real dummy objects to enhance the creative process in a collaborative way. The study also highlights the importance of the design of the user interface (UI) to get this kind of applications properly usable.

In fact, the novelty of VR as a new interaction paradigm with the digital world requires an almost complete redesign of the user experiences (UXs) compared to what designers and engineers have got used to in the last few decades [5]. This aspect is at the same time exciting but also critical: it is now possible to rethink the UXs in ways that simply were not conceivable until a few years ago with traditional desktop layouts, enabling more natural interaction systems that are not just based on the press of buttons on a keyboard. On the other hand, those who are willing to invest time and effort in the development of VR applications for such domains have to face against the lack of well-defined guidelines, protocols and assessment methods to gather a complete understanding of the potential of such solutions.

These are essentially the difficulties that this article highlights, and what follows is the description of the development of a prototype application carried out by the authors. Starting from the concept of a mid-air sketching system, a new hybrid approach is proposed: the user is still immersed in the Virtual Environment (VE), but is able to make use of a traditional tablet device which lays on a physical desk in order to produce visible strokes in VR, while having the possibility to simultaneously manipulate the position and the orientation of the scene thanks a hand tracking device to break into the third dimension. The user is able to sketch the desired strokes on a real surface while visualizing the result in the virtual environment, preserving, in this way, the natural perception of traditional means. The final goal of the application is to provide a more gentle transition from the established workflows and configurations to VR-based sketching system. In particular, the focus is on providing an alternative take to the activity of conceptual sketching, which usually comes at the very beginning of most product development processes and is relatively less bound to specific engineering requirements compared to more downstream phases. Finally, a number of considerations that indicate where future efforts should be oriented are also provided.

2 RELATED WORKS

The adoption of VR as an enabling technology to enhance the quality of the interactions in design related activities goes back to several applications developed at research level whose goal was to exploit new possibilities and alternative approaches. Above all this, the simple question about how to set up a VR environment in a reasonable way in terms of UX design is a matter that requires investigation by itself, since the technology is new, and only few solutions have actually established themselves as true benchmarks. In these regards, [10] is a valuable insight about the guidelines that developers should take into account when defining a VR environment aimed at design activities. Specifically, it would be simplistic to disregard this aspect as an accessory exercise just to make VR applications more enjoyable, because new ergonomic issues do come into play and cannot be overlooked, since oftentimes the whole workstation layout must be rethought from the ground up. The work of Chen et. al. [3] is a notable example that shifts the attention on this very aspect, with regards to user defined mid-air gesture interfaces.

Aside from this, other researchers have focused on collecting the research works available at the time of their studies to try to categorize in a rational way the plethora of applications that through the years have been proposed. For example, [1] focuses on a classification characterized by the algorithmic underpinnings of different shape generation methods, either 2D or 3D based. The paper [11] analyses the current understanding of the process of generation of ideas, in terms of their representation as conceptual shapes and tools supporting this activity.

Finally, a debate on the the same issue was already raised in [8], with a particular emphasis on the problem of comparing between each other the most relevant studies carried out in recent years at research level while underlining the importance of a well-defined protocol based on reliable metrics to provide a more rigorous testing methodology. For this reason, a small sample of users was involved in the testing activities for two different VR applications that are currently available on the market. Despite the fact that the number of sessions that were performed was still too small to provide results that could be deemed worthy of statistical relevance, some very interesting feedbacks were collected anyway based on qualitative considerations.

Firstly, a VR sculpting system called KodonVR (https://www.tenklabs.com/kodon) was tested, resulting in mixed opinions by the users. On one hand, they praised the ability to quickly shape any elementary object with very natural interactions. But as the level of complexity increased, the whole experience turned out to be more and more frustrating, especially from the point of view of the more "untrained" ones, that felt unable to control fine details in an unbounded, yet physically empty 3D space. A possible reason may be the fact that the aim of this application is about replicating in digital form the operations that are normally carried out for clay modeling, a very common process throughout the development workflow of industrial products with high aesthetic requirements. A crucial problem about this aspect is the inability in VR to effectively replicate the sense of touch, which is just as important as sight for such an approach. Despite future developments may eventually introduce on the market some new hardware tools that will improve on this situation, the technology still isn't mature enough to fully accomplish the goal. A study on this exact problem has been carried out by [9].

At this point, more positive feedbacks were gathered in relation to another VR application for 3D sketching called Gravity Sketch (https://www.gravitysketch.com/). This methodology basically consists of a 3D, virtual transposition of the traditional 2D drawing activity, where non-material entities such as line strokes and surfaces can be generated in mid-air, making it less dependent on the sense of touch. This solution is clearly not suited for advanced operations taking place downstream across the workflow, such as consistent definition of surface behaviors, but still provided great potential towards fast shape generation activities characterized by frequent reiterations. On these regards, some research works have been performed: for example, Cohen et. al. [4] recently proposed a 3D sketching system for generating NURBS-based 3D models using a Leap motion controller as input device.

A general assumption that can be made grounding on the comparison of these two methodologies is the fact that in both cases, users enjoyed the sense of presence allowed by the immersive environment and the fact that natural gestures applied to shape generation activities were much more intuitive and involving to accomplish compared the classic desktop UIs, where at least a basic level of knowledge is usually required due to their intrinsic cognitive approach, that is when commands are prompted from a keyboard [7]. A common downside though regarded the lack of dimensional references in the VE, which is a problem that ultimately alters the sense of scale from the user's point of view.

At last, when considering the ergonomic implications of VR, the absence of a supportive area like a table or a board that are normally part of a traditional workstation setup for 2D sketching, led to a decreased level of precision of the inputs and, conversely, increased level of fatigue even after short sessions, because of the lack of a constraining surface where the user would normally lean on. This specific aspect is the basis upon which the present work focuses, by outlining the alternative workstation layout that will be discussed in detail in the next paragraphs.

3 CONCEPT AND ARCHITECTURE

As mentioned in the previous paragraph, and based on a small sample of tests that the authors of the present article have been able to conduct in [8], it is clearly apparent that the greatest advantage of VR for this kind of application consists of the immersiveness of the environment, meaning that, with reference to the sketching activity, there is a natural match between the movements that the user is able to perform in mid-air, so in 3D, and the visible outcomes (line strokes, control points editing, etc.) that are generated which faithfully correspond to his/her actions in terms of spatial deployment. On the other hand, there have been great technological improvements to traditional UIs too, often introducing more advanced interaction and navigation tools based on hardware specifically developed towards designers and engineers (e.g., 3D Connexion special mices). Despite this, displaying 3D shapes on a flat monitor often forces the user to learn how a set of features inside a software works by means of a cognitive approach, which is opposite to natural gestures interfaces: in other words, the user is forced to compensate the absence of an actual third dimension by operating a set of specific commands, that in certain cases can hamper the overall experience and end up being frustrating especially for novices. It is clear from the beginning that the parametric CAD systems domain is not our field of choice when defining the scope of this research, but at the same time it is worth to reflect on why traditional interfaces are still so established and if there is anything that can be traded from those instead of rethinking the whole UX from the ground up. That said, some important considerations should be made regarding this point. If, on one hand the lack of immersiveness can apparently be considered a limitation, it is also true that parametric CAD systems have generally been working for decades based on said hardware and software set up with great reliability. The reason is that such 3D modeling systems are intended towards those design stages where the user does need a high level of precision and usually there is little or no tolerance at all with regards to the definition in space of a given product, given the technical implications that arise downstream in the development process (manufacturing and feasibility, performance evaluations, etc.). This is where exempting the user of the ability to generate 3D shapes freely becomes an advantage, since the obvious constraints imposed by this kind of configurations mean that only the rational utilization of features - in the CAD sense of the word - and the definition through rigorous parameters can provide coherent results.

Now, it is true that such technical aspects should not have a huge impact in the initial stages of the development, and more freedom is only beneficial for the designers to explore different ideas in a creative way: in currently established development workflows (i.e., automotive) a very common stage is about translating in digital form, as preliminary 3D models, the raw 2D sketches - often made on paper - by those designers that are in charge of the conceptual definition of the products in terms of their most distinctive visual characteristics. On the other hand, fully natural gestures UXs don't necessarily constitute the only way upon which VR applications should be based on, meaning that altering the scope of action for the user, when done smartly, is worth of investigation if it can enable a more gentle transition from traditional methodologies.

Figure 1: Architecture of the workstation with the point of view of the active user.

To further discuss this point, the industrial design field provides a plethora of valuable examples that describe the operative workflow of conceptual designers [1]. Aside from the obvious consideration that a 2D sketch, not being necessarily digital, can be obtained with an extremely simple array of tools almost everywhere, one of the most common gesture schemes performed by designers is about the generation of very well defined and often reiterated strokes through which they aim to immediately convey the fundamental shape of the product that they have in mind and its most striking traits. Only now though, VR technology allows to perform, theoretically, this operation in mid-air, but it's important to question if this is ideal towards the actual designer's goal aside from the advantages related to immersiveness.

This is where the proposal presented hereby comes in. In most of the current VR solutions, the lack of a supportive plane (i.e., a desk, a board) is the most frustrating aspect in the overall experience, because despite the optimizations that have been performed for each of the aforementioned applications, hand tremors and, most importantly, a tiring working position are the most limiting factor towards the full exploitation of such configurations, and also one of the most critical issues that emerged from the opinions of the testers involved in [8]. In the present article then, a hybrid solution is investigated, with the goal to take advantage of the most relevant feature of VR setups, namely the immersiveness, while trying to correct some of the inconveniences in terms of ergonomics and usability that state of the art systems do not seem to have fully addressed yet.

The workstation layout is presented in Figure 1, together with the corresponding point of view (POV) of the active user. Here, instead of being able to roam in a - physically - unbounded space to perform any sketching activity, the available area for the user is limited to the surface of a traditional tablet placed upon a static plane that is replicated as a fixed item in the Virtual Environment (VE). The dominant hand (DH) is the one in charge of producing line strokes, that by default, being only possible upon a 2D surface, can only be bidimensional, similarly to any non-VR system and basically to any drawing activity - digital or non-digital - in general. The goal is to restore that distinctive gesture set that designers are used to.

What VR allows to do at this point is to manipulate the immersive VE in terms of position and orientation of the scene relative to the user in ways that simply are not achievable through standard interfaces. For this purpose, a Leap Motion device has been implemented and placed in the area of the non-dominant hand (NDH) to track the position and orientation of one specific joint (i.e., the palm joint) and to simultaneously update the position and orientation of the whole scene accordingly. Furthermore, this dynamic manipulation through the NDH can be carried out while the DH is performing the sketching activity, which is what ultimately allows the user to break into the third dimension and to produce 3D shapes.

Before moving on to the description of the development process of the application, it's important to point out the most relevant criticalities that have been encountered so far. In a preliminary step, an extremely rough version of the system has been laid out, and it's been found out that - predictably - integrating the Leap Motion device in a seamless way is not an easy feat. If hand tremor and fatigue are detrimental for the UX of current VR sketching solutions, in this case the focus is about understanding if it can be beneficial to shift the issue from the DH that is in charge of the shape generation, to the NDH which only manipulates the scene. Therefore, several optimizations are crucial to make this solution viable, meaning that the user should ultimately be able to calibrate the sensitivity of the Leap Motion in such a way that DH and NDH can comfortably work in synergy. The next paragraph discusses in detail the more technical aspects of the development and how such issues have been tackled.

4 APPLICATION DEVELOPMENT

Based on the premises presented when describing the overall concept of the application, the most critical aspects related to such a workstation layout have been highlighted. Hereby, a description of the UX is provided, starting from the graphics characteristics to be experienced through the head mounted display (HMD) device and then going into the details of the more delicate matter of how the tasks for the DH and the NDH have been allocated in terms of interactions. The application was developed in Unity 3D and by means of C# scripts. In Figure 4, a general scheme of the working principles described hereby is provided in the form of an activity diagram.

4.1 Graphic Environment and User Interface

When a new scene is loaded, the environment is presented through the VR (HMD) device as a totally empty room, so the first thing that needs to be addressed is the representation of the virtual desktop surface. Considering the POV of the user, this plane can be calibrated to a specified height in order to match its position to the one of the physical desktop in front of the user upon which the actual tablet is placed. To achieve this operation, the user is asked to simply tap the pen in three - non-aligned - different points upon the tablet, and based on this, the application is able to calculate a preview of the virtual plane, which pops up as a pale yellow, translucent area. If this representation is faithful to the actual positioning of the physical desktop, the user can accept it, otherwise the operation is restarted. Once confirmed, the plane color switches from yellow to green, while still retaining the translucency property.

At this point, the user is able to only see an infinite plane area, meaning that the next calibration operation is about defining the correct perimeter of the virtual tablet which ultimately defines the actual drawing area. Now, since many different tablet models are available on the market, each with different drawing areas, the user is asked to pick its model from a list, so that the application can immediately load it upon the surface defined earlier as a pale yellow, translucent rectangle, whose dimensions in VR correspond the ones of the real model. In our example, a Wacom Intuos S was taken as reference.

Its positioning along the Z axis (perpendicular to the ground) is already fixed by the fact that the surface is coplanar with the infinite virtual desk plane defined above, and so is its orientation. On the other hand, X and Y coordinates are still not defined. To address this aspect, the user is finally asked to place the pen upon 2 of the 4 vertices of the drawing area of the physical tablet and once their exact position is understood, he/she can confirm which vertices they represent (i.e., top left; top right, bottom left; bottom right). The virtual tablet is finally displayed in its correct position, and its color shifts from yellow to green. Translucency

Diagram Application.JPG

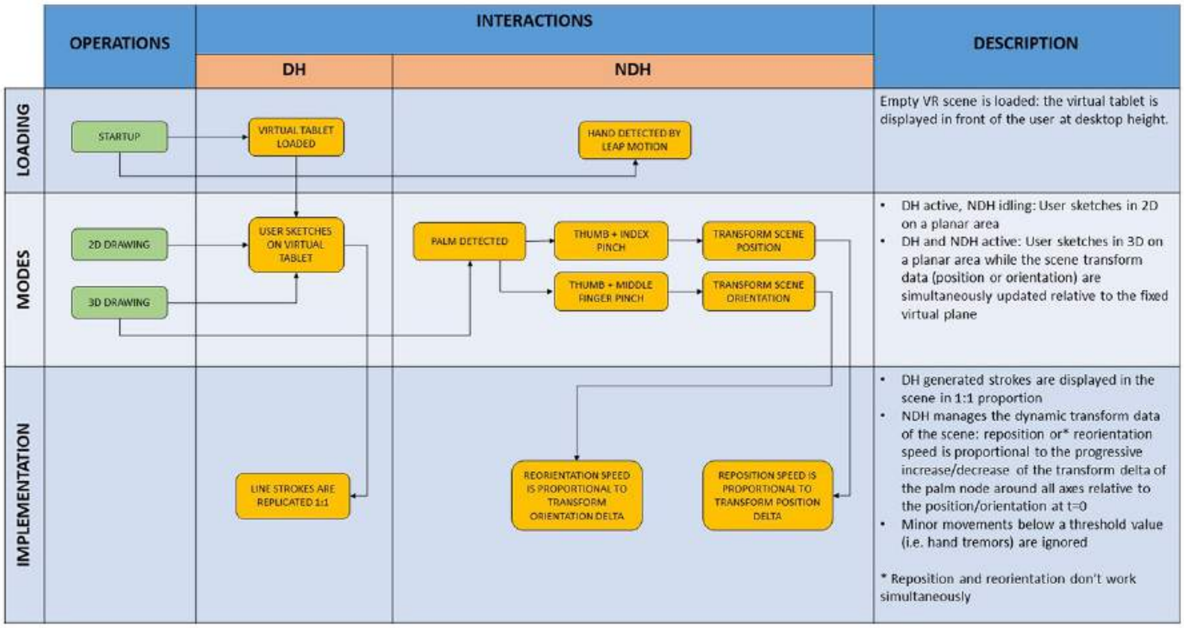

Figure 2: Activity diagram of the application.

is still retained as in the case of the desk plane defined before (Figure 3a). The drawing area is now displayed correctly in the VE; the center-point of the rectangle is also the origin of the scene, meaning that the global system of reference is defined by default. The user is now able to start sketching and interact with the scene.

An additional menu (Figure 3b) has been added to gain further control over the representation of the shapes to be drawn. In particular, the user is able to set from the UI the thickness and the color of the stroke and enable or disable the visibility of the virtual planes defined earlier.

In order to display the menu, the user performs a continuous fist gesture that is detected by the Leap Motion. In the meantime, the menu pops up at the level of the drawing area, where the desired parameters can be set by interacting with the pen tool.

4.2 Sketching

The DH is the one that supposedly holds the pen and performs the actual drawing activity. Firstly, this implies the necessity to represent the current pen position in the VE to provide the user with a visual reference. It is not necessary to reproduce a full dummy pen in VR, because by default it would be hard to represent its orientation in space, being the tip the only item that by default common tablets are able to track easily. Instead, a spherical cursor whose dimensions and color are coherent with the stroke settings described above is displayed when the physical pen is moved in proximity of the tablet. From then on, before introducing the enhanced possibilities allowed by the interaction that are made possible at the level of the NDH, the user is able to produce 2D shapes upon the available area (Figure 3c), meaning that in this state the application is nothing more than a VR transposition of common 2D sketching applications that are normally available for traditional desktop configurations, with the only, visual difference of being able to work inside an immersive environment.

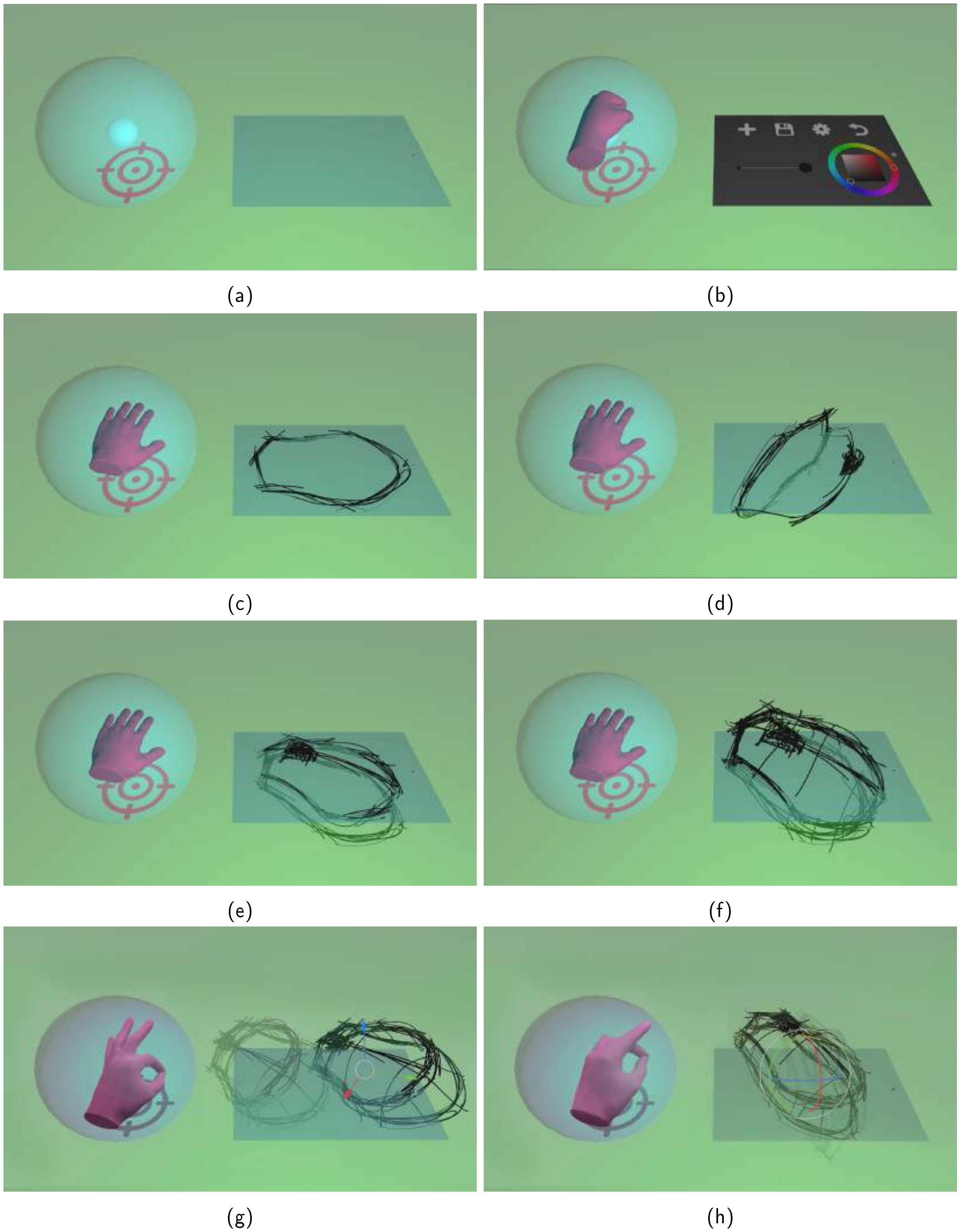

Figure 3: The figure shows the workflow for sketching a mouse. **(a)** an empty scene is instantiated; **(b)** setup of the application up with the contextual menu;**(c)** the user draws the first strokes on the plane; **(d)** the user rotates the scene and draws the following strokes along another plane; **(e-f)** the user draws the rest of the shape up to the desired level of detail; **(g-h)** pinch gestures are used to move and rotate the object.

4.3 Scene Manipulation

The main features that turn this application from basic 2D sketching to full 3D mode have been assigned to the NDH. Hardware-wise, this was possible by implementing a Leap Motion hand tracking device, to be placed below (i.e., upon the physical desktop itself), or above (i.e., attached to a dedicated support) the hand that the user requires to track, as shown in Figure 1. The core idea is about isolating a characteristic joint among the ones that the Leap Motion is able to associate to a physical hand, and, when a specific action is triggered by a defined gesture, manipulating the position and orientation of the scene, while keeping the drawing area fixed and coherent with the physical desktop plane. For this purpose, the palm joint is the one that most faithfully replicates the orientation in space of the hand.

Basically, these actions are the key to unlock the 3rd dimension, since the user is still able to perform the drawing activity and simultaneously manipulate the scene transform data (except scale). This is what ultimately allows to generate 3D curves (Figure 3d, 3e, 3f).

Regarding gestures, pinching is a common, and also comfortable action to perform in these cases. Specifically, once the palm is detected, the user can perform a continuous "thumb + index" pinch to trigger the dynamic transformation of the position of the scene (3g). Conversely, a continuous "thumb + middle finger" gesture triggers the dynamic transformation of the orientation of the scene (3h). Fine tuning the tolerance according to which the two gestures are detected is necessary to make sure that the user is triggering the desired one. The Leap Motion proved to be decently accurate in these regards after performing such optimizations. In both cases, the transformation occurs while staying relative to the fixed drawing plane and so to the default origin point defined in the calibration stage.

From the very beginning, the matter of coordinating correctly the operations performed by each hand was an issue that could not be overlooked, since especially in the earliest stages of development it presented itself as a crucial criticality. It must be noted that before carrying out any optimization, plugging and playing the Leap Motion controller in Unity and setting up the gestures system is very straightforward. In turn, the device comes with a high level of sensitivity in the recognition of the user's hand, though it cannot ignore unwanted movements like hand tremors or excessive transformations. Despite being a welcome feature in most applications where these tracking devices are implemented, in this case this aspect was quickly deemed as detrimental. In fact, replicating the hands movement for the scene manipulation without any threshold value to overcome the issue of tremor and by sticking with a 1:1 ratio between the transformation occurring at the level of the palm joint and the whole scene linked to it made the UX almost unusable.

Grounding on this, a number of adjustments had to be performed: instead of a 1:1 connection between hand and scene, a dedicated control algorithm has been developed. It implements a velocity control rather that a position control: it means that, given the initial position and orientation of the hand, the more the hand deviates from it, the faster the plane is moving/rotating according to the coordinate system placed on the virtual plane.

The pinch gesture is recognized at t=0, either for position or orientation. At this point, a coefficient - that is always less then 1 - is multiplied at each frame to the transformation delta of the scene object from t=0 to t=1. This coefficient is set to be proportional to the delta itself, meaning that it progressively increases as the position of the palm joint is displaced further away while transitioning from t=0 to t=1, while it does the opposite if, in the context of the same gesture (i.e., the pinch is still active), it's reverted back to a position closer to the one at t=0. Moreover, the global position or orientation of the palm joint at t=0 is not taken into consideration, meaning that regardless of the original configuration of the user's hand, the only parameter that manages the manipulation is the transformation delta. This guarantees that there isn't any unexpected jump when triggering the action, so the scene can be manipulated restarting from its last static configuration.

The tremor issue has been tackled by simply ignoring transformations below a defined threshold. The implementation of this feature has been carried out as follows: a sphere is represented as a visual reference of the default position of the palm joint at t=0. If, while performing the gesture, the palm joint crosses the boundary of the sphere and exits it, the threshold is exceeded, meaning that the manipulation can eventually occur. In turn, nothing happens as long as the palm joint stays within the boundary of the sphere.

5 DISCUSSION

In the current state, the application is still at a basic level of development, meaning that so far only the very fundamental features have been implemented in order to get an idea about this novel working principle and most importantly a better understanding about what are the most critical issues.

Ideally, a preliminary testing session could have addressed this matter to achieve a more complete development workflow at least with regards to the conceptual level of this approach. Unfortunately, due to the still ongoing Covid-19 pandemic and all the related problems in terms of organization of in-presence activities, testing could only be carried out by the authors themselves. Of course, this prospect is far from providing a statistically relevant sample of valuable results, especially considering that a defined testing protocol and a set of reliable metrics, as highlighted in [8], still do not exist for such kind of applications: by itself this is a matter that deserves further investigation.

On these premises, it is still worth to make a discussion about where this work positions itself in relation to other methodologies, even though based on qualitative observations. Specifically, the scope of this study is about exploring alternative solutions for VR sketching at its very core, questioning if the current limitations that hinder a truly widespread adoption can be highlighted and so if state of the art proposals can be improved in terms of interaction systems.

Current immersive environments, by providing natural gestures UXs in an unbounded, 3D physical space that synchronously trigger coherent actions in the virtual world, seem the ultimate solution when applied to the generation of 3D shapes to enhance freedom and creativity. On the other hand, it is also true that these approaches are very distant from the way designers and figurative artists in general have developed their drawing abilities through the course of the centuries. Perspective rules, use of colors for lights and shadows and other fundamental techniques that make up the wealth of skills of any experienced professional figure whose role is somehow related to the act of sketching, must be rethought or even discarded altogether at this point. Adding to this, if the intuitiveness of immersive environments seems like an enticing condition to improve the abilities of less proficient users, again, it was found in [8] that this idea is quite far from reality.

In terms of a comparison with traditional 2D sketching on the other hand, Figure 4 shows a side by side view between the output model of the desktop mouse shown above beside the representation of the same object realized with traditional drawing tools. The hand drawing is definitely cleaner to look at since it doesn't show the wireframe structure of the object, but overall proportions are preserved from one method to the other.

It might be an option for any common user to invest some time familiarizing with VR in this sort of field. That said, the rationale behind this research work and the proposed approach is about finding a way to make this transition more acceptable, by presenting an alternative, hybrid solution. The greatest challenge then is about achieving comparable outcomes to current VR solutions, namely 3D models, while relying on a set of interactions that are typical of traditional 2D sketching techniques. A supportive surface is the fundamental prerequisite to achieve this, but at the same time it represents the main obstacle to bridge in order to break into the 3rd dimension. This is where the idea of the scene manipulation through a dedicated device comes in, but it must be said that this aspect introduces new challenges in terms of hands' coordination and control, especially after longer sessions when it was found that the fatigue for the NDH to consistently keep a mid-air position was detrimental for inputs precision. Some refinements must be dedicated to this aspect in future developments.

After contextualizing this work in relation to the state of the art of VR sketching systems, it is also worth to figure out what are the advantages that this approach is supposed to bring in comparison to more established, traditional CAD solutions based on 2D interfaces. In particular the use of VR is crucial to introduce the vs 3D.PNG

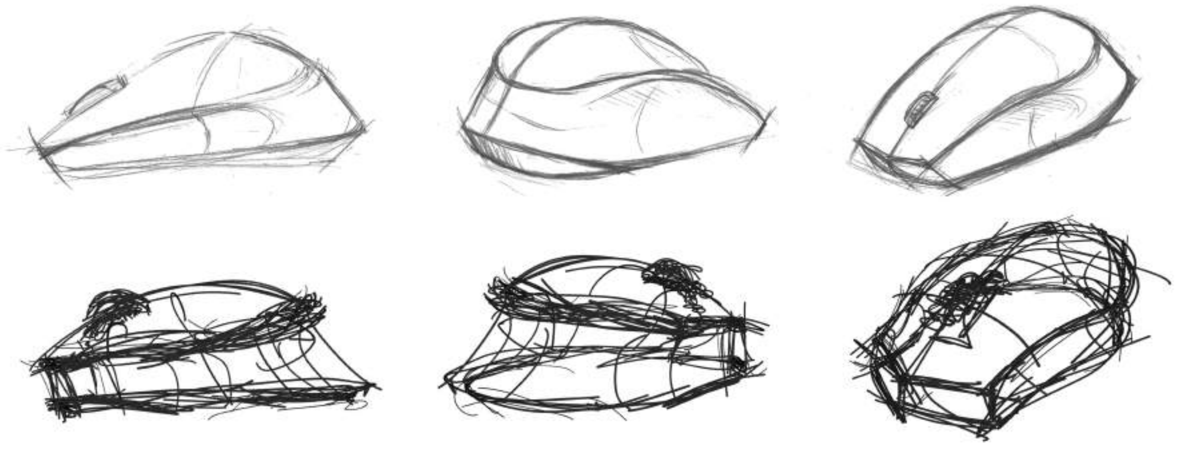

Figure 4: Comparison between hand made drawings of the mouse with traditional tools (top), and the output model of the same object obtained with VR (bottom).

immersiveness factor, something that traditional CAD tools clearly lack. It must be noted though that in this preliminary state, the application is able to produce just wireframe models that would be hard to integrate straight up into the established product development workflow even as conceptual proposals. Especially, the single stroke can be exported as a set of points that corresponds to the track of the cursors while the user is sketching. Those points could be then treated to generate more structured and defined information by, for example, interpolating the points with a polynomial function or spline interpolation algorithm. Enabling this functionality can promote the integration of this kind of application in industrial workflows; however, their implementation would require the application to either keep track of the strokes' history or exposing it at runtime (i.e. while the user is sketch, the software directly interpolates the strokes to obtain a mathematical function from them). At the current state, the application is able to save a list of points for each stroke for further off-line elaborations. Even beyond the simple generation of lines, the ability to manipulate meshes and surfaces (e.g. generates NURBS patches out of the strokes) could also be considered as a future development: besides the implementation of such an algorithm, the interaction system would be thought from the beginning to provide the proper usability to treat these new entities.

6 CONCLUSIONS AND FUTURE DEVELOPMENTS

In this research study, a new possible approach to tackle the issue of VR sketching is proposed, while considering the challenges that the adoption of this technology eventually poses.

The matter of designing a new UX that fully exploits the immersiveness of VR by means of natural gestures interfaces is a challenging aspect, in continuous development as the technology progresses and enables new possibilities. So far, traditional solutions are still deemed as more reliable, with reference to both the classic sketching activity on 2D supports and the generation of 3D models with standard CAD tools. With VR, developers are trying to bridge the gap between these two very different methods, but new problems have to be taken into consideration, especially when considering that a new set of skills is required from the point of view of most users.

Grounding on a previous research work where the best solutions currently available were preliminarily tested, a number of critical issues emerged, from which a debate upon the correct implementation of VR within conceptual sketching scenarios for Product Design was raised.

Specifically, fatigue and lack of input precision were highlighted as the two main issues that hampered the overall experience of mid-air VR sketching because of the absence of a supportive surface like a desktop on which the user can comfortably rest his arms and produce well defined strokes.

Since then, alternative interaction methods have been investigated, and this time the focus is on a hybrid configuration based on a standard 2D input system which consists of a digital tablet that works in synergy with an optical hand tracking device, through which it is possible to manipulate the virtual scene in terms of position and orientation. The related functions have been split between the dominant hand that performs the actual drawing task, and the non-dominant hand that is in charge of scene manipulation.

After a definition of the general architecture, several optimizations had to be carried out to improve usability, especially with regards to the matter of the sensitivity of the hand tracking device to tremor and minor movements in general. In the current state, the application still shows great scope for improvements, meaning that at this point it represents just a proof of concept in terms of UX design of an alternative way to exploit VR within 3D shape generation contexts. Conversely, that means that right now the goal is not about achieving a ready-made system to integrate seamlessly within the established workflows.

Furthermore, due to the current Covid-19 situation and the difficulties to conduct in-situ activities, testing has just been limited to the experience of the authors themselves, which has little significance in comparison to a more thorough session that will eventually involve external participants. This scenario will provide more valuable feedbacks to orient future improvements or may just indicate if it is worth to keep the development going altogether.

ACKNOWLEDGEMENTS

The Italian Ministry of Education, University and Research is acknowledged for the support provided through the Project *Department of Excellence LIS4.0 - Lightweight and Smart Structures for Industry 4.0.*

ORCID

Marcello Lorusso, http://orcid.org/0000-0002-0251-7770
Marco Rossoni, http://orcid.org/0000-0003-2714-9043
Marina Carulli, http://orcid.org/0000-0003-2101-5474
Monica Bordegoni, http://orcid.org/0000-0001-9378-8295
Giorgio Colombo, http://orcid.org/0000-0002-9999-8960

REFERENCES

[1] Bonnici, A.; Akman, A.; Calleja, G.; Camilleri, K.P.; Fehling, P.; Ferreira, A.; Hermuth, F.; Israel, J.H.; Landwehr, T.; Liu, J.; Padfield, N.M.J.; Sezgin, T.M.; Rosin, P.L.: Sketch-based interaction and modeling: where do we stand? Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 33(4), 370–388, 2019. http://doi.org/10.1017/s0890060419000349.

[2] Cascini, G.; O'Hare, J.; Dekoninck, E.; Becattini, N.; Boujut, J.F.; Guefrache, F.B.; Carli, I.; Caruso, G.; Giunta, L.; Morosi, F.: Exploring the use of AR technology for co-creative product and packaging design. Computers in Industry, 123, 103308, 2020. http://doi.org/10.1016/j.compind.2020.103308.

[3] Chen, L.C.; Chu, P.Y.; Cheng, Y.M.: Exploring the ergonomic issues of user-defined mid-air gestures for interactive product exhibition. In Distributed, Ambient and Pervasive Interactions, 180–190. Springer International Publishing, 2016. http://doi.org/10.1007/978-3-319-39862-4_17.

[4] Cohen, M.; Regazzoni, D.; Vrubel, C.: A 3d virtual sketching system using NURBS surfaces and leap motion controller. Computer-Aided Design and Applications, 17(1), 167–177, 2019. http://doi.org/10.14733/cadaps.2020.167-177.

[5] Cui, J.; Sourin, A.: Mid-air interaction with optical tracking for 3d modeling. Computers & Graphics, 74, 1–11, 2018. http://doi.org/10.1016/j.cag.2018.04.004.

[6] Gibson, D.: Why car designers stick with clay, 2016. http://www.bbc.com/autos/story/20161111-why-car-designers-stick-with-clay#:~:text=As%20a%20design%20reaches%20maturity,feel%20of%20a%20painted%20surface.

[7] Hummels, C.; Paalder, A.; Overbeeke, C.; Stappers, P.; Smets, G.: Two-handed gesture-based car styling in a virtual environment. In 30th International Symposium on Automotive Technology & Automation, vol. 1, 227–234. Automotive Automation Ltd., 1997.

[8] Lorusso, M.; Rossoni, M.; Colombo, G.: Conceptual modeling in product design within virtual reality environments. Computer-Aided Design and Applications, 18(2), 383–398, 2020. http://doi.org/10.14733/cadaps.2021.383-398.

[9] Price, S.; Jewitt, C.; Yiannoutsou, N.: Conceptualising touch in VR. Virtual Reality, 2021. http://doi.org/10.1007/s10055-020-00494-y.

[10] Regazzoni, D.; Rizzi, C.; Vitali, A.: Virtual reality applications: Guidelines to design natural user interface. In Volume 1B: 38th Computers and Information in Engineering Conference. American Society of Mechanical Engineers, 2018. http://doi.org/10.1115/detc2018-85867.

[11] Rossoni, M.; Bolzan, P.; Colombo, G.; Bordegoni, M.; Carulli, M.: Survey of digital tools for the generation of ideas. In Volume 9: 40th Computers and Information in Engineering Conference. American Society of Mechanical Engineers, 2020. http://doi.org/10.1115/detc2020-22443.