A VR Application for the Virtual Fitting of Fashion Garments on Avatars

Orestis Sarakatsanos
e-mail: orestiss@iti.gr
Information Technologies Institute
CERTH, Greece

Elisavet Chatzilari
e-mail: ehatzi@iti.gr
Information Technologies Institute
CERTH, Greece

Spiros Nikolopoulos
e-mail: nikolopo@iti.gr
Information Technologies Institute
CERTH, Greece

Ioannis Kompatsiaris
e-mail: ikom@iti.gr
Information Technologies Institute
CERTH, Greece

Dongjoe Shin
e-mail: dongjoe@metail.com
Metail Limited, UK

David Gavilan
e-mail: david@metail.com
Metail Limited, UK

Jim Downing
e-mail: jim@metail.com
Metail Limited, UK

ABSTRACT

In this paper, we present an interactive Virtual Reality (VR) application for fashion designers that aims to offer an immersive environment for accurately visualizing and testing garments in the design process. The VR environment simulates a designer's fitting room, offering close up inspection of an avatar wearing a garment in a plethora of movements. This, in turn, offers to the designer a better perspective of how the final product will look, fit and behave. In implementing the proposed application, we first review the option in game engines and libraries to support a realistic avatar-garment simulation, and propose a methodology for transforming the production-designed garments into a format that can be used in game engine cloth simulations.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality;

1 INTRODUCTION

During the past decade, the fashion industry has been undergoing a major shift in its design workflow with the novel 3D design software. 3D design tools such as Browzwear[1], CLO3D[2], Optitex[3] and Lectra[4] are increasingly popular nowadays, and many fashion designers are able to create garments in 3D to evaluate its fit on predefined static avatars using tailored physics-based simulation (PBS) engines. While such physics-based engines can deliver accurate simulations in many cases, they often lack in realism especially when dealing with fine details around complex garment layering. In addition, limited user interfaces in standard PBS engines lead many designers to rely on dedicated visualisation/presentation tools for the internal design communications, which normally creates significant amount of manual work, including 3D styling, adjusting camera parameters for garment renderings, stitching multiple renderings, and combining descriptive texts.

The gaming industry, on the other hand, took a slightly different route. Instead of accurate garment simulation, they focus more on the real time interaction using the emerging XR technologies. This approach is now getting mature to produce a comparable simulation result. It is worth noting that today's gaming engines have evolved significantly to deliver realistic movement of an avatar as well as its interactions with associated garment models simultaneously without dedicated GPUs. However, this means that we need to get garment designers involved in the the simulation process that will be rendered by the game engine, which often requires adjusting technical game engine parameters to ensure realistic clothing simulations. In addition, a garment model should be prepared for each character/avatar under the supervision of artists, which could be costly.

In this paper, we propose a methodology that can bring the best of both worlds to address the practical visualisation issues around the garment design cycle. More specifically, the main contribution of this paper is to introduce a solution that can transform the garment designs generated with a professional design software (i.e. Browzwear) for production, into a game-engine-ready 3D asset that can be realistically visualized in an immersive VR environment. As a proof of the concept, we develop a VR designer app, where the designers can perform virtual fitting of the garments on a range of custom avatars (e.g. fit models) and interact with the final design outcomes in more immersive and realistic environments. The proposed VR app also implements many practical user interfaces motivated by the day-to-day activities of designers, such as closeup inspection of the garment, which can give a better perspective of how the final product will look, fit and behave in a realistic setting to the designers.

2 RELATED WORK

3D professional design software has been and is continuously being enhanced to offer increasingly accurate garment simulations, limiting the fitting cycles (i.e. production of a garment and fitting its physical piece to a real fitting mannequin). Browzwear is one of the leading market 3D software for apparel design, development and presentation, supporting 3D renderings of garments and fit correction features. It comprises of multiple suites for pattern making, grading, fabric digital representation, stitching, 3D simulation and production guidelines. Very recently only (February 2021), Browzwear announced and released the animation potential, which is rather heavy in terms of the required processing power[5]. CLO3D is also a popular 3D fashion design software creating virtual garment visualization with cutting-edge simulation technologies for the fashion and apparel industries. However it is lacking a lot of precision options that traditional pattern drafting programs have. Optitex offers software solutions that include 2D/CAD design and 3D visualization platforms that cover the entire fashion supply chain. It is fully compatible with all major CAD formats. Lectra's Kaledo software allows designers to virtually create and visualize garments and textiles in a CAD environment while Modaris supports 3D prototyping and fitting. Lectra software has increased pattern making features compared to the rest 3D design software.

Creating and following the trend of digitally designed garments, there have been numerous efforts in creating the new market of digital only clothing, i.e. cloth designs that will not be produced but can only exist in the digital world. For example, TheFabricant[6]

[1] `https://browzwear.com/`
[2] `https://www.clo3d.com/`
[3] `https://optitex.com/`
[4] `https://www.lectra.com/en`
[5] `https://browzwear.com/pr-browzwear-2021-1-edition/`
[6] `https://www.thefabricant.com/`

is a digital fashion house. They specialize in 3D fashion design and animation, developing digital fashion editorials, digital clothing and occasional collections. They offer complete packages as well as single designs and animations that can be used in stores, online platforms and digital channels. In a similar direction, Scandinavian retailer Carlings[7] created a fully digital collection, offering a solution for the environmental impact of mass consumption and the ongoing demand for new clothes by online influencers[8].

VR and digitally-designed fashion have come together recently, particularly as a response to the physical distancing limitations set by the pandemic. Most efforts were made towards the presentation of new fashion designs, implementing VR catwalks and shows. For example, *The Fabric of Reality* is an Immersive VR Fashion Show and aims at taking audiences on a journey to explore the story and narrative behind the designers' collection[9]. VR also met fashion for educational purposes, when Art & Design students relied on Google Tilt Brush for quickly prototyping dress designs and designed their evening dresses using a virtual palette and brushes as they moved around a mannequin in the virtual world[10].

On the contrary, in this paper, we present an immersive VR application and tool that will allow designers to perform virtual fitting or stage runaway shows in highly realistic environments with their own-designed garments. What is more, the proposed application will make use of garments that that are originally designed for production using the selected professional design software and wont require any adjustments to produce the required simulations in the VR engine. In this way, the design workflow of fashion designers is not interrupted and the extension of their experience with our tool is seamless.

3 Physics engines for cloth simulation

In this section, we will discuss on the available software that can be used to develop a virtual fitting room. The proposed approach focuses on game engines in combination with cloth simulation libraries that support AR and VR and also allow a multi-platform build. We had to identify which engine offers the best simulation results, in what degree the simulation parameters could be configured and what additional tools are also included (e.g. for garment stitching).

3.1 Game engines and cloth simulation libraries

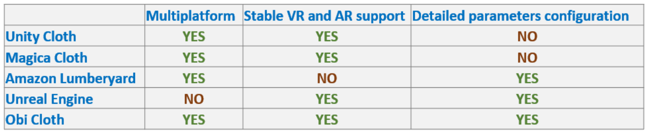
Unity is one of the most popular graphic engines offering multiplatform support for both AR and VR. In terms of cloth simulation options, while it has a built-in cloth simulation library (Unity Cloth), it comes though with limited capabilities, as it allows tweaking only a few parameters of the garment particles used for cloth simulation. However, due to Unity's popularity and active supporting community, there are a lot of third party libraries available with extensive documentation. For cloth simulation libraries, Magica cloth is a Unity plugin offering also fast cloth simulation but it doesn't really offer any significant upgrades over the built-in Unity Cloth in terms of features. Obi Cloth is another Unity plugin library, relying on particle-based physics, which compared to the two previous options offers, in addition to colliders, in-depth cloth simulation customization for both skinned and non-skinned garment meshes by the manipulation of various skin constraints.

Amazon Lumberyard[11] is a fairly new multiplatform game engine based on CryEngine[12]. In general, it is very similar to the Obi Cloth library in terms of how it approaches cloth simulation, offering similar features. However, the current multiplatform integration is in a beta stage, and as a result it is not as stable, documented and supported as Unity. Furthermore, one significant limitation of Amazon Lumberyard is the lack of support for AR applications.

Unreal Engine[13] offers cloth simulation support using NvCloth[14], which approaches clothing simulation in a similar way to Obi Cloth and Amazon Lumberyard. However, Unreal Engine's cloth simulation is GPU-based and can only be built for desktop applications, which means that it does not offer multiplatform support.

3.2 Software Selection

Towards implementing the VR designer app, we analyzed the platforms described above, to see which one offered most of the features that were needed. With respect to game engines and their capabilities to support avatar-garment simulations, the combination of Unity with Obi Cloth was the best choice for our scenario. Figure 1 outlines our selection based on the availability of required features. The multiplatform support that Unity offers (options Unity Cloth, Magica Cloth and Obi Cloth) was the key for the engine selection compared to Unreal Engine. Moreover, while the Amazon Lumberyard seems a good option for our case, their lack of stability and support due to being in a beta version made it risky to select that pipeline over the more stable and tested Unity engine. Comparing the cloth simulation options in Unity, the Obi Cloth library is the most powerful, offering detailed parametric configurations to optimize avatar-garment simulations. In the next subsection follows a more technical analysis of the key Obi Cloth library features.



	Multiplatform	Stable VR and AR support	Detailed parameters configuration
Unity Cloth	YES	YES	NO
Magica Cloth	YES	YES	NO
Amazon Lumberyard	YES	NO	YES
Unreal Engine	NO	YES	YES
Obi Cloth	YES	YES	YES

Figure 1: VR Designer Software Comparison Table

3.3 The Obi Cloth Library

Obi is a collection of particle-based physics plugins for Unity. Everything in Obi is made out of small spheres called particles. Particles can interact with each other, affect and be affected by other objects trough the use of constraints. All Obi assets use the same physics framework and share the same architectural design. There are some main Obi components that are used for every garment part we simulate.

- **Obi Solver** component is required to simulate an Obi object and it contains the general properties of Obi cloth components.

- **Obi cloth** is the component added to a specified garment in order to be simulated by the Obi library. The Obi cloth component requires a cloth blueprint based on the garment's mesh and contains parameters for the specific cloth affecting the simulation by the Obi Solver. For our case, where we need skeletally animated meshes we will be using a skinned cloth blueprint

Creating a blueprint allows the editing and the manipulation of the following cloth properties:

- **Skin Backstop:** How much is each particle allowed to "enter" the mesh (shown in green in Figure 2) Positive values allow entering while negative values slightly "push" particles out.

[7] https://carlings.com/en/

[8] https://www.jantrendman.com/en/trends/carlings-digital-collection

[9] https://www.fialondon.com/projects/ryot-studios-the-fabric-of-reality/

[10] https://news.illinois.edu/view/6367/1329212207

[11] https://aws.amazon.com/lumberyard/

[12] https://www.cryengine.com/

[13] https://www.unrealengine.com/

[14] https://github.com/NVIDIAGameWorks/NvCloth

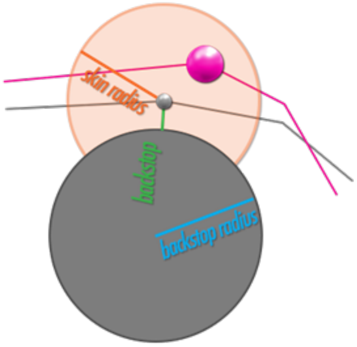


Figure 2: Skin Backstop/Backstop Radius/Skin radius properties

- **Skin Backstop Radius:** This parameter (shown in blue in Figure 2) controls the radius of a collision sphere that keeps particles from penetrating the skinned mesh.
- **Skin Radius:** This parameter controls the radius of a sphere (shown in orange in Figure 2), representing the distance that each particle can move during the simulation. It is expressed in world space units. A radius of 0 will cause particles to be 100% skinned, no simulation will take place.
- **Skin compliance:** Defines skin resistance, similar to other constraint type's compliance. A value of 0 will try to achieve complete stiffness, whether or not it is reached depends on the simulation budget (time-step size and amount of skin constraint iterations).

4 The VR designer application

The VR designer app comprises of several components; First, the VR scene, i.e. the room where the designers will be able to visualize how their designed garments fit and interact with a 3D avatar. The VR scene, besides the room, includes all the UI options that the app requires in order to allow the interaction between the user and the scene assets. The assets in our case include the avatars, the garments and their animations and simulations. More specifically, the designer will have the option to select among various avatars, including both pre-defined and scanned avatars of their own human fit models.. These avatars will be rigged so as to enable animations like cat walking, stretching arm, or other movements that let designers understand how the garment behaves.

Animation is the second component of the VR environment. A plethora of animations will be available through Mixamo[15], while the user will be able to upload a new custom animation of their own as an FBX file. Mixamo is a 3D computer graphics company that develops and offers 3D character animations. Some of the available animations are fashion related, such us catwalks and the ones we used for our application.

The third component is the garments, which have been designed in order to reach production through professional fashion design software (Browzwear). The transformation of a for-production-designed garment to a game engine asset that can be realistically simulated on an animated avatar is the most challenging part (more detail in section 4.3). Production-ready garments are designed in multiple 2D parts (patterns), which might be of different materials with different properties. These garment parts are simulated in an initial static pose of a 3D avatar in Browzwear and exported in a 3D-container file (FBX in our case), with each part as a separate mesh. These sub-meshes of the garment parts are skinned to the avatar in that pose and inserted in the Unity game engine. Afterwards, they are transformed in Obi components and the cloth parameters are added. These parameters include both the general cloth properties for each part (e.g. stiffness) and the particle weights (i.e. for each particle corresponding to a mesh triangle the backstop and radius parameters that indicate how freely is that particle allowed to move). Finally, the sub-meshes are automatically stitched using the Obi stitcher so that the garment moves as a single garment and does not tear apart.

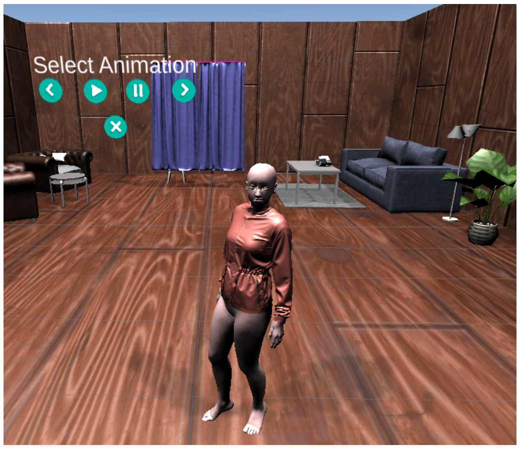


Figure 3: The VR scene

4.1 The VR Scene

4.1.1 Selection Panel

The VR scene is a 3D room that resembles a fitting room, where the users can interact with the assets of the scene. In more detail, the users (designers in our case) will be given a selection panel, through which they will be able to select from a number of existing avatars, different clothes and different animations. Using the VR headset's controller, the user will be given the choice to click on the corresponding VR buttons and change the avatar, the clothes or the animation to the combination they want to visualize. Additionally, camera movements around the visualized avatar / garment / animation pair will be enabled.

Towards giving to the users the aforementioned interactivity options, we explore three different approaches. In the first one, the user needs to use the controller's pointer and click on floating buttons in the VR scene. This kind of interaction is quite intuitive for new users to understand how to interact with the scene. However, this might bloat the scene with buttons, while clicking on floating buttons with the controllers might not be easy to manipulate. The second approach is to map all these actions to the buttons in the controller itself. This sounds as a more elegant approach, keeping the whole scene clearer so that the user can focus better on the visualizations. This resembles the interaction that people currently have with games with controller clicks, which might require learning time on how to use the controller, especially for people not used to playing video-games. Finally, there is also the hybrid method, where selection panels emerge in the scene on demand by using the controller and then controller guided clicks are enabled as in the first option. This resembles the interaction user currently have with smartwatches but with controller-enabled laser clicks instead of haptic-enabled buttons. For the current working demo, we have relied on the first approach, but we will also explore the other two options in the future

4.1.2 Interaction with the garment

Additionally, Obi Cloth library contains a tool called “Obi Particle Dragger”. When this tool is attached to an Obi Object it allows the

[15] https://www.mixamo.com/

user to manually click and pull the garment and see how it naturally moves. For example, pull a jacket slightly and see how it moves while the avatar wears it. This interaction will also be enabled in future versions.

4.2 The avatar and the animations

In the VR scene there are three main assets, the avatars, the garments and the animations. In order to utilize a human avatar in the VR scene, it needs to be properly rigged. This means that it needs to have a complete humanoid bone structure. In order for Unity to recognize an avatar and allow animations on it, it needs to be generated as humanoid rig, i.e., a skeleton containing all the body structure information.

After the avatar and the rig are imported into the scene, an animator component is added to both of them using Unity's animation controller. Unity's animator editing allows to import a number of animations and create transitions between them (i.e. allowing an avatar to change from performing one animation to another one). A plethora of animations for humanoid rigs are available from Mixamo and can be easily imported in the Unity VR designer app.

Finally, since the avatar and the garments are different objects, the animation controller should be passed to the animator component of all these objects that need to animate together. For ensuring that the avatar and garment are synced and smoothly follow the same animation in each frame, in the animator tab, both the Root Motion and the Animate Physics must be selected. Root motion enables different objects to use the same animator, while Animate Physics is required for the cloth simulation.

4.3 The garments

The primary functionality of our VR designer app is to transfer the accurate simulation results from professional garment design software to the immersive VR space created by standard game engines. This means that we need to implement a scalable data processing pipeline that can ingest the garment assets prepared for design and production and not for game engines.

Conventional garment designers typically work on a 2D space, producing multiple 2D cutting patterns for a single garment. These 2D design patterns are normally accompanied with more complex stitching instructions (e.g. which garment patterns are stitched together, and how to stitch them around the body part) which eventually produce a 3D garment through PBS. Although it is possible to export a 3D garment model in a game engine compatible format (e.g. FBX) directly from the design software, it is often not straightforward to transfer the stitching information and simulation properties between two different platforms. This section will explain the details about how we prepare this data.

4.3.1 Exporting garments from Browzwear

Professional design software (such as Browzware) normally represents a single 3D garment with multiple sub-meshes, corresponding to 2D cutting patterns. This would result in duplicated vertices at stitching seams, but this is generally preferred for texturing, as it gives more flexibility when storing associated map data (e.g. diffusion map at the overlapping areas). If we wanted to create a single mesh for the whole garment, without duplicated vertices, we would need to recompute all the UVs (texture coordinates) and make them continuous for the whole topology, which is not a trivial task. So we chose to export the garment with multiple submeshes, each one pointing to a different UV area of a single texture (sometimes called texture atlas, since it is made of a collection of pieces).

If we rigged the garment and simply applied skeletal animation to it, two duplicated vertices would always share the same position as they animate. However, because we are going to apply physics to them, the duplicated vertices may drift apart. That may appear as if the sleeve tears off around the shoulder, for instance. Because we exported the garment as separate submeshes, we can loop through these submeshes in Unity and find the shared vertices between two submeshes and add an Obi Cloth ”stitch” between them. This way the garment gets stitched again.

Another constraint for the exported garment model is that the number of garment vertices should not be too high. This is against a general rule of thumb for the high quality rendering (i.e. maintaining high polygon counts as much as we could). However, this will be inevitable when we are considering an agile PBS renderer running on a mobile platform. To minimise the number of vertices we export the outer meshes only (NB. garment mesh can be two-sided) with a fixed mesh sampling rate of 1.5 cm, and the fine detailed garment accessories such as a zippers are often simplified before export.

4.3.2 Skinning and the effect of the initial pose

An important requirement for the garment models is that they need to be skinned based on the same humanoid bone structure of the avatar the garment will be fitted upon. This results in the correct syncing of any animations of the avatar and the garment it wears.

A high-quality garment skinning is similar to a humanoid character skinning, which normally requires a skilled 3D designer to place a bone structure and then define the skinning weights for each garment vertex. However, to implement a more scalable pipeline, we assume that most garments will closely follow the movement of the underlying body model. This allows us to reuse the same bone structure from a body model for a garment after running PBS for a target garment on a rigged body model in Browzwear.

However, assigning skinning weights for a garment vertex is trickier than allocating a bone structure, as there is no clear one-to-one mapping between garment and body vertices. To simplify the problem, we developed an algorithm that finds the K-Nearest Neighbors (KNN) on the avatar mesh of each garment vertex. We then average the known skinning weights of those K vertices and transfer them to the associated garment vertex. Since this is based on the distance, an incorrect mapping can also be produced, particularly when a single garment vertex is close to multiple body parts, e.g. a sleeve vertex at I-pose (arms downs touching the legs) could have body vertices from a torso and an arm part. To address this, we can adjust the distance threshold in the KNN, or the K value, or use a T-pose for a clearer vertex separation.

The initial pose of the avatar/garment pair on which the skinning was computed is also an important factor affecting the simulation result. In more detail, the default pose for Unity to recognize a humanoid rig is the T-pose, which is widely used as the default pose in gaming engines. However, computing the automated skinning on a T-posed avatar, creates distortions and wrinkles in the garment and affects the Obi simulations afterwards. In tackling this issue, we selected the A-pose as the initial static pose, which is more natural in terms of how a garment drapes on an avatar and the automated skinning computes more fitting values for a more accurate simulation. In order to visualize the aforementioned, we can see in Figure 4 the wrinkles that appear in the T-pose, which will continue to appear during the simulation as well. This is due to the fact that the game engine considers that this initial posed simulation is the garment's mesh resulting in wrong cloth simulation.

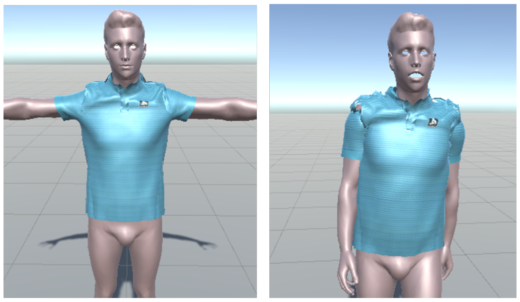


Figure 4: Left: garment skinned with T-pose in initial pose. Right: simulation of that garment during catwalk

With the use of the A-pose as the initial pose for skinning, we can see in Figure 5 that the chest wrinkles are avoided, and the garment maintains the shape it naturally has.

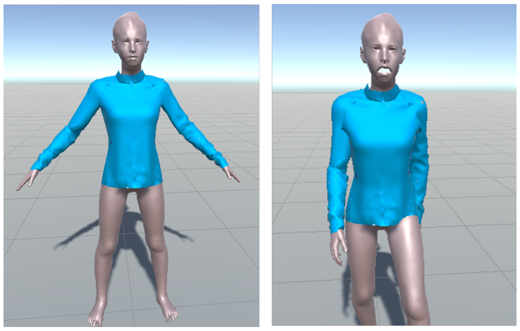


Figure 5: Left: garment skinned with A-pose in initial pose. Right: simulation of that garment during catwalk

4.3.3 Garment properties

As mentioned before, the garments are skinned meshes based on the skeleton rig of the corresponding humanoid avatar. By default, despite the character appearing to wear clothes that move according to the avatar movements, there is no physics-based cloth simulation (e.g. draping, wrinkling). In order to add cloth simulation to the garments, the Obi library allows us to give two types of properties to the garments.

First, each garment part has general properties affecting the whole sub-mesh of that part, indicating the properties of the material it is made of, e.g. stiffness, elasticity, etc. In addition, for each sub-mesh, the Obi library assigns one particle to the location of each mesh polygon. For each particle, we can set certain properties (e.g. skin backstop, radius, etc.), which affect how much is each particle allowed to move with respect to its initial position and neighbor particles. By effectively setting up all these properties, we can achieve a realistic simulation. Our objective is to develop a machine learning model that will predict the values of these properties automatically using as ground truth the simulations from the Browzwear software, so that the final simulation is of equal quality.

4.3.4 Stitching

Production-designed garments, more often than not, consist of many parts, each of them potentially being of different materials and having different properties. As a result, each garment that can be exported from professional design software consists of various sub-meshes, correctly simulated in a specific 3D pose. In order for these sub-meshes to move together and not fall apart once an animation is added, they need to be stitched in the game engine environment. The Obi library provides the Obi stitcher component, which can stitch two meshes together.

To automatically stitch the whole garment, the process is the following:

First, we scan all the particle positions of each sub-mesh of the garment. Then, for all possible combinations of two sub-meshes, if the particles positions of the one sub-mesh overlap (or are really close) with the particles of the second sub-mesh, the algorithm stitches these specific edges. For example, when our algorithm considers the combination of a sleeve and a main torso sub-mesh, it will identify the particles on the top of the sleeve that are in the same positions with the torso part, and then stitch these together. However, when the algorithm considers the combination of the left and right sleeve sub-meshes, no edges will be stitched since there are no overlapping particles. During simulation, this stitching algorithm manages to scan and stitch all the sub-meshes successfully without any significant delay in terms of performance. It is worth noting that while the stitching information is provided by the designers in the professional fashion design software for the 2D patterns and we could potentially export a single mesh for the whole garment, this cannot be optimal in terms of simulation accuracy, since we would face restrictions to the variety of values used for the simulation properties. On the other hand, stitching the sub-meshes within Unity allows us to have different properties for each sub-mesh, and as a result, each one of them can behave as a different Obi component, with different materials, properties and simulation values. Summarizing, stitching in the Unity engine with the Obi stitchers allows us to:

- Have a similar visual result as if the garment was already a single mesh.
- Edit and manage each sub-mesh separately which adds more depth and accuracy to the simulation.

5 An alternative approach

Realistic cloth simulation is a very challenging task, especially when considering the computational complexity limitations of existing hardware. Professional 3D design software such as Browzwear, with its own built-in cloth physics-based simulation engine and being a dedicated fashion software, offer high quality simulations with increased detail. While such software typically offers simulations on static poses without animation capabilities, Browzwear in its latest update[16], they expanded their platform's technologies and they have included animated simulations of garments. However, they only offer a demo animation catwalk at the moment, and it requires significant processing power to render. In addition, Browzwear is a desktop software offering no capabilities on VR and immersive environments.

Comparing game engines and professional fashion design software, game engines offer what a virtual garment simulation application requires in terms of interactivity and immersiveness. However, reaching a sufficient level of accuracy in the simulations is not straightforward. On the other hand, professional fashion design software offers a high level of accuracy. Considering the above, we have also examined a different approach, combining the benefits of both worlds by using Browzwear-processed detailed animated simulations as input to a game engine. In this direction, we have created the garment simulations in the Browzwear software, exported it in Alembic files and inserted them in a game engine. Alembic files are a computer graphics file format, capable of storing all the information regarding the garment and avatar meshes, textures, behavior, and movement. Alembic files can capture the highly accurate animation of the human body and the garments behavior on it but it has to be pre-computed offline and acts like a 3D video, i.e. it offers zero interactivity with the garment and the simulation itself.

[16] https://browzwear.com/pr-browzwear-2021-1-edition/

In this case we can offer the highly detailed simulations in an immersive environment. However, it is worth noting that this kind of simulation lacks any kind of interaction in the scene apart from a simple camera rotation around the animated avatar-garment pair. On the other hand, game engine simulations offer more interactivity options (e.g. drag the garment at a specific place and see how it reacts). Moreover, with the Browzwear-based Alembic files, the user can visualise a specific movement/animation/pose of a character wearing a garment that cannot be changed after (i.e. it has to be pre-captured). As a result, the alembic file approach on the one hand does allow the designers to have a very clear idea of how the product will look on a real body. On the other hand, however, since the whole process is pre-computed and pre-captured, the designer won't be able to test multiple poses or change anything on the go. A new Alembic file would be required to test any other pose.

In terms of simulation quality, we have created animations utilizing both pathways (i.e. Browzwear pre-computed animations transferred in VR though Alembic files vs. real time game engine simulations with the Obi Cloth Library). The results for a single pose of the animations can be seen in Fig. 6. We can see that the Browzwear simulation (left) is more detailed with respect to the fabric wrinkles and draping compared to the Obi-based simulation (right), which shows rougher deformations that could happen with a tougher fabric. In addition, considering that Alembic animations are pre-computed, there is no limitation on the mesh quality, compared to the Obi-based simulations which pose specific limitations on the number of polygons so that the simulation can be real time. This has also an effect on the texture visualization, which can be seen in Fig. 6.

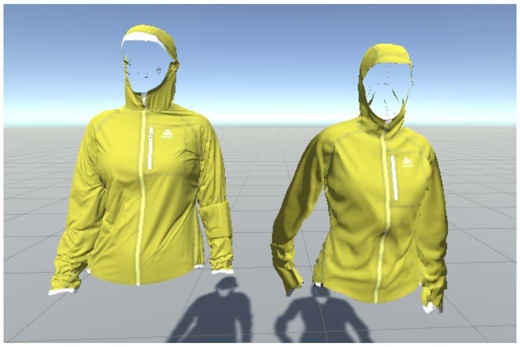


Figure 6: Cloth simulation of Browzwear (left) vs. Obi Cloth (right)

To conclude, we can see that there is a trade-off between animation/simulation accuracy and interactivity/real time capabilities with these two approaches. What is best depends on what the designer needs on that specific time, accuracy or being able to edit on the go. Nevertheless, the combination of Browzwear simulations with a game engine is a pathway worth of investigation. Thus in the developed application, we have implemented both pathways and in future work we plan to pilot both implementations with fashion designers and get feedback on which one fits mostly their requirements. Finally, it is also worth noting that the parameterization of the Obi garments is not optimal at the moment, so the accuracy of this pathway can increase in the future. In fact, one of our plans for future work is to develop an algorithm for automatically discovering the optimal parameters for each fabric and garment design.

6 Conclusions and future work

In this paper, we presented an interactive VR environment for facilitating the designers to perform virtual fitting and testing in an immersive and realistic environment. The VR environment allows close up inspection of the garment and avatar movements and also provide some interaction tools that enable the designer with a better perspective of how the final product will look, fit and get interacted with (e.g. by allowing them to freely move (with 6 DoFs) around the 3D avatar and garment while it performs the required animations). The main challenge for creating such an application is how to transform a garment designed for production using a professional garment design software (Browzwear in our case) into a game engine ready version without interrupting the designers' workflow. In this direction, the proposed approach includes guidelines for exporting the garments from Browzwear, automatically skinning the garments to the selected avatar, stitching the different parts in Unity and setting the properties for each part to achieve realistic simulations. In the future, we plan to perform user testing of the application with fashion designers and design a deep learning algorithm for predicting the optimal values for the garment particles.