



eTryOn - Virtual try-ons of garments enabling novel human fashion interactions

| | |
|---------------------------|---|
| Project Title: | eTryOn - Virtual try-ons of garments enabling novel human fashion interactions |
| Contract No: | 951908 - eTryOn |
| Instrument: | Innovation Action |
| Thematic Priority: | H2020 ICT-55-2020 |
| Start of project: | 1 October 2020 |
| Duration: | 24 months |

Deliverable No: D2.1

Core functionalities of the software for the avatar-garment simulations and basic garment collection for development purposes

| | |
|---------------------------------|--|
| Due date of deliverable: | 31 May 2021 |
| Actual submission date: | 11 June 2021 |
| Version: | 1.9 (final) |
| Main Authors: | Ursina Häfliger (ODLO), Robert Boland (Metail) |



Project funded by the European Community under the H2020 Programme for Research and Innovation.



| | |
|----------------------------|--|
| Project ref. number | 951908 |
| Project title | eTryOn – Virtual try-ons of garments enabling novel human fashion interactions |

| | |
|-------------------------------------|---|
| Deliverable title | Core functionalities of the software for the avatar-garment simulations and basic garment collection for development purposes |
| Deliverable number | D2.1 |
| Deliverable version | Final |
| Contractual date of delivery | 31 May 2021 |
| Actual date of delivery | 11 June 2021 |
| Deliverable filename | eTryOn_D2.1_final.docx |
| Type of deliverable | Demonstrator |
| Dissemination level | PU |
| Number of pages | 48 |
| Workpackage | WP2 |
| Task(s) | T2.1, T2.2, T2.3, T2.4 |
| Partner responsible | ODLO |
| Author(s) | Ursina Häfliger (ODLO), Robert Boland (Metail) |
| Editor | Elisavet Chatzilari (CERTH) |
| Reviewer(s) | Martina Pugliese (MLZ) |

| | |
|-----------------|---|
| Abstract | In this deliverable, the abstract classes of the simulation algorithms, the existing software and the first set of garments will be provided. |
| Keywords | 3D garments, VStitcher Browzwear, Simulation algorithms |

Copyright

© Copyright 2020 eTryOn Consortium consisting of:

1. ETHNIKO KENTRO EREVNAS KAI TECHNOLOGIKIS ANAPTYXIS (CERTH)
2. QUANTACORP (QC)
3. METAIL LIMITED (Metail)
4. MALLZEE LTD (MLZ)
5. ODLO INTERNATIONAL AG (ODLO)

This document may not be copied, reproduced, or modified in whole or in part for any purpose without written permission from the eTryOn Consortium. In addition to such written permission to copy, reproduce, or modify this document in whole or part, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced.

All rights reserved.

Deliverable history

| Version | Date | Reason | Revised by |
|-------------|------------|--------------------------------|---|
| 1.0 (alpha) | 06/04/21 | Table of Contents | Ursina Häfliger (ODLO) |
| 1.1 | 10/05/21 | Initial version | Ursina Häfliger (ODLO) |
| 1.2 | 14/05/21 | Additional Content by | Robert Boland (Metail) |
| 1.3 (beta) | 17/05/21 | Beta version | Ursina Häfliger (ODLO) |
| 1.4 | 19/05/21 | Version with reviewer comments | Martina Pugliese (MLZ) |
| 1.5 | 20/05/21 | Implemented comments | Ursina Häfliger (ODLO), Robert Boland (Metail) |
| 1.6 | 21/05/21 | Reviewed version | Ursina Häfliger (ODLO) |
| 1.7 | 03/06/2021 | Pre-final version | Elisavet Chatzilari (CERTH) |
| 1.8 | 11/06/2021 | Additional content | Robert Boland (Metail) |
| 1.9 (final) | 11/06/2021 | Final version | Elisavet Chatzilari (CERTH) |

List of abbreviations and Acronyms

| Abbreviation | Meaning |
|--------------|------------------------------------|
| BW | Browzwear |
| CAD | Computer Aided Design |
| DL | Deep Learning |
| dxf | Drawing Interchange Format |
| fbx | Autodesk Filmbox format |
| FW | Fall Winter |
| GDPR | General Data Protection Regulation |
| M | Month |
| PBS | Physics Based Simulation |
| SS | Spring Summer |
| u3m | Unified 3D Material |
| VR | Virtual Reality |
| VS | VStitcher |

Table of Contents

| | |
|--|----|
| 1. Executive summary | 9 |
| 2. Introduction | 10 |
| 3. 3D Garment Collection | 11 |
| 3.1 Season of pilot phase | 11 |
| 3.2 Garment Categories | 11 |
| 3.3 Fabric Construction | 12 |
| 3.4 Gender and Sizes | 13 |
| 3.5 Designing in VStitcher | 13 |
| 3.6 Styling of garments | 19 |
| 4. Additional preparation of the 3D garment for the eTryOn project | 21 |
| 4.1 Size set pattern and grading | 21 |
| 4.2 Size set avatars | 23 |
| 4.3 Saving and exporting 3D garments | 26 |
| 5. Exporting Browzwear-designed garments for game engines | 27 |
| 6. Creating avatars from mobile scans, ready for use in VStitcher and eTryOn apps | 29 |
| 6.1 Combining with avatars from QuantaCorp Scan | 29 |
| 6.1.1 Fix scale and axis | 29 |
| 6.1.2 Extract landmarks | 30 |
| 6.1.3 Clean up scan | 30 |
| 6.1.4 Mesh & skeleton fitting | 31 |
| 6.1.5 Create VStitcher Avatar | 32 |
| 6.2 Demo of end result | 33 |
| 7. Combining the Garments and the Avatars | 35 |
| 7.1 Garment rigging and skinning | 35 |
| 7.2 Animations for Use Case 1 - VR Designer App | 35 |
| 7.3 Proof-of-Concept (manual process) to create ObiCloth garments from Browzwear files | 36 |
| 7.4 Prediction of Garments Physics parameters for Unity-Obi Cloth | 36 |
| 7.4.1 Optimisation techniques | 36 |
| 7.4.2. Rendering parameter mapping with DL | 38 |
| 7.5 Proof-of-Concept (partly manual) process to create garments in Snapchat lens filter from Browzwear files. Technology towards Use cases 2 and 3 | 40 |
| 7.5.1 Matching the Snapchat skeleton | 40 |
| 7.5.2 VStitcher Garments in Lens Studio | 41 |
| 7.5.3 Demo of end result | 46 |
| 7.6 Pipeline for the magic mirror | 47 |
| 8. Conclusion | 48 |

List of Figures

| | |
|---|-----------|
| Figure 1. Example of Smart Design Toolbox with different basic blocks of men midlayers | 14 |
| Figure 2. Steps of designing a 3D garment in VStitcher between pattern and design department | 14 |
| Figure 3. Digitized 100% real size pattern in VStitcher | 15 |
| Figure 4. Digitized pattern is stitched together in VStitcher | 16 |
| Figure 5. Arrange Mode with clusters in VStitcher | 16 |
| Figure 6. Parametric base size avatar man | 17 |
| Figure 7. Physical fabric properties and captured fabric texture in VStitcher | 18 |
| Figure 8. Automated garment simulation process in Vstitcher | 18 |
| Figure 9. Automated dressed garment without manual styling | 19 |
| Figure 10. Dressed garment after manual styling | 20 |
| Figure 11. Graded pattern in VStitcher, also called a nest or nested pattern | 22 |
| Figure 12. Dressed 3D garment in Size XS, S and M from left to right | 22 |
| Figure 13. Dressed 3D garment in size L (left) and size XL (right) | 23 |
| Figure 14. Scanned avatar imported as fbx file with anchor points in VStitcher | 24 |
| Figure 15. Scanned avatars in Size XS, S and M from left to right | 24 |
| Figure 16. Scanned avatars in Size L (left) and XL (right) | 25 |
| Figure 17. Avatar size set organizer in VStitcher | 25 |
| Figure 18. Pipeline from Garment to Unity Scene | 27 |
| Figure 19. Impact of different grid settings on number of polygons | 28 |
| Figure 20. Comparison of different UV layout options | 28 |
| Figure 21. The scan avatar before and after the scale and axis is corrected | 30 |
| Figure 22. Landmarks, marked in green, added to the avatar | 30 |
| Figure 23. Showing where ‘webbing’ artefacts have been cleaned up in the model scan | 31 |
| Figure 24. The avatar after the skeleton has been fitted | 32 |
| Figure 25. VStitcher Anchor Points | 32 |
| Figure 26. The VSticher avatar of Andri (one of Odlo’s male fit models) after it has gone through the pipeline | 33 |
| Figure 27. The VSticher avatar of Chiara (one of Odlo’s female fit models) after it has gone through the pipeline | 33 |
| Figure 28. Feedback and current issues with the Andri VStitcher avatar | 34 |
| Figure 29. Example of the particle constraints used in ObiSkinningCloth | 37 |
| Figure 30. Example of VStitcher animation, where we can compute the distance distribution of each vertex of a garment mode | 37 |
| Figure 31. Examples of estimated garment movements of a dress (left) and a tight t-shirt (right); the red indicates a garment vertex moving farther away from the nearest body surface during the motion whilst the blue follows the linear skinning position. | 37 |
| Figure 32. Available rendering parameters in Browzwear’s VStitcher | 38 |
| Figure 33. Data pipeline diagram outlining relevant sections, partners and where the ‘Physicist’ fits in | 39 |
| Figure 34. Demonstrating the loss function inputs - comparing the rendering from Browzwear with the rendering from Unity | 39 |
| Figure 35. SnapChat Lens Studio Skeleton | 40 |
| Figure 36. Avatar with Metail Skeleton compared to Avatar with Snapchat Skeleton | 41 |
| Figure 37. Settings for exporting a garment as a multi-mesh FBX | 42 |
| Figure 38. Showing some issues with stitching the garment to a single mesh | 43 |
| Figure 39. Transferring the skinning weights from the target avatar into the garment | 44 |
| Figure 40. Converting the multi-mesh FBX into a single mesh rigged FBX | 45 |
| Figure 41. Converting the multi-mesh FBX into a single mesh rigged FBX in a T-pose | 45 |
| Figure 42. Simulating the garment on an ‘average’ male body shape | 46 |

1. Executive summary

This deliverable explains the process and state of the art of the core functionalities of the different software, which are used for the avatar-garment simulations and gives detailed information about the 3D garment collection, which is the base for the three applications of the eTryOn project. As the process of the avatar-garment simulation for the different applications within the different software is not yet finalised, we are still working on developing better ways and new processes and workflows and this deliverable describes the current state, but might change, improve and look slightly different in later versions. This report describes in the first few chapters how a 3D garment is created, what is needed so as to develop a fully virtual piece of clothing, what are the components and how they need to be put together to have something ready for the second version of this Deliverable. In the second part of this document the process of transforming garments from Browzwear (BW) files to their Obi-library-compatible form is described and also how to create avatars from the QC scan for the VR application. Finally, the last chapter is dedicated on the process of creating garments in the Snapchat lens filter using the BW-designed garments.

This deliverable describes the research work which is done for the WP2, Simulation algorithms for avatar-garment interactions which is mainly done by Metail and ODLO, in collaboration with CERTH who is leading the UI application implementation (WP5).

2. Introduction

In the eTryOn Project we are developing three different applications, which all of them provide a specific interaction between an avatar (either that of the user or of a fit model) and a virtual piece of garment. This interaction and the requirements are different for each application. The common ground for all applications is the utilization of garments that have been designed for production using professional 3D design software (i.e. Browzwear and VStitcher in our case) and its transformation in a compatible form for each application. In this direction, we first describe the design workflow and the post-processing required for all applications and then the additional processing steps for each application based on its requirements.

In the first application (VR designer app), where the target audience are fashion designers, the interaction will happen using VR glasses by the designers to view their 3D design in an VR environment on their chosen avatar which is a scanned person of their choice, mostly an in-house fitting model or person who has the average size and body shape of the target consumer of the brand. For the designer to be able to view his or her 3D design within the VR application on their fit models, the utilization of QuantaCorp-scanned avatars in the VStitcher environment is described. This is the first step in order to get accurate fitting results on specific avatars and then transfer the experience in the VR environment.

For the second application (DressMeUp app), which is targeting influencers and other daily social media users who are keen to try on garments virtually and upload either an image or video of their new outfit, there is an accuracy requirement, since the result needs to be of high visual fidelity and needs to be realistic. However, since in this case there is no real-time constraint, a server-side back end will be developed to process the input images/videos. Eventually, we will rely on the components developed for the other applications and optimize them in terms of accuracy in a second step.

The third application (Magic Mirror app), which is targeting online shoppers who want to see the 3D garment they are thinking to purchase on their body, aims to simulate the physical experience of try-ons in the digital space. For this interaction, the two main requirements is that the garment needs to simulate on an avatar's body as true to life as possible and this to be done in real-time. For this, we rely on game engines, i.e. Unity and its cloth simulation library Obi Cloth, and in this document we describe how we can transform BW-designed garments to their Unity Obi library compatible counterparts so as to have accurate garment simulation in real time. We present the approaches we have focused on and discuss which are more feasible and which are more risky to take.

Finally, we have developed a Snapchat Lens filter that can augment a garment on the Snapchat user in real time. This filter can be considered a multi-platform scaled down version of the third application, offering the real time simulations but without fitting information. In this document, we describe the process of transforming a BW-designed garment in a form that can be used in Snapchat Lens Studio (e.g. skinning, stitching), including also the skeletal changes in the utilized avatars.

3. 3D Garment Collection

All three applications need as a base virtual developed 3D garments from a CAD software. With these 3D garments the individual user of each application can react with the garment as the application allows him to do so. In this chapter we are looking into how such a 3D garment collection gets built, what it needs to get a virtual T-shirt or pants and what are the different steps such a garment has to transform through before it can be used in one of our applications.

Odlo has already more than 4 years of experience in 3D product development and came to the point where all of their seasonal, inline, cut&sewn styles are created in 3D. This journey from the old traditional physical way of developing sports garments had quite some challenges and took the company a while to figure out how to transform to a more digital, faster, efficient and sustainable way of developing and also presenting and selling their sport products. In this section, we will discuss on the main parts of the design workflow that are of interest to the eTryOn project and the requirements for the 3D garment collection to be used during the pilots.

3.1 Season of pilot phase

The eTryOn pilot phase is planned to happen from M18 till M24 which is from May 2022 till October 2022. In this time period different 3D garments from different seasons are needed for action for each different application.

For the Designer application in this time period the designers are designing their styles for SS23 and some updates on the already designed FW22 collection.

For the DressMeUp application in this time period the influencers are posting images wearing garments from the current season which are right now for sale and available on the webshop. These are SS22 styles.

For the MagicMirror application in this time period the online shoppers are using the app for trying on garments which are for sale now on the webshop and this affects again SS22 styles.

We can see here that not for all app users the same 3D garments will be used at the same time.

3.2 Garment Categories

At Odlo we have several different garment categories. This is different if we are looking at a Spring-Summer season or at a Fall-Winter season. In a Spring-Summer season we have the categories of outdoor, sports underwear, running, cycling and accessories. In a Fall-Winter season we have the categories of outdoor, sports underwear, running, cycling and accessories, but additionally we have also nordic in the range. In the following, we describe these categories and denote what kind of garments belong in each category.

- Outdoor
This is one of Odlo`s biggest categories and contains all kinds of garments which you would wear for a hiking trip, or a nice walk in the outdoors. Odlo offers a wide variety of products. Different technical t-shirts or long sleeve shirts, depending if it is summer or winter season, hiking pants and shorts, midlayers and hoodies, a wide assortment of jackets, rain jackets or insulated puffy jackets, parkas, windproof jackets, ski jackets and ski pants. Everything which lets you enjoy

spending time in the outdoors while hiking and walking or skiing depending on the season.

- Sports Underwear / Baselayer
This is Odlo's core category and where they have the most and longest skills and knowledge. In winter season, Odlo offers a lot of technical long-sleeved underwear with or without face mask and long tights which keeps you warm next on your skin while skiing or snowshoeing or in any other sports you do in the outdoors during winter time. Additionally, this category contains panties, boxers, briefs, strings, sports bras and singlets or tank tops, which can have a cooling effect if they are developed for the summer season or a warming effect if it is for the winter season.
- Running
For running, Odlo develops all kinds of garments which a runner can use, if he/she runs in the outdoors, trail running or also in the city with a lot of reflective details, to be sure that the runner can be seen. The product range is from shorts, thighs, t-shirts, tank tops, midlayers, vests over jackets and with cooling technology in summer or warming effects and insulated for the winter runner.
- Cycling
The cycling category is mainly related to summer. Odlo develops a lot of useful cycling gear if you are more of a mountain biker or you prefer cycling on the road. The products are different t-shirts, tights with and without suspenders and including different bike paddings and vests and jackets in different qualities, for example wind or waterproof.
- Nordic
In this category the garments are all specifically developed for cross-country skiing. This is only available in winter season. This category includes tights, pants, jackets and vests. Important in this category is that the garment should keep the sports enthusiasts warm while skiing, but also not too warm as they produce their own warmth with their movements.
- Accessories
The last category contains all the different items which the customer can use additionally to the garments he/she is wearing explained in the categories above. Here Odlo develops socks, hats, headbands, gloves, tubes, face masks, leg-warmers and arm-warmers.

From all other categories will be some garments offered for the project's pilot phase. Most items will belong to the running category and the outdoor category. As already said the pilot phase will be during SS22 summer sale, so we will not show any nordics or ski styles. The idea is to start with ten running styles (5 woman, 5 men), ten outdoor styles (5 woman, 5 men), ten cycling styles (5 woman, 5 men) and ten sports underwear styles (5 woman, 5 men). The sports underwear category in summer is smaller than in winter and for a start with this project garments like panties, boxers or strings aren't the best ones to test. Also bras are not recommended as this would only work if the customer would scan himself naked and try these things on, because a bra underneath or a boxer over a boxer does not give you the required accuracy and visual fidelity. These products can be included in the project when the project has gone over the first phase and is more established and has earned the trust of the consumer. Sports underwear garments which can be used in the pilot phase from the beginning are t-shirts and tank tops, which cover the body more.

3.3 Fabric Construction

The different garments mentioned in the chapter above in each category are developed and produced in different fabric constructions. It is important to understand these different construction ways, as the 3D virtual development only supports and allows to realistically

develop the garment in one of these three construction ways. The three fabric constructions are:

Cut & Sewn

The garments which are developed as cut & sewn garments are made of panels of fabric with graded pattern pieces. The exact pattern will be cut out of the fabric panel and sewn together to a wearable garment. This way is very common, but has also some disadvantages as it is for example not so sustainable. Firstly, a very long panel of fabric gets produced and only the pattern parts which get cut off that panel will be used for the garment. The smaller parts around the pattern and left of the fabric panel are useless. There is a lot of fabric waste which can't be reused. This is why it is very important how the pattern shape is developed and placed on that fabric panel in production to get the most amount of fabric out of it with as little fabric waste as possible. This placing process is called "marker making" and would be a nice additional feature in the 3D software.

Seamless

Garments which are developed with a seamless technology, here the name already reveals the purpose. The garment is not made out of a produced fabric panel with a pattern and then sewn together, it is seamlessly knitted directly in the shape or size of the pattern the garment should have its final fit and look. This means the production of the fabric itself already contains the information of the pattern and this way it is more sustainable as you don't have fabric waste.

Flat knit

This technique is a mix between cut & sewn and seamless. It is also fully knitted, as in seamless, but not directly knitted in the shape of the garment; it is first knitted flat as a rectangular surface and out of that surface the pattern pieces are cut and then sewn together as a cut & sewn product.

The software which Odlo is using to develop their virtual 3D garments is called VStitcher from Browzwear. VStitcher is designed only for cut & sewn garments as their technique and their simulation needs pattern shapes as a base. This is one reason why Odlo develops only all their cut & sewn garments virtually and none of their seamless or flat knit products. Also, in the eTryOn project we will only focus on cut & sewn products which we have in all categories available. What will also not be offered in the project are items of the accessories category as most of them are seamless, like socks or knitted in a way which makes it difficult to develop in VStitcher and not taken in consideration from Odlo to offer that item as a 3D garment.

3.4 Gender and Sizes

Odlo offers garments for women, men and children. In 3D they develop all their cut & sewn styles for both genders and the children. In this project, we will showcase our applications in adults only, including both genders. The size range for women's garments is from XS to XL (XS, S, M, L, XL) and the size range for men's garments is from S to 2XL (S, M, L, XL, 2XL). These sizes will be all also available for each 3D garment.

3.5 Designing in VStitcher

Everything starts with the designer designing a garment in VStitcher. The designer gets input from their category manager about what kind of product they should design, what is the story Odlo wants to tell to customers in that season and what are the needs of the market. With this input the designer can either start from an already existing 3D garment

from older seasons or use a base pattern, called basic block, that they can choose from a Smart Design Toolbox and start from there.

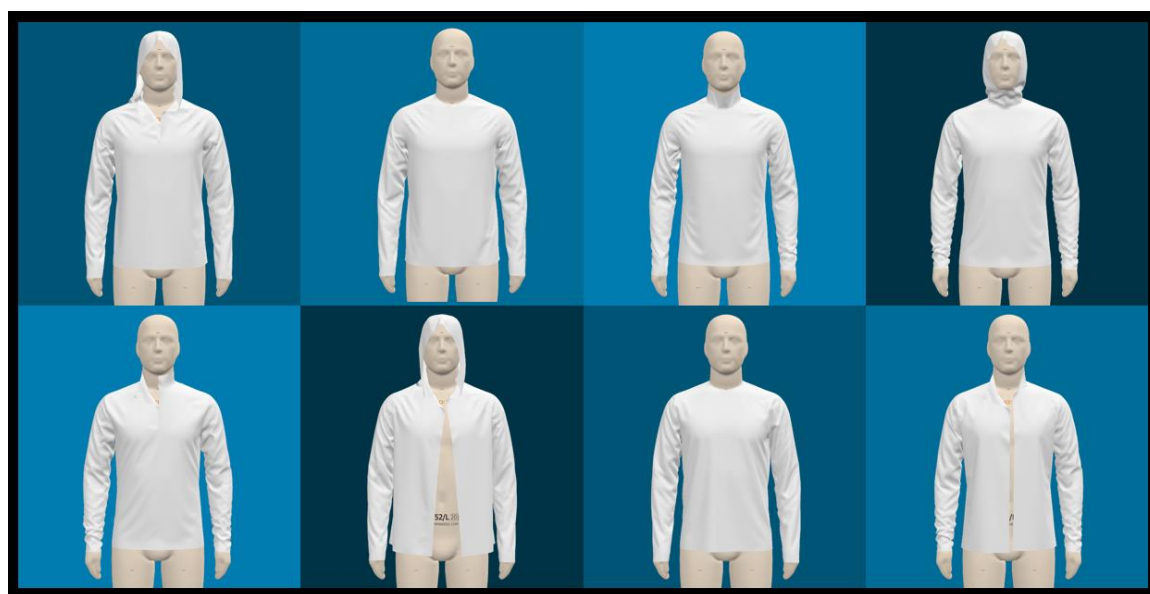


Figure 1. Example of Smart Design Toolbox with different basic blocks of men midlayers

In the image above you can see different men midlayer variations with half zips, full zips, no zips, hoods, collars or crew necklines and the designer can choose the one which is close to the planned design. After this the designer draws the design lines and new ideas directly on the 3D garment. He/she can change the base shape, add cut lines, inserts, shorten or lengthen it and more.

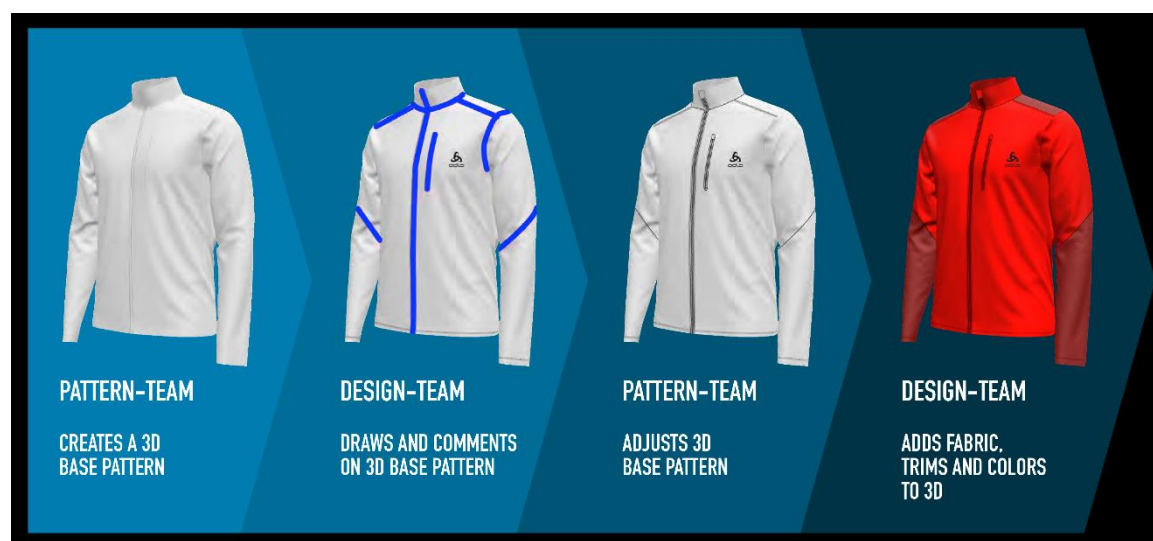


Figure 2. Steps of designing a 3D garment in VStitcher between pattern and design department

The imagery above shows the steps of the development process when the designer has chosen a white basic block from the Toolbox. It is a ping pong between pattern maker and designer, though the designer has no pattern skills, he/she draws new shapes on the base pattern and the pattern maker afterwards takes that file over and adjusts the design shapes to real pattern edges and keeps the measurement and fit accuracy consistent. This process can have several rounds, depending on the complexity of the style.

In order to visualise the garment (e.g. as in the images above), certain components are needed. As already said, there must be a pattern and also an avatar and digitized fabrics

which have the correct physical properties and the captured texture maps. With all this information together the software algorithm is able to simulate the virtual garment on the avatar if you give some more additional information like stitching and pattern piece arrangement in the cluster mode.

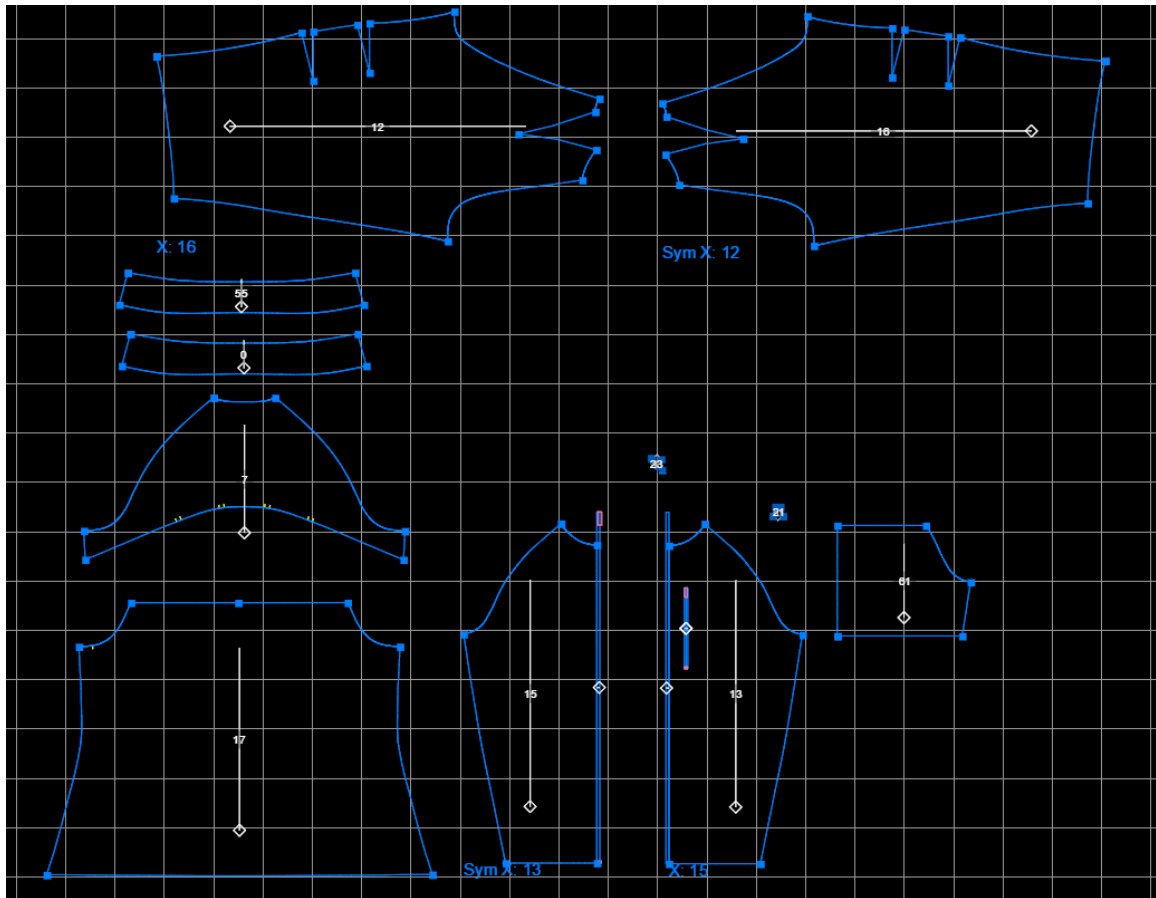


Figure 3. Digitized 100% real size pattern in VStitcher

The image above shows the pattern of a running jacket in 100% real size in the base size S for men. The pattern contains information like front zipper and pocket zipper length and placement and darts on the pre-shaped sleeve as well as the grain line and pocket bag. These separate pattern pieces need to be stitched together so that the software knows which pattern edge does belong to another edge. It is the same as in reality when sewing up real pattern pieces.

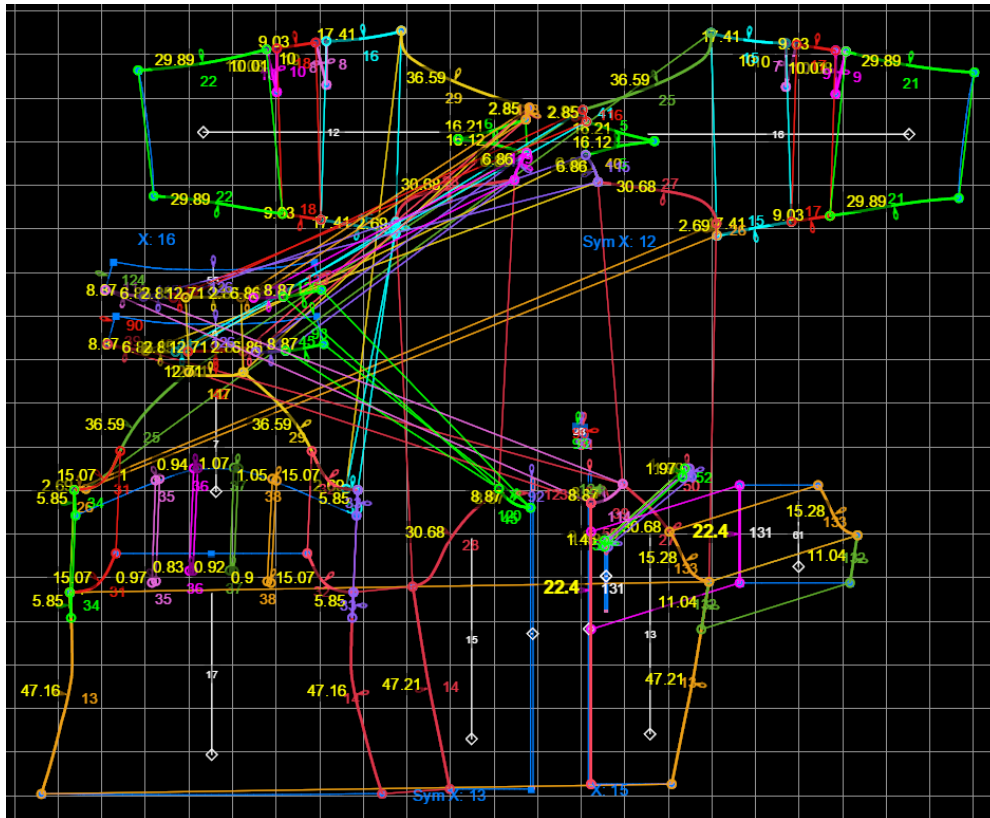


Figure 4. Digitized pattern is stitched together in VStitcher

Every stitch is colored in another color to better differentiate each single stitch seam. This is also a manual process and the user has to select each pattern edge and let the software know to which other edge it belongs and should be sewn together.

Now the software must know what kind of garment this is and where on the body does it belong, if it is top or bottom, has sleeves or not and more. For this you have to put each single pattern piece into a cluster in the arrange mode to give the software this information.

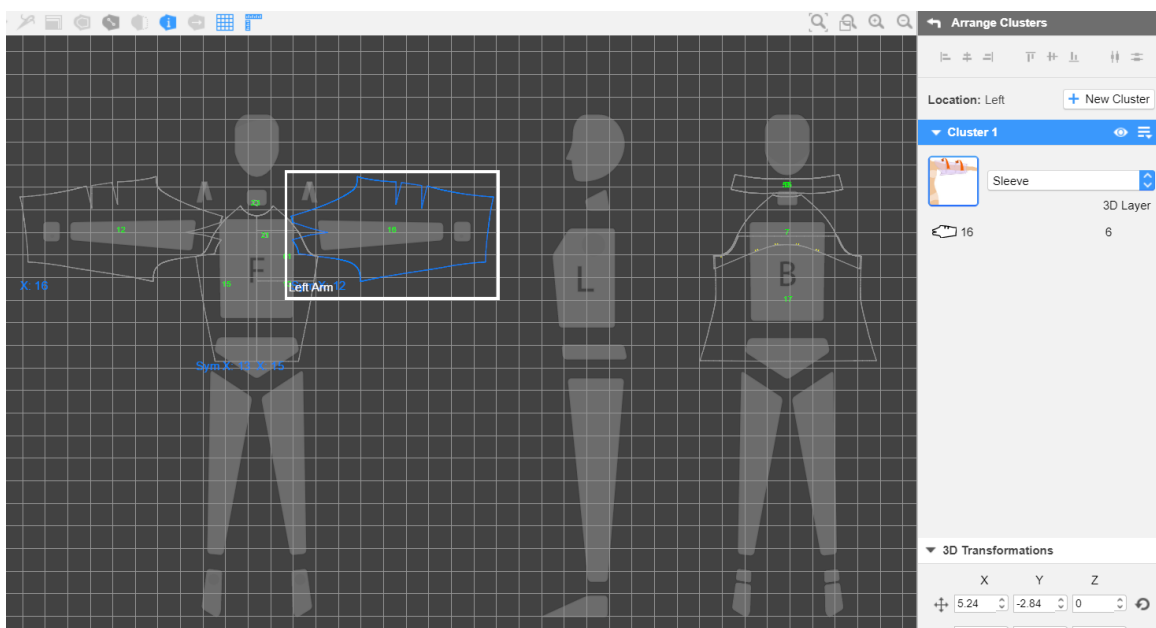


Figure 5. Arrange Mode with clusters in VStitcher

Here you can see the pattern pieces arranged on the human figure in different clusters. This needs to be done for each piece, as well little zipper garages or flaps or only inside visible pocket bags or lining. Otherwise it does not get taken in the simulation process and the garment can't be shown and falls apart or to the floor.

What we now also need is an avatar on which we can dress our garment. Odlo is using two parametric avatars, one for woman and one for man. They have the measurements of the base size, this means for woman size S and for man size L. As this running jacket is a man's jacket we need to load the man base size L avatar.

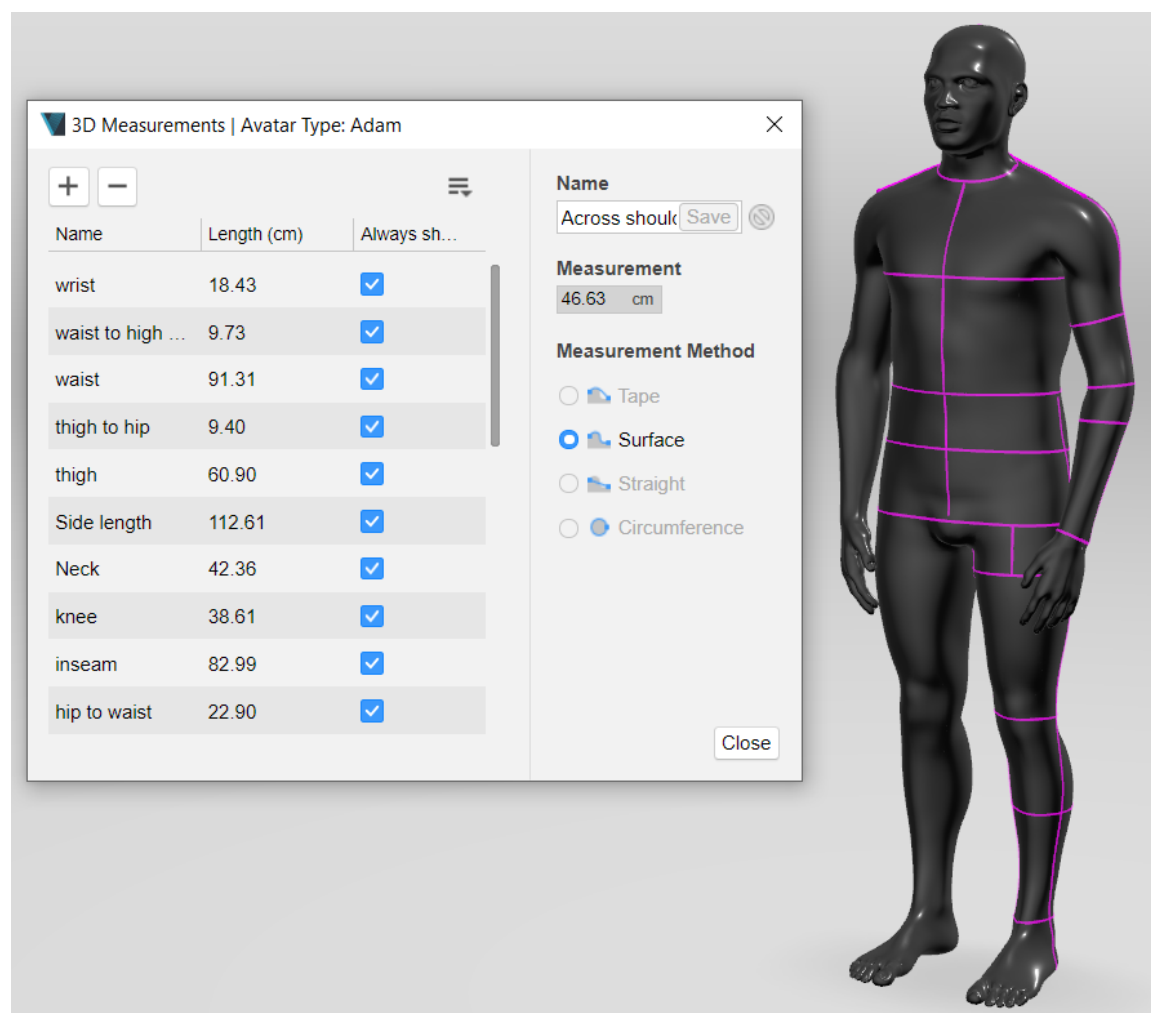


Figure 6. Parametric base size avatar man

The pink lines on the avatar are measurement tapes which give you the accurate length or circumference of the different body parts. They also are useful as guidelines to check where the garment should end or be placed, for example where the pants should stop on the ankle or where the pants should be placed horizontally on the waist. It is very important that the measurements of the avatar are accurate to be able to design it virtually. Otherwise you cannot judge the fit and look of a garment.

The last thing we need to start the simulation in VStitcher are the fabric physics and the texture maps. Odlo tests the fabric physics internally with a special machine. Also the textures of the fabric gets captured with a special lightbox, which is done internally by Odlo as well and these two things, fabric physics and texture maps need to be put together in VStitcher and exported and stored as a u3m file, which is an open sourced file format for materials, in the 3D library. From there every designer and external garment maker has access to use these digitized fabrics for their virtual 3D designs.

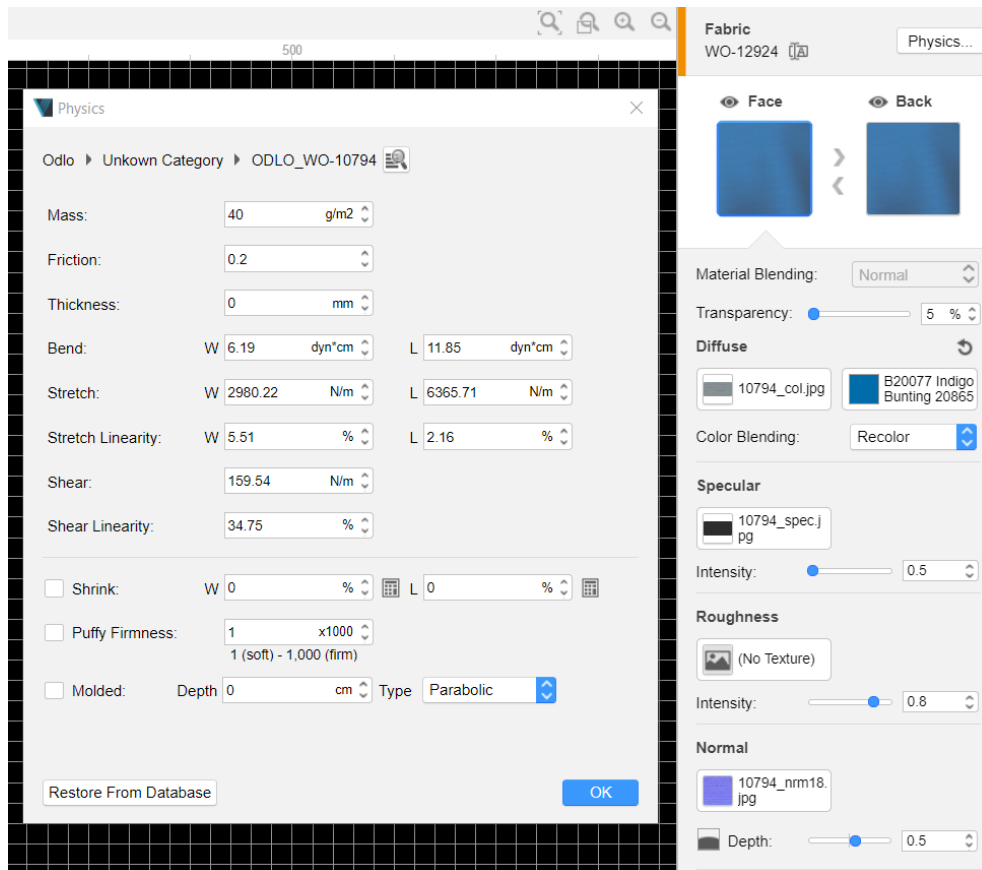


Figure 7. Physical fabric properties and captured fabric texture in VStitcher

In Figure 7. you can see the physics of this woven fabric on the left and on the right you can see the different texture maps with their settings. When this fabric is loaded and assigned to the pattern pieces which should be made out of that fabric, the software can run the simulation.

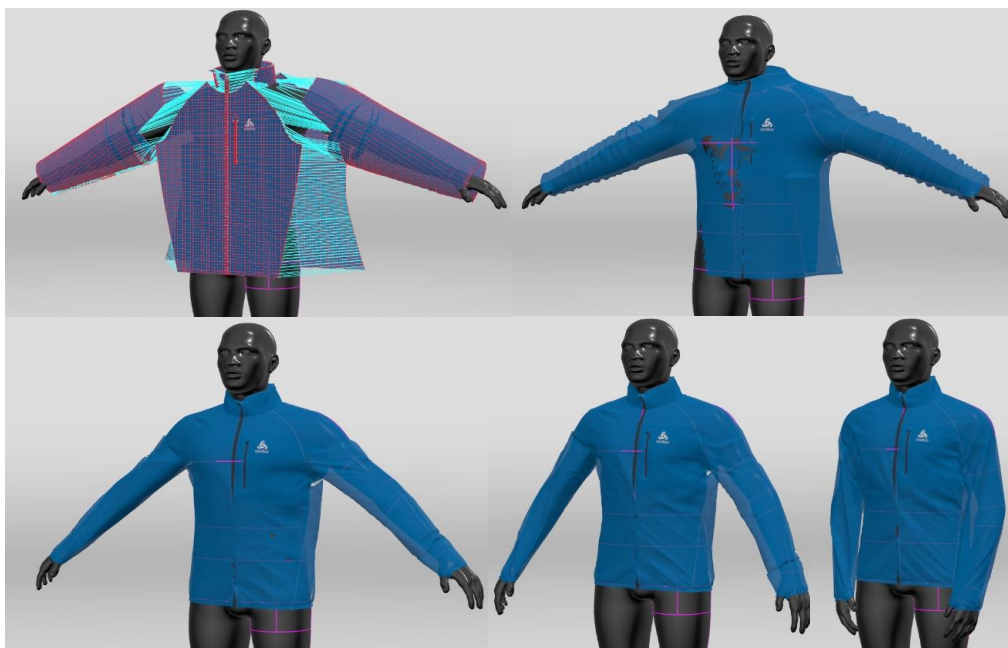


Figure 8. Automated garment simulation process in Vstitcher

The simulation can take a while depending on the complexity of the garment. When the simulation is finished most of the garments need some styling and pinching work to look nice and correct. This you can see also here in the example of Figure 8. and Figure 9.

3.6 Styling of garments



Figure 9. Automated dressed garment without manual styling

As you can see in Figure 9. the automated simulation in VStitcher does not look fully accurate and nice dressed on the avatar. The sleeves are going through the thumbs, the center front zipper is slightly twisted, the zipper puller is off and the hem is not horizontal and everywhere are draglines and wrinkles on the two front panels. The backside is also wrinkling and dragging down to the wearer's left side.

All these issues can be only solved with manual adjustments of draping, pinching, flattening and freezing parts of the garment with different tools in VStitcher. This work is very time-consuming and so far no automated solution is available. It is the same work that you would do for a photoshoot in reality on a model or on a dummy.

Actually a lot of the time working in 3D is used only for styling purposes. Of course some simple t-shirts don't take long to style manually and they look already quite nice after the automated simulation by the software. But when it comes to insulated jackets, or jackets or midlayer with hoods, you have to spend more time on styling than on everything else you did before in the software. This additional styling, which did not exist before designing in 3D, has increased the workload of everyone using 3D. Because you don't only design now, you also do a part of the marketing job, which is done during the photoshoot and this part is now earlier in the whole process and affects the designer, pattern maker, and generally everyone who is working in 3D.

In Figure 10. you can see the same jacket after manual styling process in the VStitcher software.



Figure 10. Dressed garment after manual styling

After the garment is nicely styled and wrinkles and twists and other issues are solved the garment needs to be rendered and these images will then be later used for digital product books, presentations, on garment boxes or tag labels and other channels.

4. Additional preparation of the 3D garment for the eTryOn project

So far the whole process described before is the normal workflow each 3D designed garment from Odlo goes through step by step. But for the eTryOn project the 3D garment needs to go through some additional steps, because so far the 3D garment is not graded and only developed in VStitcher in the base size of each gender. For women this means in size S and for men in Size L. But we need to have the 3D garment in each size dressed and styled so that we can export it as an FBX file. An FBX file is a 3D model saved in the Autodesk Filmbox format. This is a model which can be used and shared in various digital modeling and creation programs. They are often used in game development and animation.

4.1 Size set pattern and grading

As already explained earlier, at Odlo the 3D garments are developed only in the base size of each gender. This is to reduce the complexity of the 3D file and the whole development process. For the eTryOn project other sizes are needed to be displayed in the 3D file. Especially for the MagicMirror app where the user can choose different sizes it is mandatory to have the full information of all sizes available in the 3D file. When you have a pattern graded in different sizes it is called a size set. This means the pattern is available in smaller and bigger sizes of your base size, and the way they are smaller or bigger is defined with the grading. For this you have grading rules or grading tables which define the distance from each point to the same point in another size. VStitcher from Browzwear offers a Sizes and Grading tool, but it is very basic and not comparable with the tools famous 2D CAD pattern software are offering like Lectra or Gerber. The functions are very limited and still have some teething problems and are also not commonly used by Browzwear customers. The reason for that is because it is a very complex topic and many of their customers aren't so experienced in 3D yet and they are still in the beginning of learning the software and the tools. Also the grading tool itself is not so self-explainable in VStitcher and has its limitations compared to other software which are only for pattern making. Another very important point which we should also not forget is that the knowledge of grading is slightly disappearing on the brand site. Many brands don't do their own sizes and grading for their styles, they have it outsourced to the factories. And the knowledge of the pattern makers which are working for a brand is not as good as it was in regards to 20 years ago in terms of grading. They only do the base sizes and send this pattern over to the factories and the factory creates the sizes and the grading, adds shrinkage and linings and pockets and prepares the pattern finally for production. The graded size sets we use for this project are created in another 2D CAD system and brought back into VStitcher. At Odlo we still have the knowledge of sizing and grading and we also define the size and grading rules, but we do not grade all our patterns. The work is split between inhouse pattern makers and pattern makers from our factories. But also Odlo pattern makers are not grading the pattern in VStitcher, the pattern will be exported as dxf file from VStitcher into Lectra and graded there and afterwards as dxf brought back into VStitcher. Unfortunately the reimport sometimes has some bugs and does not properly translate the grading from Lectra and there are mostly some manual adjustments in VStitcher needed. This is something Browzwear is aware of, but has not a solution to offer at the moment.

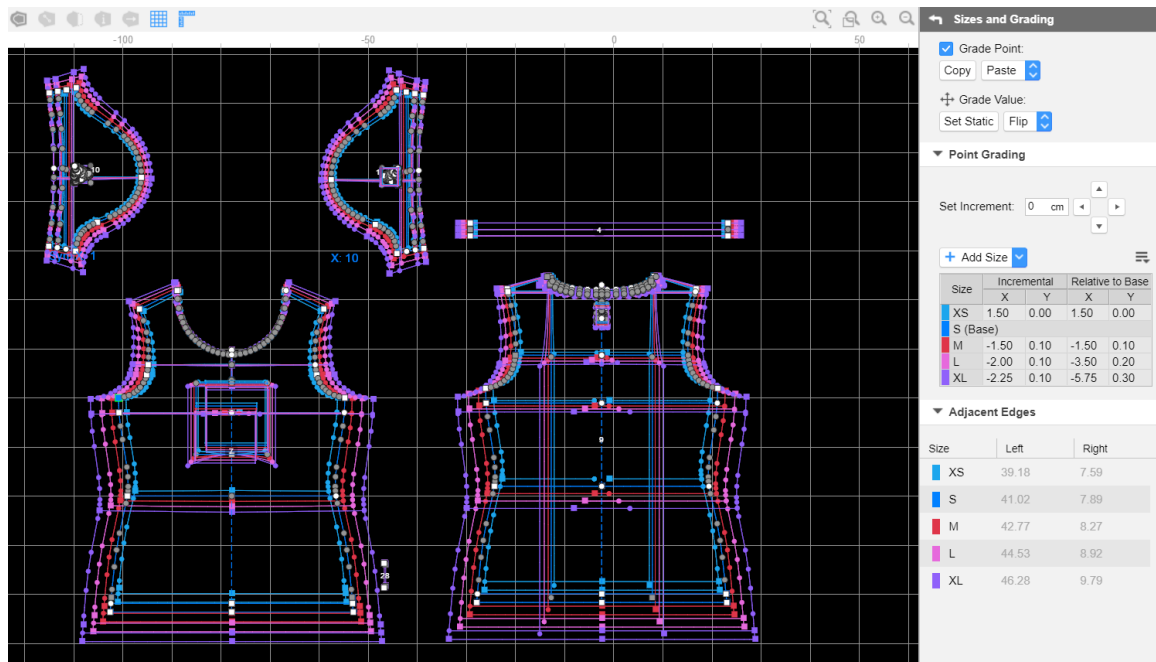


Figure 11. Graded pattern in VStitcher, also called a nest or nested pattern

In Figure 11. you can see a graded set-in t-shirt of a woman style. The base size S in dark blue with the grey points and the white points which have a grading rule applied. The other sizes are displayed in other colours like red (size M), pink (size L), violet (size XL) and light blue (size XS). The good thing is that you need to sew up only the base size in VStitcher and the software automatically translates your updates to the other sizes as well. This saves a lot of time.



Figure 12. Dressed 3D garment in Size XS, S and M from left to right



Figure 13. Dressed 3D garment in size L (left) and size XL (right)

4.2 Size set avatars

To be able to visualize the garment in all different sizes in VStitcher it is also needed to have the avatars in all these different sizes, or let's say body shapes. It does not make sense to dress a pattern in size XS on an avatar which would only wear garments in size XL. This means for every size of the garment you need a corresponding avatar on which you can dress the garment. For the project we have used the fitting models Odlo is checking their garments on their bodies for a while. Every garment size has a person which has a specific size, mean measurements and body shapes which is an average of the Odlo customer in that size. There is for every size a standard which is covered by a real human who gets measured regularly to be sure the body shape and measurements maintain as close as possible to that standard. These avatars are scanned fitting models and then imported in VStitcher. For more information about the scanning process of people and how to prepare them for being ready to import in VStitcher please read the D1.1 and Section 6 of this document.

The avatars are in an FBX format and can be easily imported into VStitcher. In the software the different body parts are marked with different anchor points you define manually on the body. These points are very important for the simulation of the garments as they define where is which body part on the avatar and this is related to the cluster mode where you define which pattern piece belongs in which cluster on the body.

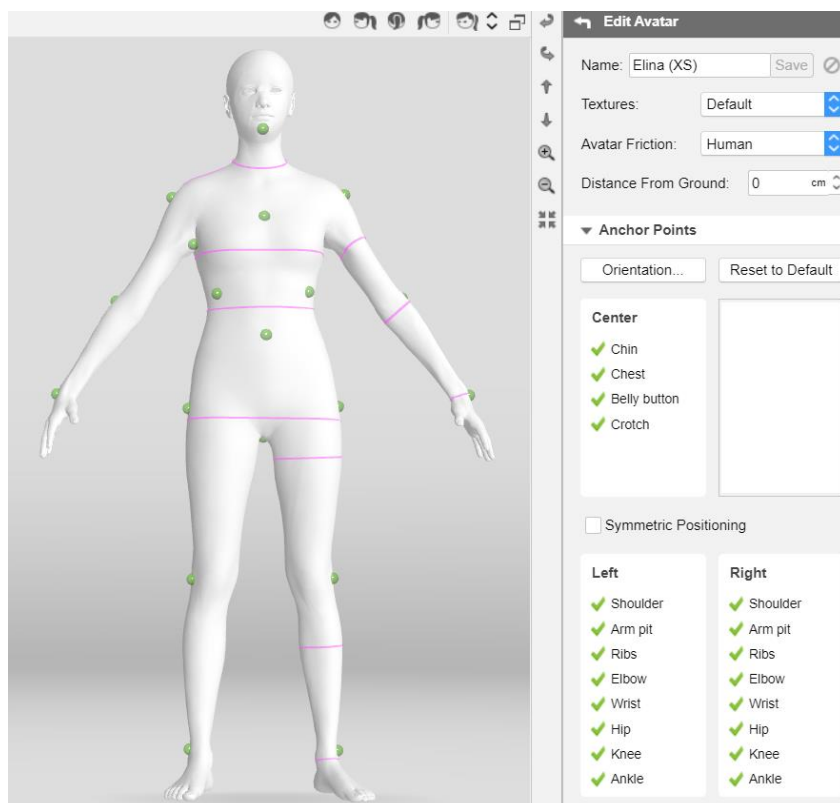


Figure 14. Scanned avatar imported as fbx file with anchor points in VStitcher

As you can see in Figure 14. the green points are the already mentioned anchor points and the pink lines are some measurements taken from the body to cross check the size. The avatar is in the A-pose, the same pose as the person got scanned.

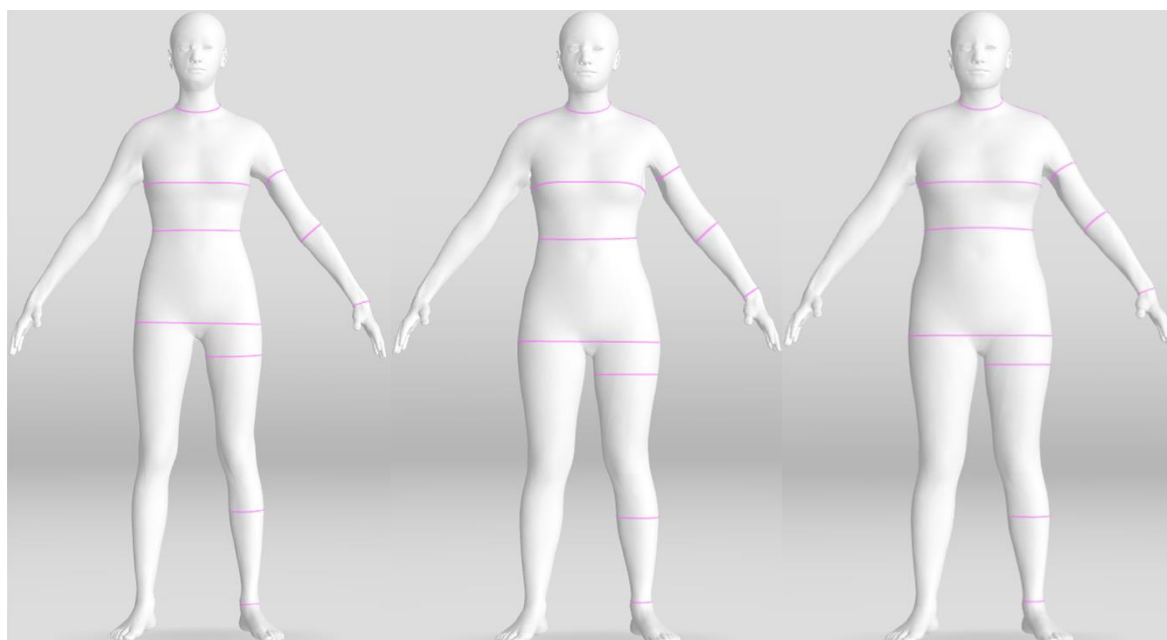


Figure 15. Scanned avatars in Size XS, S and M from left to right

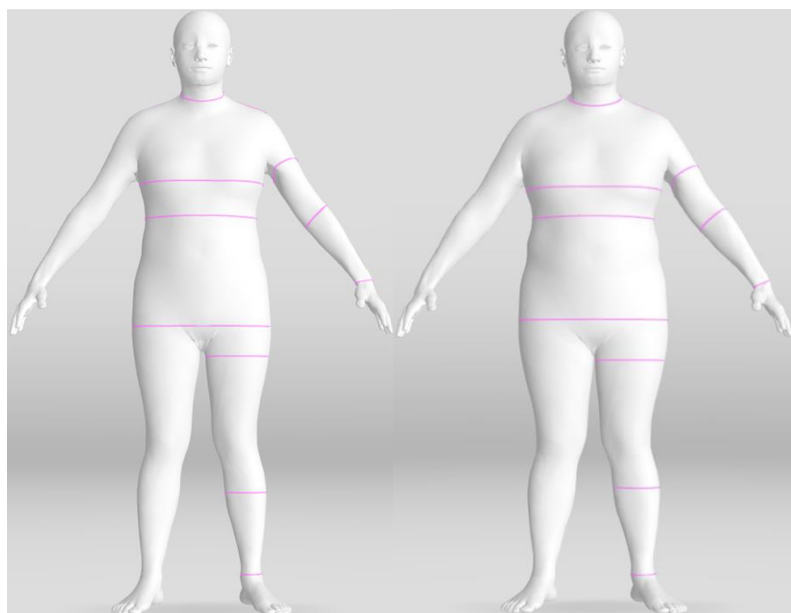


Figure 16. Scanned avatars in Size L (left) and XL (right)

In VStitcher the different avatars can be put together to an avatar size set and when the user can let the system know which pattern size belongs to which avatar. And when the user is loading a size to work with, it is automatically dressed and visualised on the correct avatar and you can quickly jump from one garment size to another and preview the garment, check the dressing, fit, artworks, logos and trims etc.

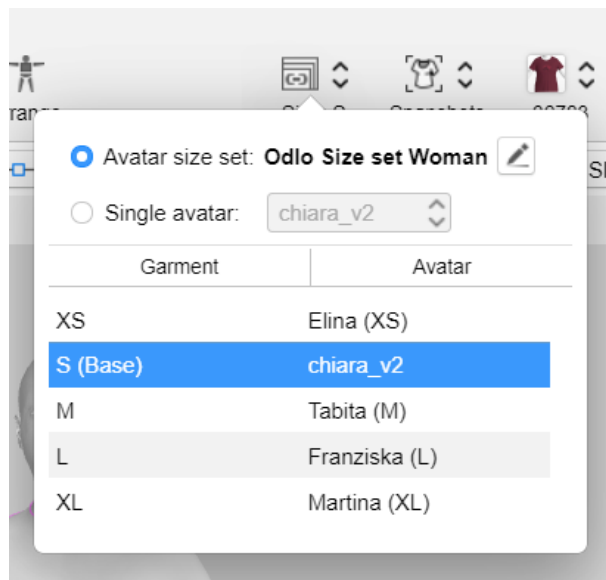


Figure 17. Avatar size set organizer in VStitcher

In the BW file you have now all the information you need about this one style. You have the different sizes in which the garment is available and produced, you have it dressed on the matching avatar for each size, you have the full information about the materials, the colours, the trims like artworks, logos, bindings, labels etc and also the stitches.

4.3 Saving and exporting 3D garments

The 3D VStitcher file is the starting point for the garment simulation in eTryOn which is explained in more detail in the next chapters for the various applications. So far there is no additional saving or export modification defined. But as the whole garment simulation for all three applications is still under work in progress we can't define a final way to save it at the moment. At the moment, the garments are simulated on VStitcher in 3D and this 3D instantiation is exported in an FBX file. This might change and develop during the project lifetime.

5. Exporting Browzwear-designed garments for game engines

As described above, the Browzwear VStitcher software is designed to create 3D garments from patterns - it is used by professional garment designers, technical specialists, and pattern makers for designing garments which will be physically created and sold.

For the eTryOn use cases, we intend to use the accuracy and detail from the design software but transfer it into AR/VR use-cases (i.e. MagicMirror app, Snapchat filter and potentially VR designer app). For the MagicMirror app, since it will be built using Unity (a 3D platform designed for VR and games), we need a way to map the output from VStitcher into Unity without losing the key characteristics from the design software. Importantly, we don't want to lose the drape and mechanics from the original garment - especially as the use case will show animation when the user is moving.

In summary, the technology we build will take a 3D digitally designed garment and transform it into a Unity compatible 3D asset - allowing rendering and animation that is true to how the physical garment would move in reality.

The figure below summarizes the data flow to get a garment design in Unity. The first step is to prepare the garment and select the correct export parameters. VStitcher can export meshes and materials for rendering, but no physical properties for simulation. We need to develop a tool to rig the garment, that is, adding a skeleton to it, and to estimate its physical properties so they can be understood by Unity and ObiCloth.

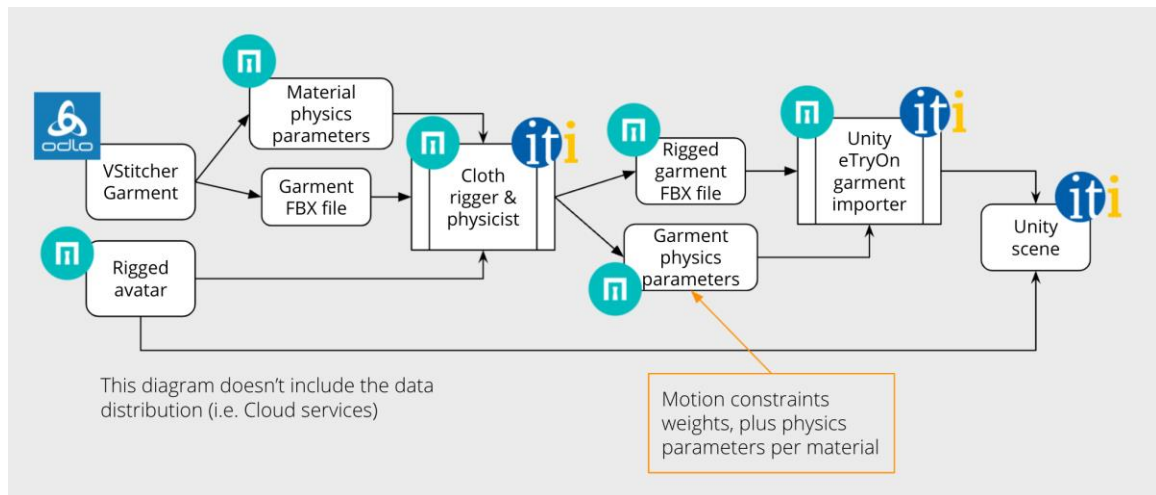


Figure 18. Pipeline from Garment to Unity Scene

Since we are targeting mobile devices, there are some constraints that we need to take into account when exporting garments. For garments to be very detailed, they use textures of the order of hundreds of Megabytes per garment, and very detailed geometry as well, in the order of hundred-thousands polygons. It is hard to define a data budget without actually testing the final assets, but we looked at Lens Studio documentation (<https://lensstudio.snapchat.com/guides/3d/3d-software/blender-3d-object-export/>) to get an idea of what the target should be, since Unity does not state clearly a data budget. They recommend keeping the whole scene under 10,000 triangles, and the number of joints for skeletal models under 100 joints. For the size of textures, the limit in mobile GPUs is around 4096×4096 pixels for a single texture.

To reduce the number of polygons, we can adjust the grid size of the cloth simulation within VStitcher, the maximum being 4cm. The figure below shows the impact in the number of polygons of different grid settings.

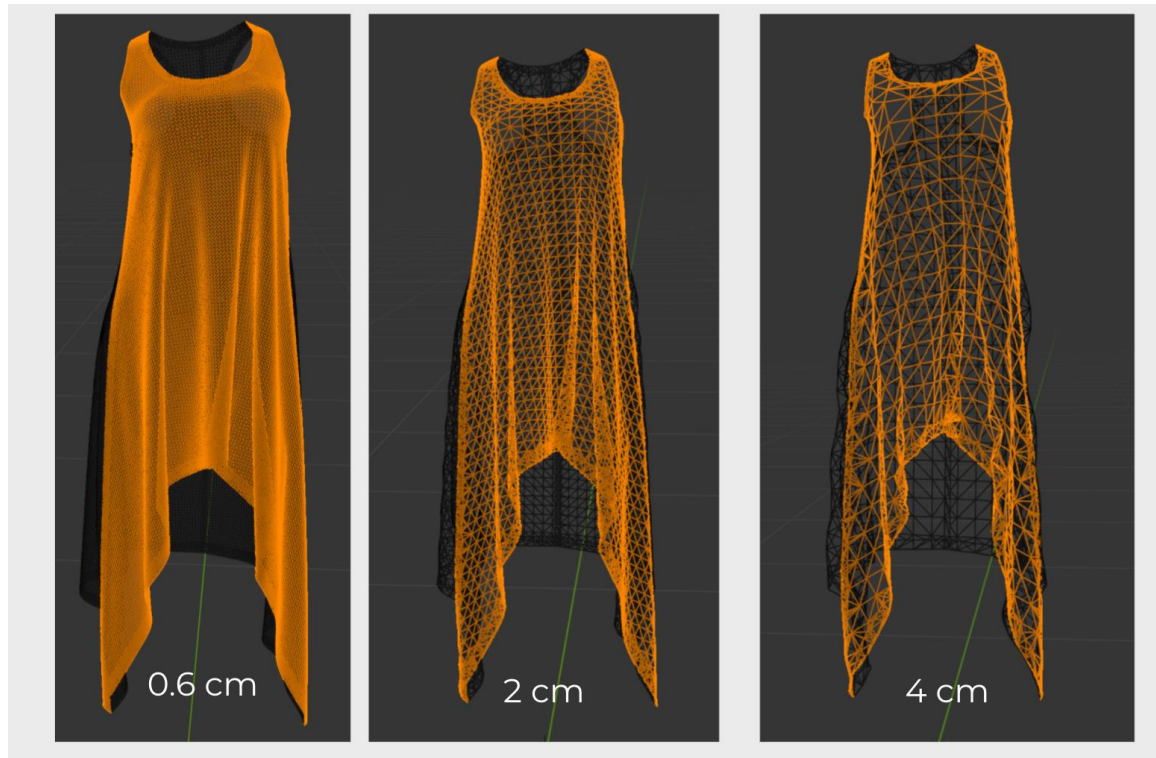


Figure 19. Impact of different grid settings on number of polygons

Another thing we can do to reduce the number of polygons is avoid exporting interior faces. This is an option in the VStitcher garment exporter.

To reduce the texture size, we can select the output DPI during export, up to a minimum of 50 DPI. Also, we can group all the pieces together into a single UV layout. See the figure below for a comparison of the different UV layout options. The preferred choice of a single UV layout produces one texture per material layer, i.e. one albedo texture, a normal texture, a specular texture, and a roughness texture (there can be more textures depending on the materials).

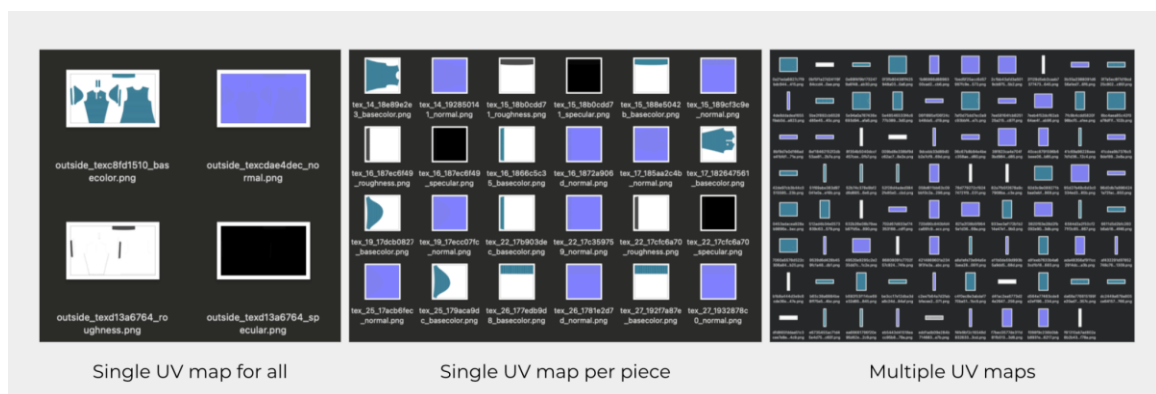


Figure 20. Comparison of different UV layout options

6. Creating avatars from mobile scans, ready for use in VStitcher and eTryOn apps

We are aiming to help designers share their designs in an innovative way (VR), showing realistic fitting of the digital garments on photorealistic 3D avatars.

Technical designers tend to work to a shape specific to their brand. Often they will have 'fit models' which match this shape and garments are designed on. In this project, we intend to provide a way for designers to create an avatar using a mobile scan which will enable them to import that fit model into VStitcher, then subsequently dress and create imagery.

6.1 Combining with avatars from QuantaCorp Scan

In eTryOn D1.1 we describe the work to create the core functionalities of the mobile application and software for creating scans of a person. In this work package we cover the work to combine the output of those scans with the 3D garments described above.

Metail's proprietary technology is used to take the output of the Quantacorp scan (which is an OBJ file), and create a 'Metail scanatar', which is a rigged avatar. This rigging is important because it provides the skeleton necessary to provide the ability to repose (and therefore animation).

There are a couple of manual steps in the creation of avatars in Metail's production pipeline. However, for the eTryOn project we want to apply a fully automated pipeline. The steps for a fully automated pipeline using QC scans as input are as follows:

- Fix scale and axis
- Extract landmarks
- Clean up scan to avoid webbing artefacts
- Mesh fitting and auto-rigging
- Create VStitcher avatar (FBX file)

These steps are described in more detail below.

6.1.1 Fix scale and axis

By default, the scan does not come in a format that is required for Metail's pipeline. The scale is unknown and the axis of rotation can be inconsistent. To make this compatible, we use the real height of the model to ensure the scale of the model is consistent. With the height as input, we have written a script to fix this compatibility issue. The script simply applies an affine transform that rotates the axis, and transforms the scale of the model to meters. If we measure the distance between the lowest vertex of the mesh and its highest vertex, along the vertical axis, and divide that distance by the real height of the person, we obtain the scale. See Figure 21:

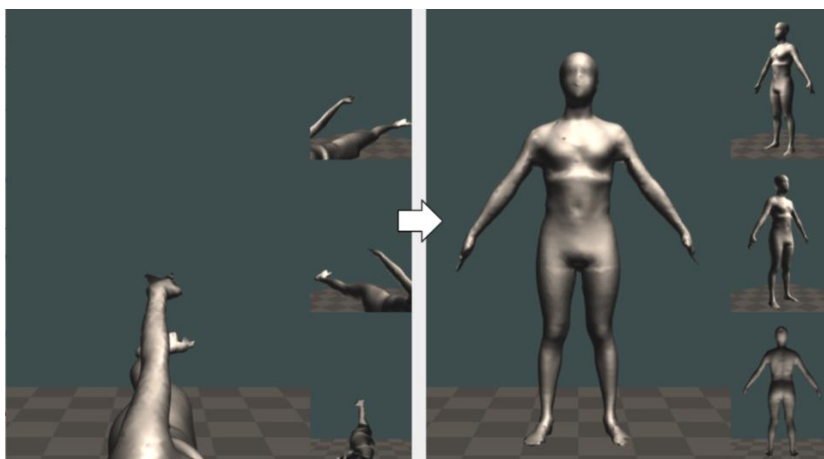


Figure 21. The scan avatar before and after the scale and axis is corrected

6.1.2 Extract landmarks

The input scan OBJ does not contain landmarks (points of interest) on the body. These are necessary for Metail's mesh registration & rigging pipeline, and can also be used to take measurements. These landmarks are now automatically generated using SizeStream software. Their software runs in a lambda function in AWS. We call that lambda function and obtain a list of landmark points on the body mesh.

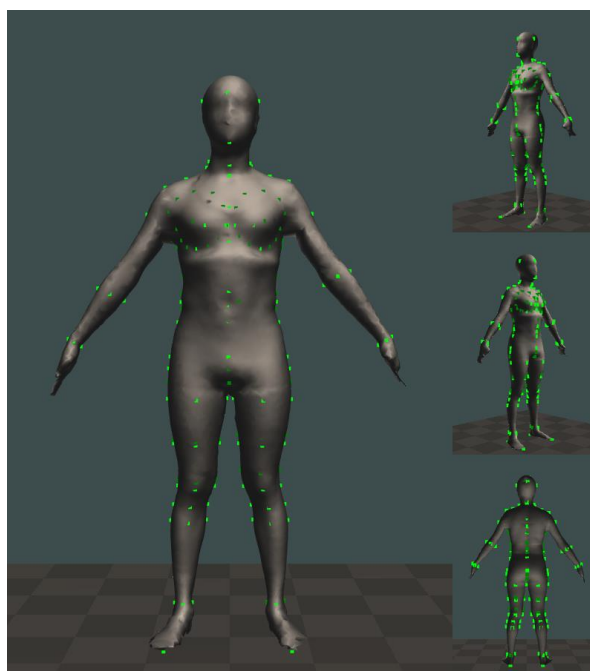


Figure 22. Landmarks, marked in green, added to the avatar

6.1.3 Clean up scan

The scans often have artifacts which can make dressing difficult. This is most often exemplified in the armpits. Because the scan lacks detail around this area, it is easy to end up with 'webbing' artefacts in the 3D, where part of the upper arm is glued to the torso. We can't automatically clean up all the problems in the mesh, but we automated the removal of vertices around the armpit area. Because we have labelled landmarks that we extract per mesh, we simply define a vertical projection from those landmarks along the

vertical axis, and remove any triangles in that area. The mesh fitting in the next step will fill up again any holes that we create.

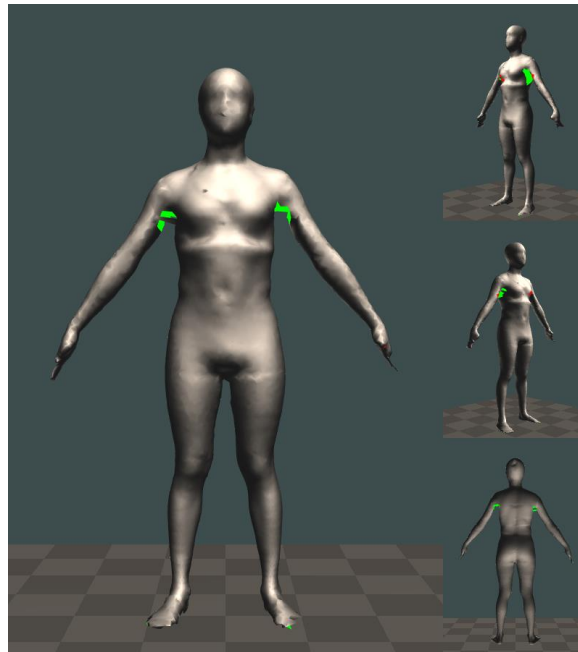


Figure 23. Showing where 'webbing' artefacts have been cleaned up in the model scan

6.1.4 Mesh & skeleton fitting

The next step is to fit the mesh and skeleton to the avatar. This is necessary for reposing the avatar and for animation. Metail has technology which automatically fits our preferred skeleton into the supplied avatar mesh.

The algorithm requires a watertight template model. The model is already rigged, that is, it contains a skeleton and skinning weights per vertex. This template model can be created manually, since it only needs to be created once. Then, given a target model scan, we use the non-rigid Iterative Closest Point (ICP) algorithm to morph the template into that scan. The landmarks are used to drive this process. The resulting mesh is also watertight, since it has exactly the same topology as the template.

The skeleton fitting step is an optimization process that tries to obtain the same mesh we just obtained through mesh fitting by applying translations, scalings, and rotations to the joints of the template model. Once it converges, the resulting skeleton is applied to the mesh.

The automated fitting output looks as follows:

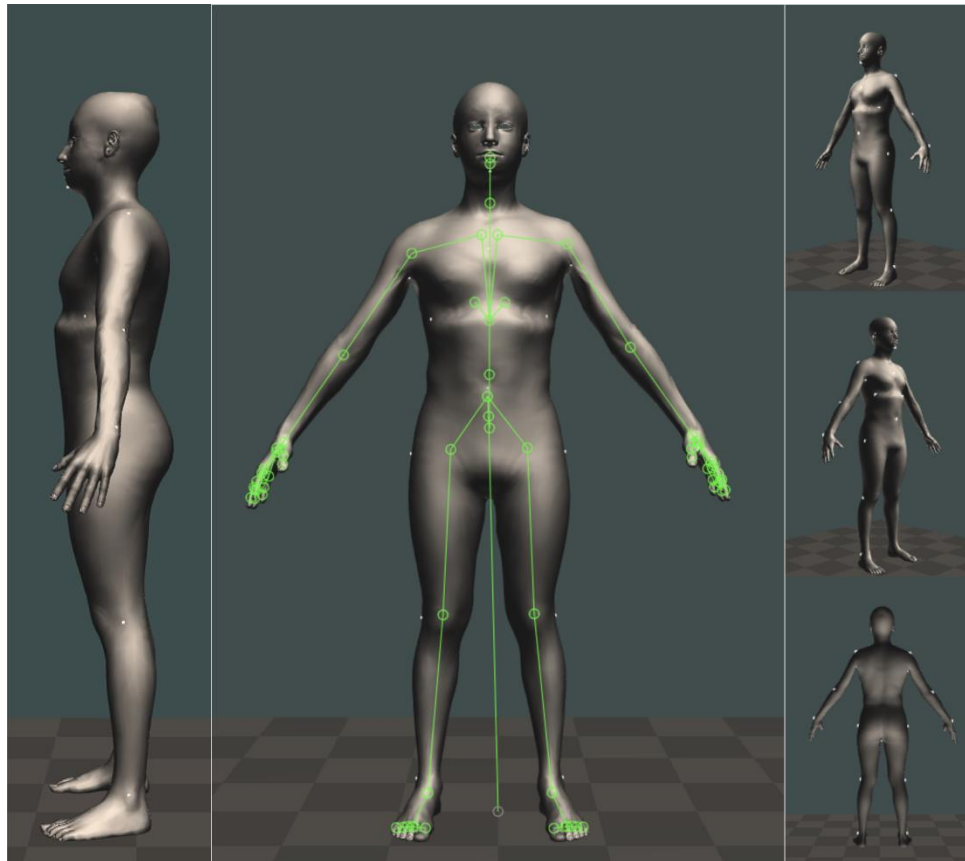


Figure 24. The avatar after the skeleton has been fitted

6.1.5 Create VStitcher Avatar

The final step is to make the avatar VStitcher-ready. This adds the VStitcher anchor points to the file so that it is ready to import into the 3D software. VStitcher uses these anchor points to know where to 'hang' the different pieces of a garment and dress the avatar. We manually place these anchor points over our template avatar, following the specification from VStitcher software. The specification consists of a series of illustrations of where to put those anchor points. See the figure below. Because we place them on the template model, this needs to be done only once, since there's a per-vertex correspondence with all the models that we generate through mesh fitting.

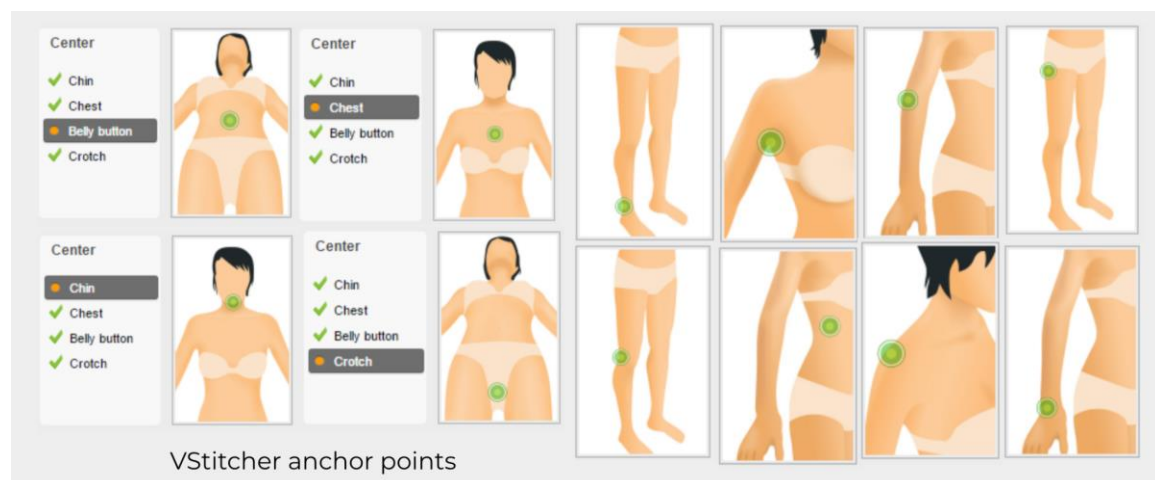


Figure 25. VStitcher Anchor Points

6.2 Demo of end result

The slides below show two of Odlo's fit models created using the automated pipeline described above.

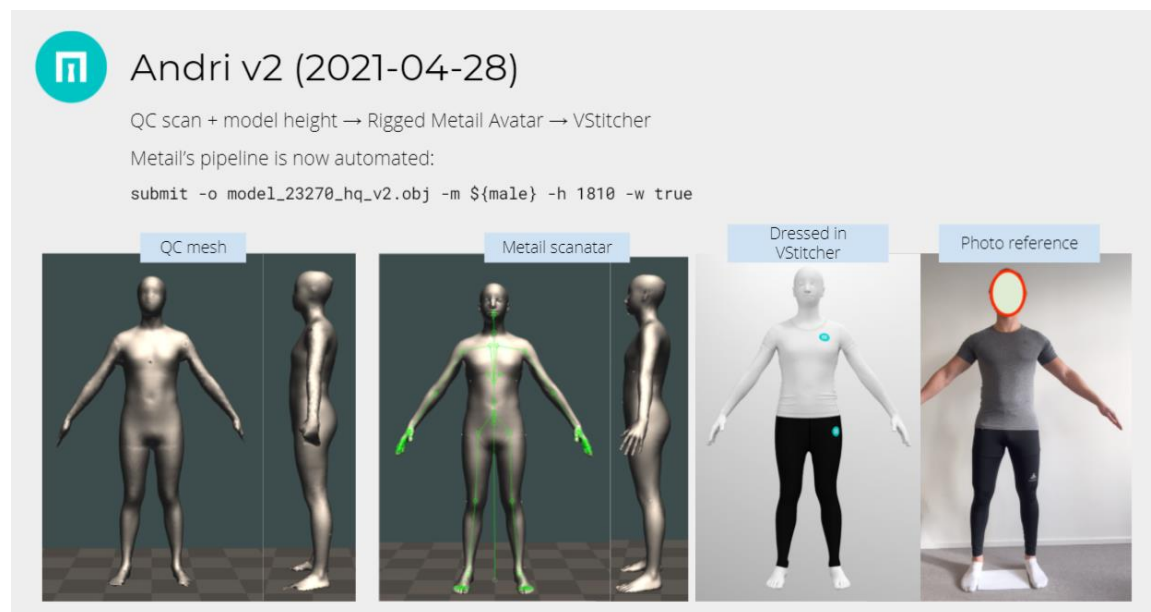


Figure 26. The VSticher avatar of Andri (one of Odlo's male fit models) after it has gone through the pipeline

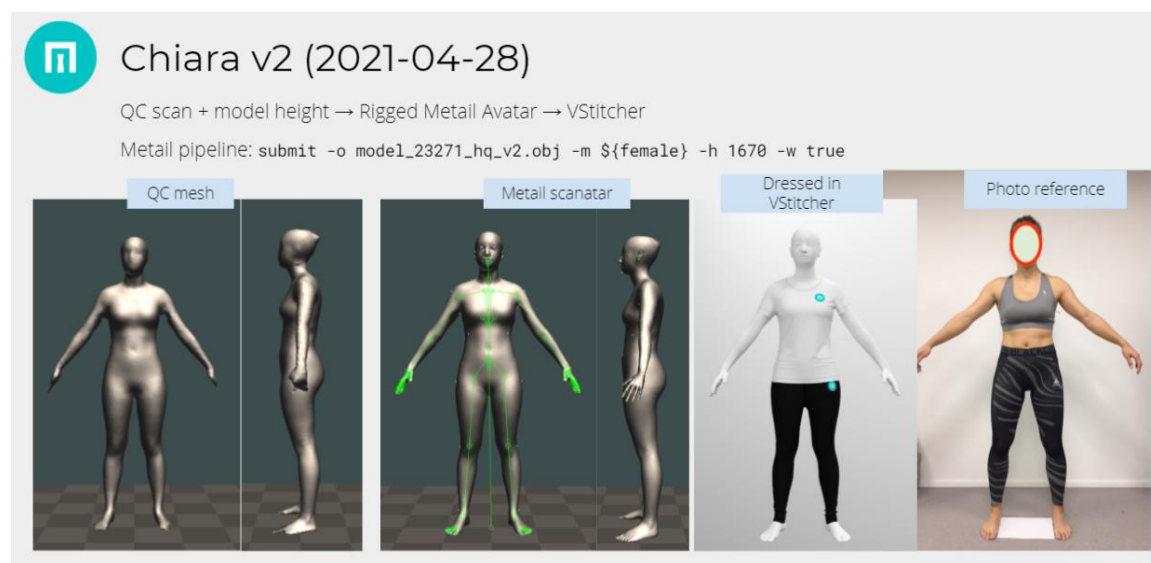


Figure 27. The VSticher avatar of Chiara (one of Odlo's female fit models) after it has gone through the pipeline

Current Status

The current pipeline allows an automated way to go from the scan generated as part of WP1 to an avatar which can be dressed in VStitcher. The dressed avatar is then used as input into the use case applications as described in sections 5 and 7 of this document. Currently the mesh fitting part of the pipeline can take around 90 minutes to complete. For the designer use-case this is not necessarily a problem, but it is an area with scope for further research and improvement.

There are also a couple of open questions about the quality of the avatars generated. This assessment about the differences between the fit model and avatar is possible because the avatar is now available in VStitcher. The Figure below shows some of the current issues in the final output - these issues are being investigated as part of future iterations of WP1.



Figure 28. Feedback and current issues with the Andri VStitcher avatar

7. Combining the Garments and the Avatars

In sections 5 and 6, we describe the creation of animation ready 3D garments and avatars respectively. In this section, we will describe our first investigations into combining these assets together. A particular challenge we have found in this area is the compatibility and differences between skeletons in different software. We will describe some of these challenges and how we have overcome them below.

7.1 Garment rigging and skinning

An animation is just a description of changes along a timeline. For a 3D object, there are mainly three ways of transforming its triangle mesh to create an animation:

1. Animation through affine transforms, which are usually rigid. With rigid transforms we can move or rotate a character or a garment. With a more generic affine transform, we can also scale the garment up or down.
2. Animation through skeletons, attached to the character or garment. The rigid transforms (sometimes with scaling as well) are applied per limb. The process of attaching a skeleton to a mesh is usually referred to as *rigging*.
3. Animation through morphing of the mesh, that is, moving each vertex of the mesh separately and storing its new location, or by describing its change through special functions. In 3D modelling software like Maya, you can create these morphs using something called Blend Shapes. But these morphs could come from a Machine Learning module as well (e.g. a neural dynamics model).

The game engine, Unity in this case, can apply affine transforms if the scale needs to be adjusted to match the target scene, or if the garment needs to be resting on a table instead of standing. But in order for the garment to move along with the character, we need to rig it, that is, add a series of connected joints that form a skeleton, and skinning weights to it.

A skinning weight is a number from 0 to 1 associated with a particular joint. You can have more than 1 weights per vertex, and the sum of all them must be equal to one. What that weight says is how much each joint contributes or affects each vertex.

In section 6 we describe the process we introduced to skin garments. A skinned garment can be animated, but it will look stiff without cloth simulation. The next section describes how to make the garment appear more realistic in real time.

7.2 Animations for Use Case 1 - VR Designer App

For the VR designer application, one of the requirements is to provide animation - to show how the garments would drape and move during movement. We are still in the process of deciding the best way forward for this, and have two possible approaches:

1. Use the ObiCloth simulation described in section 5
2. Export the animations from VStitcher directly as Alembic files

The advantage of the first approach is that it means we can share the same pipeline as the other use cases and there will be consistency for the garments across the applications. It also enables flexibility in the animations because the user would be able to upload animations from which the output could be generated without having to re-simulate and export in VStitcher

The advantage of the second approach is that the simulation would be using the native PBS from VStitcher and the 3D designer would be able to easily preview the animation whilst using the software at the design phase.

We are in the process of assessing the quality of the output produced and requirements from Odlo. We will be investigating the pros and cons of these approaches in more detail as the project continues before deciding on the final methodology to create these animations.

7.3 Proof-of-Concept (manual process) to create ObiCloth garments from Browzwear files

Obi is an external library for Unity available for purchase, which is an advanced particle-based physics engine capable of simulating a wide range of deformable material behaviors. Of particular significance for this project is ObiCloth which is designed for cloth simulation.

- Our first approach is to use ObiCloth as a version of real time Physics Based Simulation (PBS), and develop a deep learning (DL) model or optimisation tools to figure out the best ObiCloth parameters to match rendering results from VStitcher.
- A secondary approach would be to create a new deep learning model for garment simulation at different poses and body shapes (called “neural dynamics model”). This could then replace the VStitcher PBS in the project.

The first approach looks more feasible but we predict that this result will not match the alembic animation from VStitcher. One of the reasons for this is that ObiCloth does not support collision detection for skinned garment models. This means the intersection between a garment and a body model will be approximated by a few particle properties, which sometimes result in unwanted clipping artefacts. We should also be aware that the backend PBS of ObiCloth (called “Oni”) basically approximates a garment model with a sparse set of particles, so that the final rendering result will not match the result from full featured PBS renderers (e.g. VStitcher or Clo3D).

The secondary approach is more risky. However, we see more value in this direction as this can work around the time-consuming PBS process with a more efficient data-driven solution, which will be a crucial requirement for implementing real time applications. A key challenge of this is learning the relationship between garment movement and underlying body motion and body shape effectively. As this is an emerging research area, we have found no mature technologies we can directly adopt at the time of writing.

7.4 Prediction of Garments Physics parameters for Unity-Obi Cloth

We have developed a range of ideas to tackle the problem of estimating proper ObiCloth parameters. Ideally, we want to replicate the PBS result from VStitcher with ObiCloths by tweaking a few sets of ObiCloth parameters.

7.4.1 Optimisation techniques

The first idea is developed based on simple optimisation techniques. Assuming that 3 skinning parameters will be good enough to control the general behaviours of an ObiCloth skinned garment (i.e. skin radius, backstop, backstop radius in the fig below), this approach tries to figure out best values for those parameters based on a reference animation produced by VStitcher.

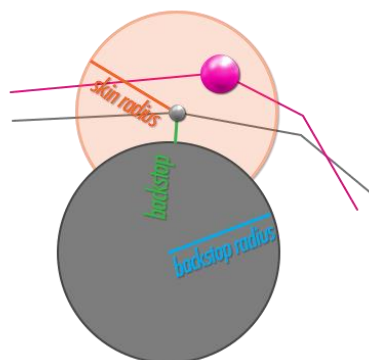


Figure 29. Example of the particle constraints used in ObiSkinningCloth

(<http://obi.virtualmethodstudio.com/tutorials/skinconstraints.html>)

The most important parameter in this case will be the skinning radius, which defines how freely a single particle can deviate from the skinned position. Since we can export the full 3D animation of a body and garment model from VStitcher as our reference, we can directly compute the distance distribution of each garment vertex during the animation (see Figure 19.). This information can be converted into a probability map and used to define initial skin radii for optimisation.



Figure 30. Example of VStitcher animation, where we can compute the distance distribution of each vertex of a garment mode

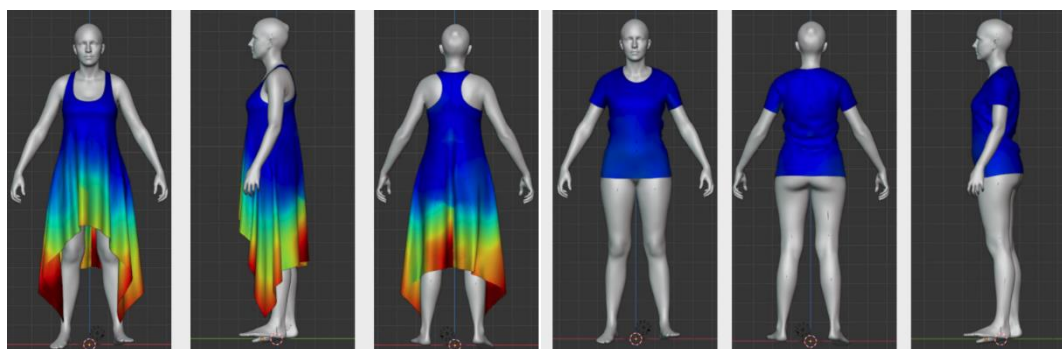


Figure 31. Examples of estimated garment movements of a dress (left) and a tight t-shirt (right); the red indicates a garment vertex moving farther away from the nearest body surface during the motion whilst the blue follows the linear skinning position.

The remaining engineering question is how to design a data interface for ObiCloth, so that we can pass our estimation to Obicloth as a form of a texture map. ObiCloth supports this but we need to assign non overlapped UV texture coordinates for each vertex of a garment model, which is sometimes tricky as VStitcher produces a single garment model in multiple sub-meshes.

A more challenging question is how to define optimal backstop radii to avoid any body intersection whilst complying the computed skinning radii. Since ObiCloth does not support collision detection for skinned animation, this will be a critical constraint to avoid the clipping artifacts. Our current approach to solving this problem is to use a simple optimisation technique, such as updating backstop radii at each iteration to minimise body interaction.

7.4.2. Rendering parameter mapping with DL

The second idea is more about developing a machine learning model to learn the nonlinear mapping between VStitcher physics parameters and ObiCloth physics parameters.

Figure 21 shows a set of parameters that potentially affect the Physics Based Simulation in Browzwear:

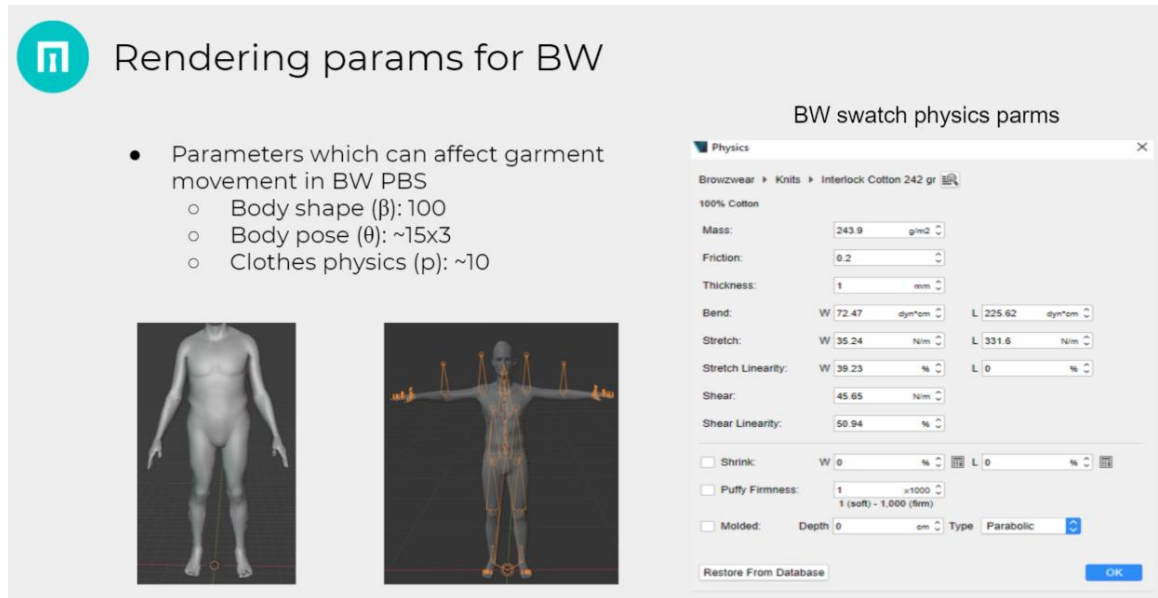


Figure 32. Available rendering parameters in Browzwear's VStitcher

We can also identify similar parameters in ObiCloth. These include per-vertex particle parameters (i.e. backstop, skin radius and backstop radius as mentioned earlier), and some garment specific global parameters (i.e. stiffness and friction). However, we do not know how these parameters are related to VStitcher parameters.

Thus, our aim in this approach is to train a machine to learn this nonlinear mapping between a ObiCloth parameter vector and a VStitcher parameter vector. This can be nicely embedded within the previously developed data pipeline (see the processing block named "physicist"):

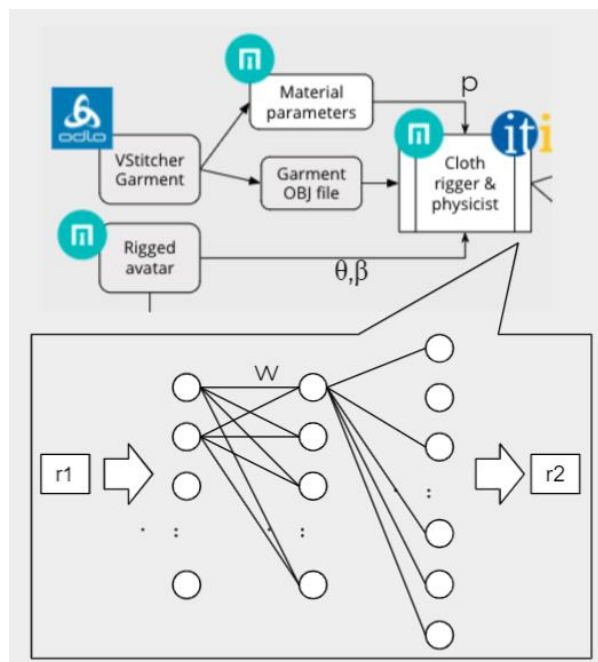


Figure 33. Data pipeline diagram outlining relevant sections, partners and where the ‘Physicist’ fits in

Assume there is a model (f) which can map the rendering parameters of VStitcher (r_1) to the rendering parameters for Unity ObiCloth (r_2), i.e.

$$f(r_1; w) = r_2,$$

where w is the trainable weights, and r_1 and r_2 are rendering vectors. The weight (w) will be found by minimizing a loss function of the garment model produced by VStitcher and Unity respectively (see fig.).

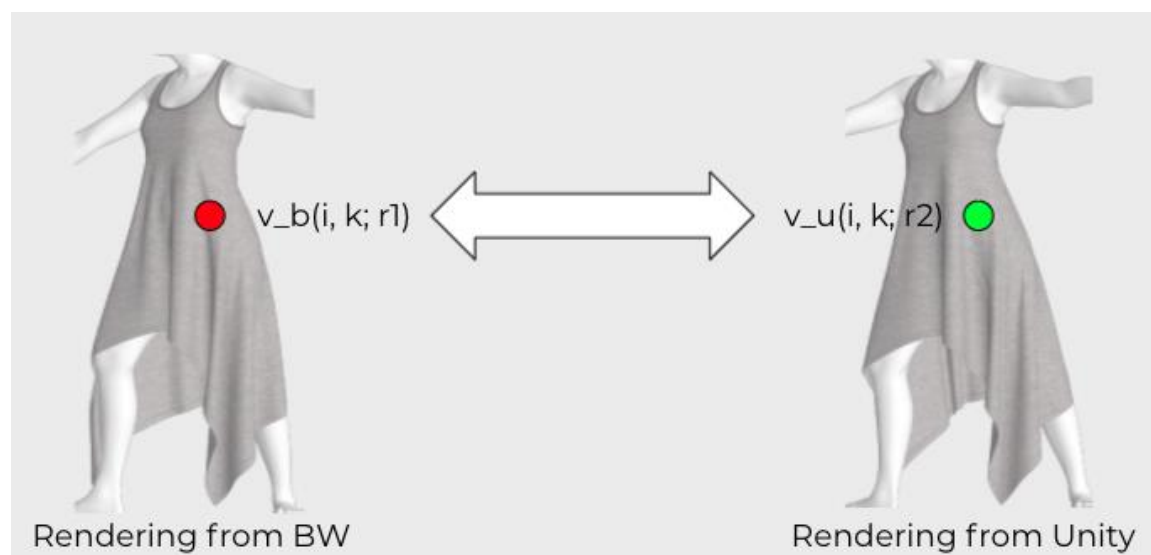


Figure 34. Demonstrating the loss function inputs - comparing the rendering from Browzwear with the rendering from Unity

The main challenge of this approach is populating data during training time. To train this model, we ideally need to evaluate a garment model at each iteration from VStitcher and Unity from an input rendering parameter. However, this does not seem practical and will require considerable computing time. We are currently investigating alternative solutions to avoid online garment model creation.

7.5 Proof-of-Concept (partly manual) process to create garments in Snapchat lens filter from Browzwear files. Technology towards Use cases 2 and 3

Lens Studio by Snapchat is a framework designed to build augmented reality experiences for Snapchat. Because it has many built-in features, including custom shaders and body tracking, it is a good way to quickly get to a demonstration showing the goals of the Magic Mirror app (use case 3).

There were several steps required to get a working demonstration using our avatars and VStitcher garments into Lens Studio. In this section of the document, we will describe the work to make this work.

7.5.1 Matching the Snapchat skeleton

To make use of Snapchat's 3D body tracking, we need to match the Snapchat skeleton:

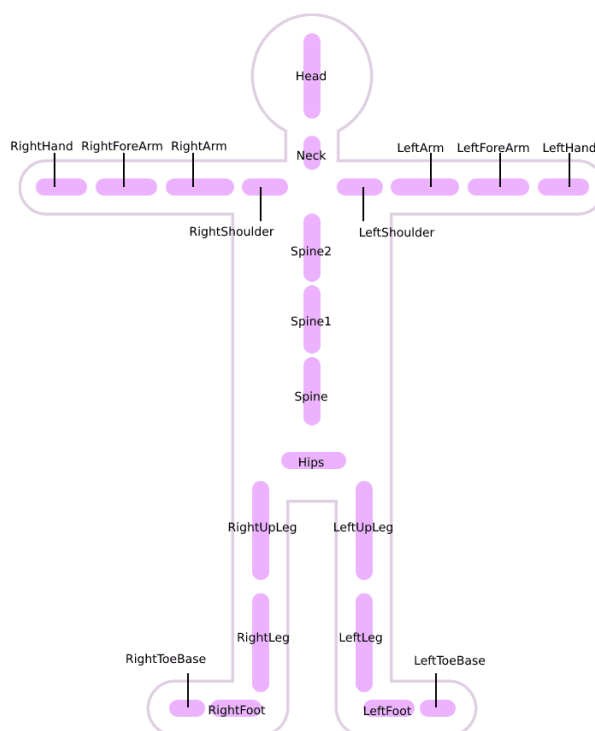


Figure 35. SnapChat Lens Studio Skeleton

This involved the creation of a command line tool capable of modifying Metal's rig to match Lens Studio specification. The command line tool includes functionalities to match the skeleton in hierarchy, orientation and naming, and also recompute the skinning weights. We created our own format to describe these transforms, which simply describe actions such as "remove left toe joint", "reparent abdomen joint", or "rename chest joint to Spine2". When a joint is removed, so are all its children. The skinning weights of that joint and all its children need to be reassigned to the parent joint. If we reparent a joint, we remove it, but attach any children it may have to its parent. The skinning weights of that joint are also assigned to its parent.

After applying this transformation, the template model looks as in Figure 36:

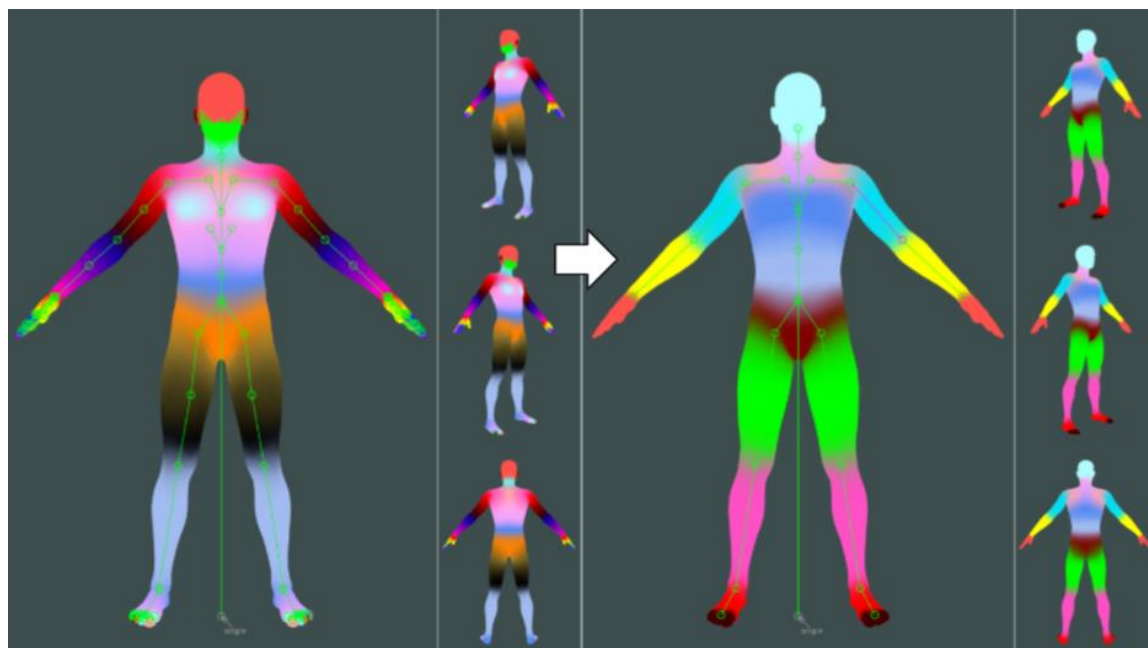


Figure 36. Avatar with Metal Skeleton compared to Avatar with Snapchat Skeleton

Joints in the figure are represented as green circles, and bones are the green lines that connect those circles. You can see that there are fewer bones in the avatar on the right. For instance, all the fingers have been removed. This is now compatible with Lens Studio. The color illustrates the different skinning weight areas. For instance, the arms on the source model have 4 joints, each one affecting a different section of the arm. You can therefore see 4 distinct colors, smoothly blended because a vertex is affected for more than one joint. The arms of the final avatar on the right have only 2 joints, so there are only 2 distinct colors.

7.5.2 VStitcher Garments in Lens Studio

The first step is to export the garments from VStitcher as an FBX. VStitcher provides various options for export. In section 5 we described the geometry and UV export options, and our choice to keep the output file within budget, as recommended by Lens Studio. A summary of all the export options can be seen in the figure below.



Figure 37. Settings for exporting a garment as a multi-mesh FBX

There are some outstanding problems still to be investigated at this stage (circled in red in figure 33):

1. Some of the materials look incorrect when viewed in Blender
2. The inside faces seem to be exported despite this option not being selected
3. The alpha of the logo hasn't been applied to the texture

We need to work with Browzwear's support team to resolve these issues.

The next stage is to stitch the exported multi-mesh FBX to a single mesh. In the FBX, the UVs are not continuous because they belong to different pieces of the garment. If we collapsed 2 duplicated vertices into a single vertex, that would introduce UV discontinuities as illustrated in the figure below.

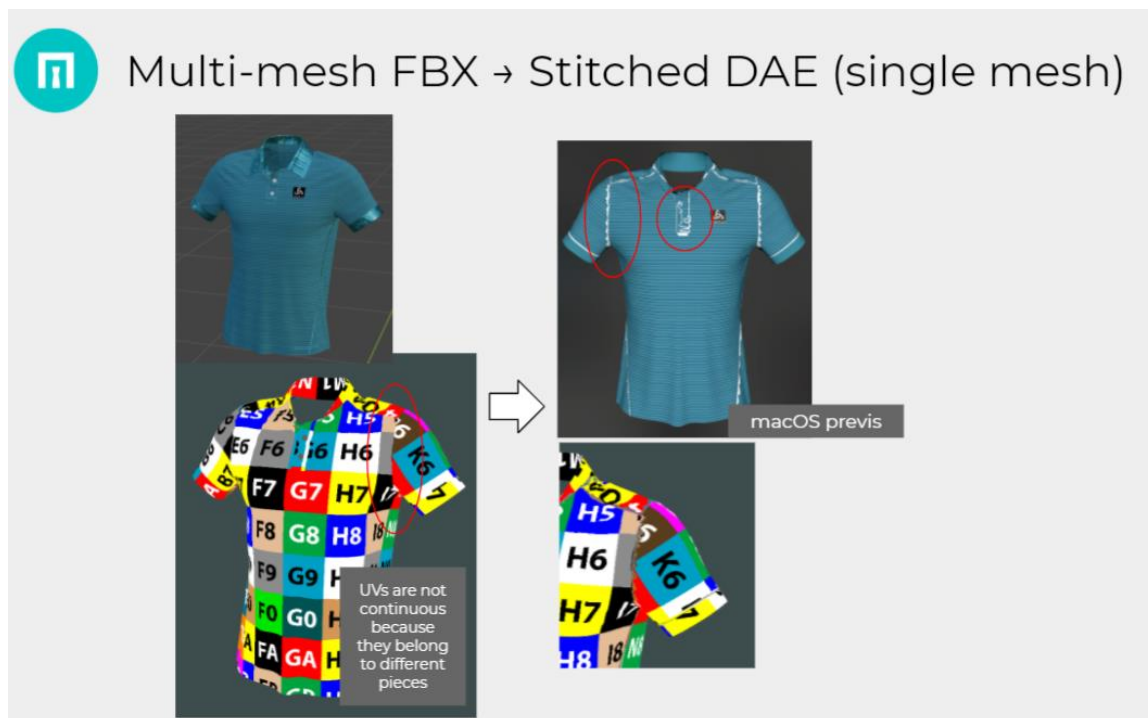


Figure 38. Showing some issues with stitching the garment to a single mesh

To address this, we've kept the duplicated vertices in a single mesh.

Once we have a stitched mesh, we need to rig the garment so it can be animated in Unity and Lens Studio, as explained in section 5. To rig the garment, we combine its mesh with the Snapchat skeleton described previously. This will enable the garment to be reposed and animated. Metail have written a command line tool which will do this:

1. reorient bones to match the Lens Studio requirements, as explained earlier; the transform associated to each joint must contain a rotation that points to its parent;
2. transfer the skinning weights from the target avatar into the garment, which is done by aligning the garment with the source model and copying the skinning weights from the avatar, which is already skinned

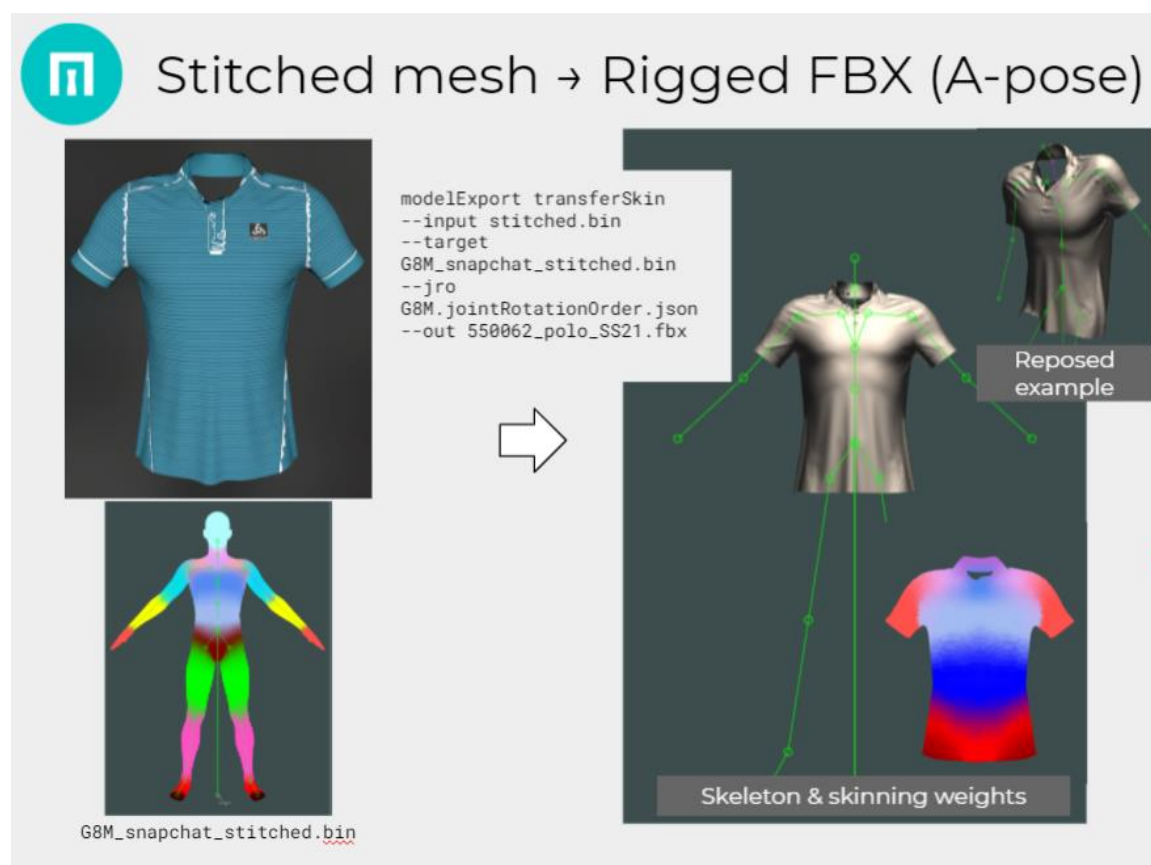


Figure 39. Transferring the skinning weights from the target avatar into the garment

Although it mostly works as expected, there are some issues in this process that we also had to overcome:

- UVs, normals, and materials are lost during stitching. We have had to add UVs, normals, and materials support to our pipeline, using the FBX SDK, and modified the command line so it can group submeshes based on the material assigned to them.
- The skinning weights are not smooth. This produces some artefacts when reposing. To address this, we've introduced a K-nearest neighbour smoothing process: for a given vertex in the garment, instead of simply copying the closest skinning weight from the avatar, we look for K nearest vertices in the avatars and average the result. Note that it doesn't completely solve the artefacts around the armpit.
- Lens Studio expects a T-pose as the bind pose, so sleeves are wrongly positioned in Snapchat (the wrinkles created when stretching the arms out would look unnatural in most other poses).
 - To help address this we have introduced an option to change the bind pose on export, so the garment can be simulated on a different pose. We apply the skin transfer using the pose used for simulation, and then repose the final model to a T-pose. See Figure 37.

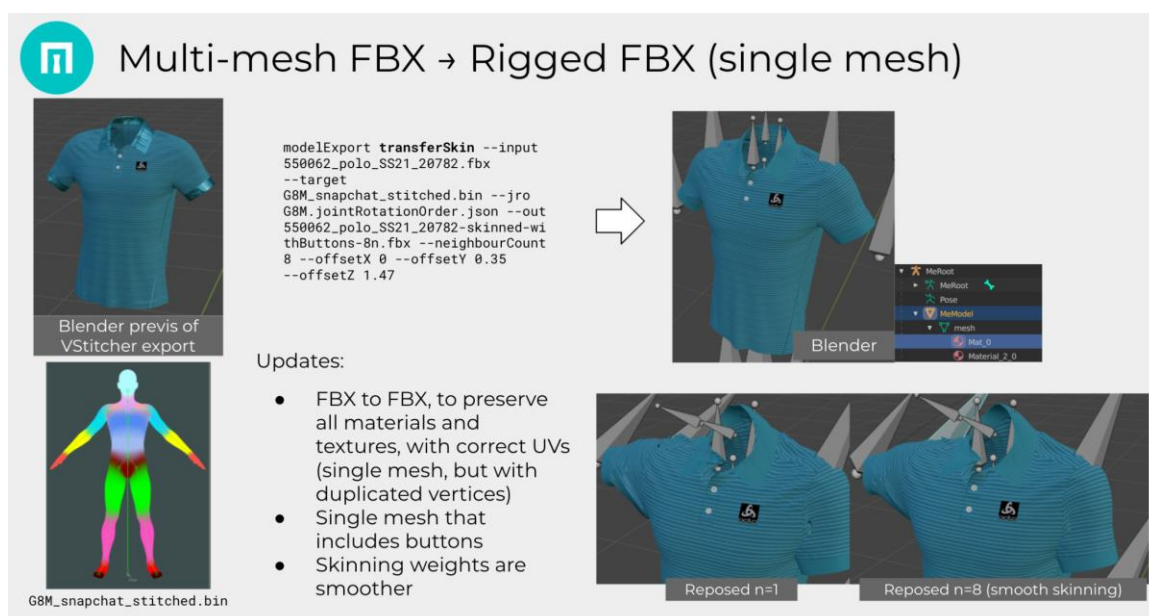


Figure 40. Converting the multi-mesh FBX into a single mesh rigged FBX

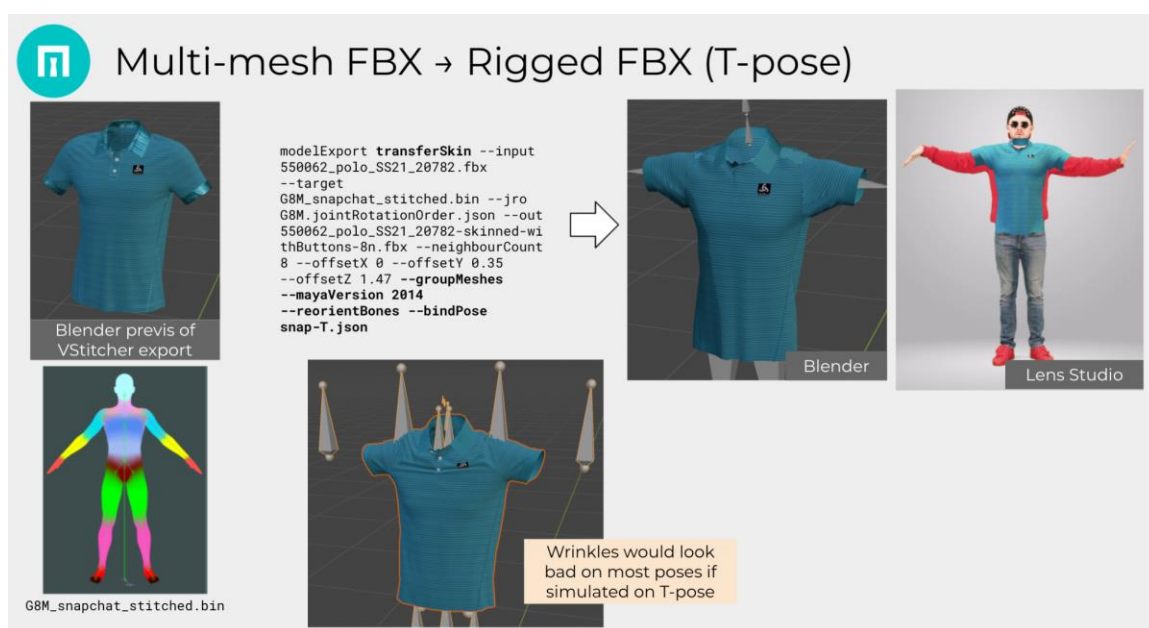


Figure 41. Converting the multi-mesh FBX into a single mesh rigged FBX in a T-pose

The result wasn't universally satisfactory because of the shape of the muscles that you can see in the avatar from the underlying mesh, so we decided to create an average female and an average male model based on Metal's parametric avatars technology.

The average body shapes are extracted from the CAESAR body scanning survey (<http://www.shapeanalysis.com/CAESAR.htm>) population, of around 4000 subjects. These parametric avatars have yet another different skeleton and a different bind pose, a T-pose, so we had to create new avatar templates and register these templates against those parametric avatars. This registration process is basically the mesh fitting process described in section 6.

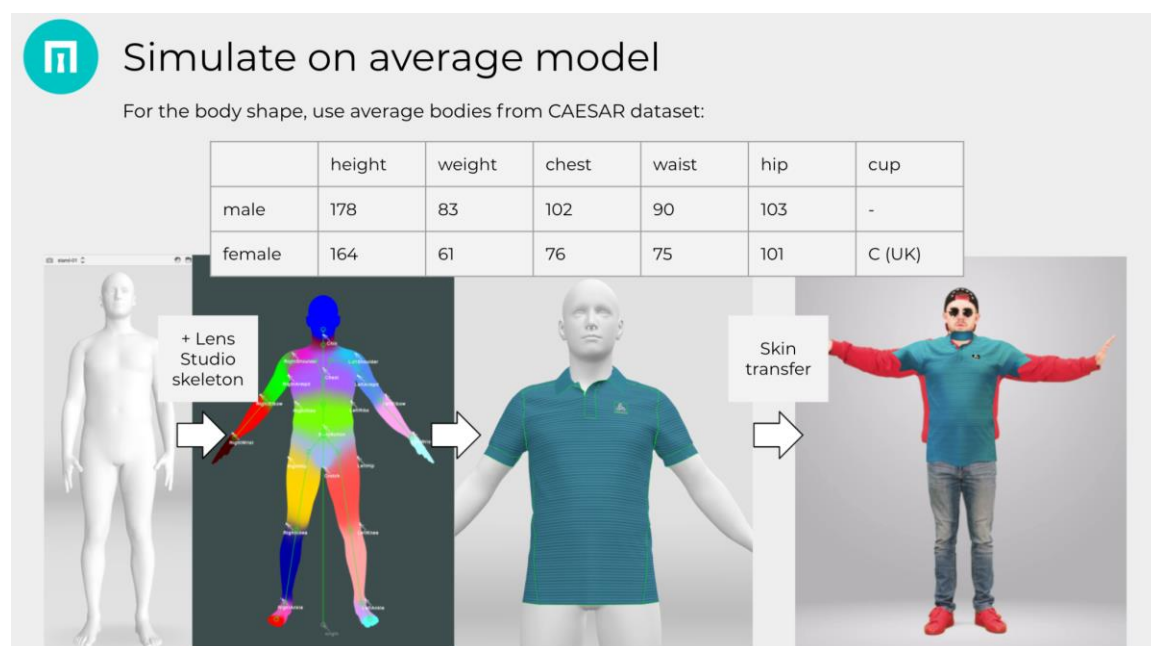


Figure 42. Simulating the garment on an ‘average’ male body shape

There are some remaining questions about the quality of the mesh. For starters, we have removed buttons because they are very high-poly, way over the recommended values from Lens Studio. We’ve also changed the simulation grid in VStitcher to create a more coarse mesh. We also need to set up some occluders in the app, so when the arm crosses the chest, the arm occludes the garment. At the moment, we have only included a neck occluder to hide the back of the neck.

7.5.3 Demo of end result

This video below shows the Snapchat AR filter in its current state. You can see an Odlo T-Shirt rendered in 3D which follows the dancer’s body as he moves:



[Video of Odlo garment virtual try on SnapChat lens filter](#)

7.6 Pipeline for the magic mirror

The final magic mirror is currently planned to be built in Unity but targeting only iOS because body tracking is only supported through Apple's ARKit at the moment. Most of the work described above will still be relevant, but there will be some requirements for our pipeline:

- Skeletal differences in ARKit which will need to be accounted for
 - In Lens Studio we had to remove, re-parent and rename bones as described above. Our initial investigations for ARKit show that we may also need to add extra bones. This is not trivial because it will mean new skinning weights may need to be painted manually. Researching this further will be an important next step in the project
- Automation of the steps above where possible. The avatar creation has been already automated and it currently runs in the cloud using AWS Batch. The garment skinning CLI is run manually, but it can be automated. However, it is not currently a requirement for the Magic Mirror app since the skinning doesn't happen during runtime: garments are prepared beforehand.
- Scripting the garment preparation steps where possible, to facilitate the work of designers.
- Garment simulated on several reference shapes and selection of asset on the basis of the closest shape. This depends on the outcome of the advances in Work Package 1.
- Prediction of ObiCloth parameters and weights, so the garment simulation in Unity gets closer to the Physics-Based Simulation reference generated by VStitcher.
- Transfer garment from reference shape into target body shape. The actual avatar changes from user to user, but garments are simulated only in a few reference body shapes. If the garments aren't transformed, when the avatar is dressed it may look as if a different person is wearing them.

8. Conclusion

The objective of this deliverable was to describe the core functionality of the avatar-garment simulation for the eTryOn Project use cases and explain the 3D garment collection process. While creating the 3D garment collection process in an established 3D software like VStitcher is very streamlined and clear of the possibilities and workarounds it can be well explained and has no new innovative technology behind. The second part of this document describes the avatar-garment simulation with new developed technologies and is still under development. But we can say that we have successfully created the first set of garments and have better understood some of the issues when transforming true to life digital garments into digital assets, which can be used in different systems like Unity or Snapchat. In this document we have outlined our current approaches to managing the simulation between virtual garments and avatars. The demonstration video show how we have created an initial prototype combining the output of WP1 with the created digital garments from Odlo in VStitcher which can be animated in Snapchat.