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: D1.1

Core functionalities of the mobile application and software for the self-scanning application

Due date of deliverable:	31 May 2021
Actual submission date:	4 June 2021
Version:	1.4
Main Authors:	Thomas De Wilde (QC), Axl François (QC), Yannick Francken (QC)

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 number	1.1
Deliverable title	Core functionalities of the mobile application and software for the self-scanning application
Deliverable version	1.4 (Final)
Previous version(s)	1.0, 1.1, 1.2, 1.3
Contractual date of delivery	31 May 2021
Actual date of delivery	4 June 2021
Deliverable filename	eTryOn_D1.1_final.docx
Nature of deliverable	Demonstrator
Dissemination level	PU
Number of pages	45
Work package	WP1
Task(s)	T1.1, T1.2, T1.3, T1.4
Partner responsible	QC
Author(s)	Thomas De Wilde (QC), Axl François (QC), Yannick Francken (QC), Alexander Vandevelde (QC)
Editor	Elisavet Chatzilari (CERTH)
Reviewer(s)	Jim Downing (Metail), David Gavilan (Metail)
Abstract	This deliverable presents the work done to increase the attractiveness and accuracy of the 3D body models. This is achieved by implementing a new body space, evaluating different matching approaches and developing an improved segmentation algorithm to extract body silhouettes. In addition, a texturing approach has been presented. The base SDK modules are laid out, explaining how QuantaCorp integrates with the eTryOn applications.
Keywords	3D modelling, human body models, body matching, segmentation, SDK, projective texturing

Copyright

Deliverable history

Version	Date	Reason	Revised by
1.0	06/04/2021	Table of Contents	Thomas De Wilde (QC)
1.1	12/05/2021	Initial Version	Thomas De Wilde (QC)
1.2	25/05/2021	Reviewer's Version	Jim Downing (Metail)
1.3	28/05/2021	Final Version with reviewer comments incorporated	Thomas De Wilde (QC)
1.4	02/06/2021	Final version with editorial changes	Elisavet Chatzilari (CERTH)

List of Abbreviations and Acronyms

Abbreviation	Meaning
B2B	Business-to-Business
B2C	Business-to-Consumer
BMI	Body Mass Index
CAESAR	Civilian American and European Surface Anthropometry Resource
CNN	Convolutional Neural Network
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
HFI	Human Fashion Interaction
IoU	Intersection over Union
JSON	JavaScript Object Notation
JWT	Json Web Token
PaaS	Platform-as-a-Service
PCA	Principal Components Analysis
PoC	Proof of Concept
SCAPE	Shape Completion and Animation for PEople
SDK	Software Development Kit
SMPL	Skinned Multi-Person Linear
STAR	Sparse Trained Articulated Human Body Regressor
UC	Use Case
UI	User Interface

List of Figures

List of Tables

Table of Contents

1 Executive Summary

The objectives of Work Package 1 of the eTryOn project are the generation of accurate and attractive 3D models and the development of a Software Development Kit (SDK) that facilitates the communication with the QuantaCorp API and the capture of the required data to generate the models. So far, we have investigated the problems with our current algorithms and have focused on improving the visual aspects of the models, specifically on regions near the head, hands, feet and armpits. We evaluated and tested new body spaces to tackle the problems in appearance and we have also improved our segmentation algorithm. The general idea of what the SDK should do and how it will integrate in the eTryOn applications is laid out.

In this deliverable we describe the work that WP1 members have put into making the avatars more accurate as well as more attractive, and what the next steps are in improving accuracy and attractiveness. We also give an overview of what the SDK will look like, and how it will be used within eTryOn's various use cases during the pilot phase.

2 Introduction

The eTryOn project's main objective is the development of technologies that allow the virtual fitting of garments. This would allow people to interact with fashion items in an innovative way at the comfort of their own homes. These interactions of humans with fashion items (Human Fashion Interaction, HFI for short) can be split up in various domains, with at its core the creation of a personalised three-dimensional avatar.

QuantaCorp's current technology stack offers a quick way of creating such 3D avatars using height and two photos (one of the person's front in A-pose, and one of the side in I-pose) as only input. In the past years, QuantaCorp has been focusing mostly on developing a platform for business-to-business (B2B) applications, with the main emphasis put on the delivery of accurate measurements. More on this can be found in chapter 3 where we give an overview of the existing platform.

Given eTryOn's objectives and the fact this project is more customer focused, the attractiveness of the avatar starts to play an important role. During the eTryOn project, QuantaCorp aims to research and develop the technologies that will allow people to create an avatar that looks attractive enough to be used in a multitude of HFI applications, while preserving the level of accuracy we are able to deliver on our B2B platform. Besides improving the avatar's attractiveness, we will also develop an SDK to integrate the communication with the QuantaCorp API in web or mobile applications and facilitate the capture of the required input data.

As mentioned earlier, in chapter 3 we give an overview of the existing platform. The existing platform refers to the state of the platform before the eTryOn project started. We describe what QuantaCorp's core technologies are, and how we allow other businesses to interact with our technology via our API, web portal, and mobile application.

In chapter 4 we will go over the issues with the attractiveness of the avatar and what has been done to address these issues. We go over the improvements done to reduce artefacts around the chest area. We talk about the changes made to export the avatar with the pose as captured in the photo. We look into how we can improve the quality and level of detail of the hands, feet and head. And we discuss an approach to texture the avatar.

Chapter 5 elaborates on the implementation of a new body space which we conclude to be a necessity in chapter 4. After an introduction, we compare QuantaCorp's current body space with the state-of-the-art STAR-model. Next, we discuss alternative approaches for the existing matching algorithm.

Measurement accuracy is the topic of chapter 6. After a brief introduction, we take a look at the effects of segmentation on measurements and we explore the upper limb measurements and how to refine them.

In chapter 7 we take an in depth look at the improvements made to QuantaCorp's silhouette segmentation. We dedicate an entire chapter to this topic because pixel-perfect segmentation results in more accurate measurements, as was concluded chapter 6. First, we introduce the subject of segmentation. Next, we look at the relationship between the number of available pixels and the quality of the segmentation and we look into the delicate problem of edges. And we end with our conclusions on the progress that has been made.

Chapter 8 covers the development of the SDK. We explain why we shifted from a standalone scanning app, to the implementation of an SDK. We explain the communication flow between the eTryOn applications, the SDK, the QuantaCorp Cloud and the eTryOn Cloud. And finally, we describe how the scanning component will work, and what it may look like.

3 Platform Overview

The QuantaCorp sizing platform first came to light early 2016 with the goal to optimise business processes by leveraging body measurements in day-to-day customer interactions. It allows B2B businesses to work more efficiently by providing a cutting-edge body scanning technology that is fast and cost-effective. Today's cloud platform is offered as a completely GDPR compliant platform-as-a-service (PaaS) that aids with the digitisation of body measurements. In the past five years it has known a steady pace of development, growing into a product that provides a better fit to the needs of our customers in workwear, health and retail with every new iteration.

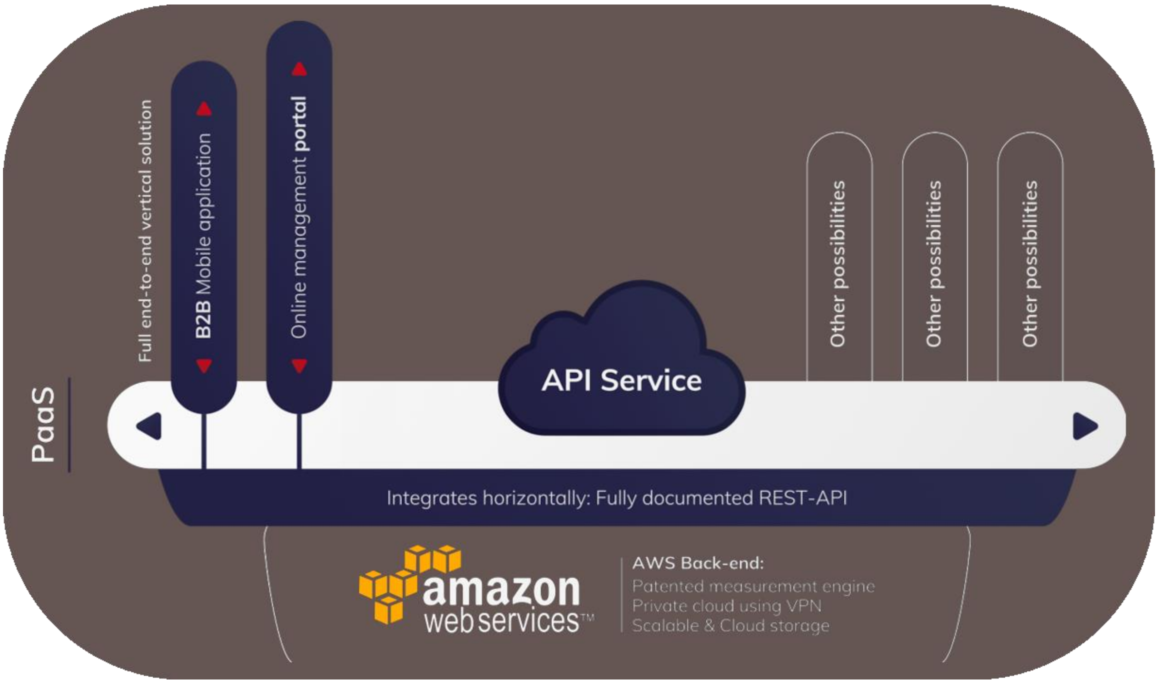


Figure 3-1: Overview of our Platform-as-a-Service.

The core technology consists of multiple microservices that interact with one another to turn two pictures, a front and a side picture, into an avatar of the person by employing a statistical model. The first and arguably the most important component we developed for our cloud platform is the public interface that gives access to our core technology. This fully documented application programming interface (API) uses representational state transfer (REST) to communicate over the world wide web (WWW). It is the base component that allows us to develop all other components that make up our platform, and allows other businesses to develop their own custom platform if desired. The API allows our customers to integrate our solution into third-party systems such as SAP, Microsoft Dynamics and plenty of other management systems.

Though this interface allows businesses to build their own platforms with our technology, most of our customers make use of the one we provide. Usually their first experience with our platform is through our web portal. In this web application, we provide the user interfaces (UI) for a plethora of API features. Customers are given an overview by means of a dashboard. Customers can manage their garments and configure the accompanying sizing tables or bundle them into collections. The portal also allows customers to manage projects and provision projects with *size passports*. In our platform, information about the model, the most common measurements of the 3D model and garment size advice are bundled into what we like to call a size passport. Other portal features include the ability to download a size passport’s 3D model and extracting a data export of the size passports.

In parallel with the web portal, we developed a mobile scanning application that aids our customers during measuring sessions. The app is used to capture the pictures of the model and communicates with the QuantaCorp cloud services using the API. It is developed for the Apple platform and supports both iPhones and iPads. App distribution happens via the App Store. Customers are trained and coached in using the app to get the best results possible. Its easy-to-use UI allows the user to create and edit size passports, take a scan, and consult the scan results and size advice once processed. A huge benefit of the app is that it can be used offline. Once a connection to the internet is established, the app will automatically start uploading scans to the cloud for processing.

Figure 3-2: Web portal.

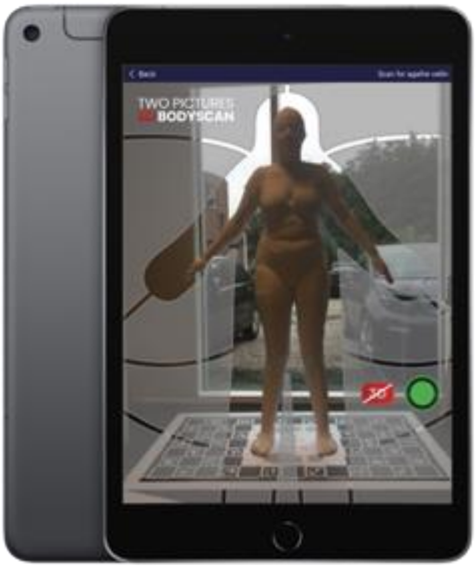


Figure 3-3: Mobile application.

When a scan is uploaded via the API, the first microservice that runs is our segmentation service. This service takes the colour images as input and is able to segment out the body shape, or silhouette, of the model. This segmentation is then outputted as a black-and-white image and is ready to be used by the next microservice, the matcher service.

During matching a proprietary algorithm looks for the best possible shape and pose match based on the black-and-white silhouettes. This results in a 3D model, for which we have a standard set of over one hundred measurements readily available. The final measurements are further optimised through our proprietary algorithm. Scan results are securely stored away and can now be accessed through the API in the app and portal.

All members in the eTryOn consortium have been given access to the QuantaCorp platform since the beginning of the project. The documentation for our platform and API is available at: https://docs.quantacorp.io/.

For the duration of this project, we chose to work with a fork of our existing platform. Any new features, bugs or issues reported by consortium members are addressed on this fork.

4 Attractiveness of the Avatar

4.1 Introduction

Considering that the QuantaCorp solution is targeted towards body measurements, it is expected that its originally produced 3D models are not very attractive. A big issue is the level of detail in the 3D models; this is most prominent in the head, hands and feet and a webbing issue near the armpits. During tests performed within eTryOn (i.e. ODLO performing scans of their fit models and comparing the resulting avatar with the actual fit models) some other issues were brought to our attention that affected the attractiveness and measurements extracted from the avatar. In this chapter we focus on the problems regarding the attractiveness of the avatar, while more on measurement accuracy can be found in chapter 6. The first reported issue is the presence of artefacts near the chest area and confirmation of armpit webbing. A second reported issue is the shape of the legs.

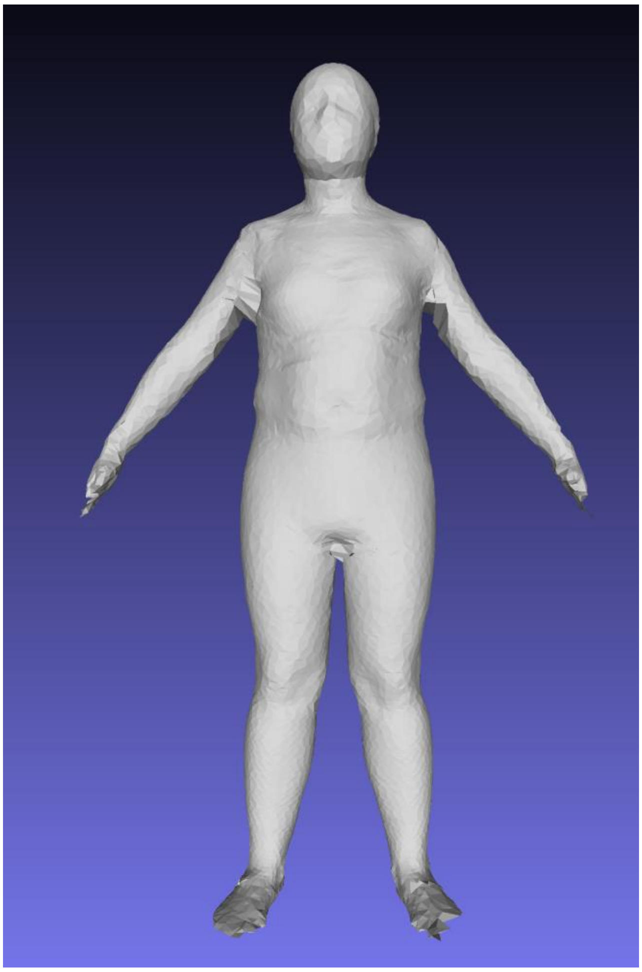


Figure 4-1: Example of a 3D model.

Besides the mentioned defects of the 3D mesh, our avatars are exported without texture. To further increase the avatar's attractiveness, we looked into a way of giving the avatar a livelier look by generating a texture using the original colour images.

4.2 Chest Improvements

In our initial investigation of the artefacts near the chest area, we concluded that they were caused by the degree of freedom that was given to the matcher algorithm. Giving the matcher its freedom allows the algorithm to optimise for measurement accuracy, though this could result in a disparity between the statistical body shape and the actual body.

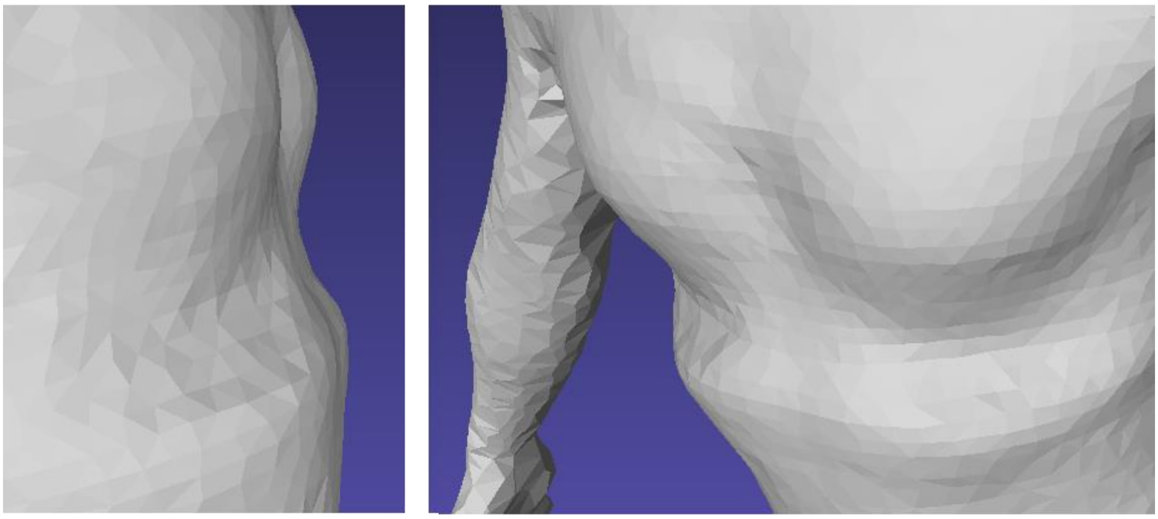


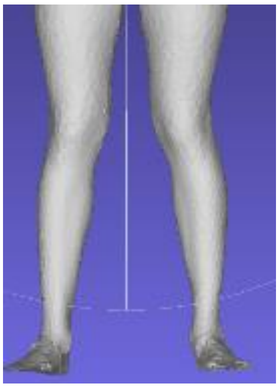
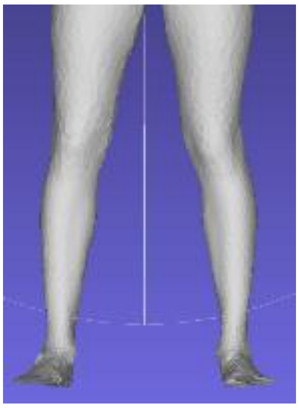
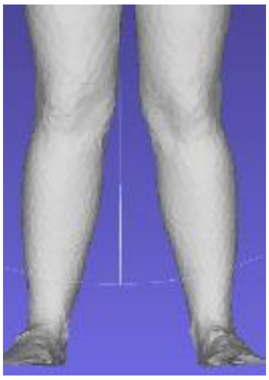
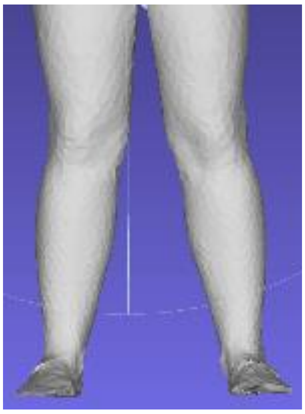
Figure 4-2: The area below the chest clearly shows an artefact.

To reduce artefact creation and deliver 3D models with more attractive and representative body shapes, we branched off a separate matcher service for eTryOn to address this problem. In this branch we put up constraints which keep the matching algorithm in check. Even though this did result in a better-looking shape, we realised this is a less than ideal solution and should not be the accepted solution for this problem. The reason being that we were sacrificing accuracy for attractiveness since bodies that heavily deviate from the average body cannot be expressed by the limited shape parameters. In order to keep the same level of accuracy with an improved avatar attractiveness, we evaluate our current body space to state-of-the-art body spaces With the current body space we mean all possible bodies that can be generated by QuantaCorp’s PCA model.

4.3 Pose Improvements

Besides the artefacts on the chest, the evaluation through the ODLO fit models comparison also indicated that the avatars suffered from what they described as X-legs. It is the bending of the legs in such a way that when you draw a line following the legs, you get two outward curves, effectively forming an X when the knees touch.

Table 4-1: Exported pose vs. matched pose.

Exported Pose	Matched Pose
	
	

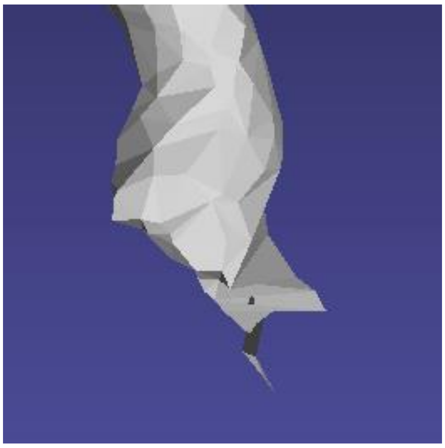
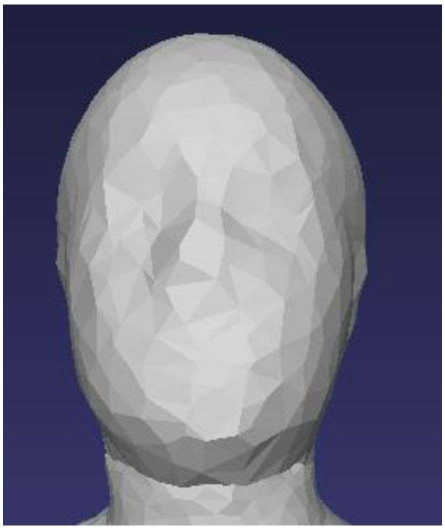
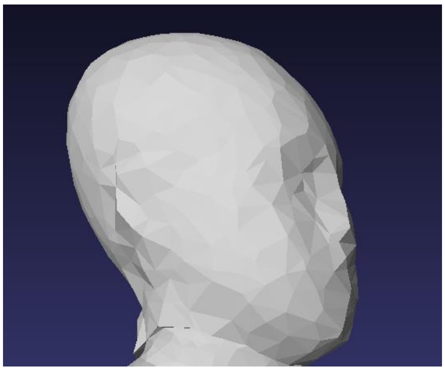
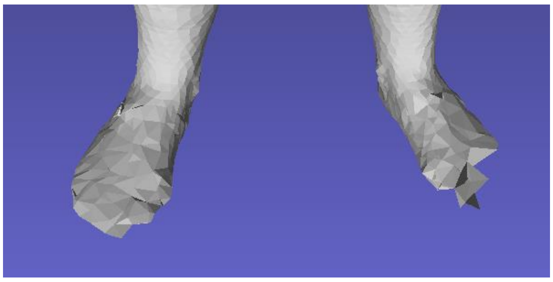
Upon further investigation the issue seems an accumulation of errors. The first problem being the exported pose in the QuantaCorp system is a preset neutral pose. This neutral pose introduces X-legs. The second problem is related to the matching around the knee area. Sometimes the matching in this area is less restrictive than for instance chest matching. The matching algorithm allows for the reprojection to go outside of the segmentation, if that provides a better match in the upper body. The third issue is related to the camera perspective that is used during the image acquisition process.

We added a feature to our dedicated eTryOn matcher that allowed us to export the model in its matched pose. When looking at the resulting avatar, we could still see the presence of the leg artefact. As a conclusion we would say that this issue is partially caused by matching. This gave us a second reason to investigate alternative body spaces.

4.4 Hands, Feet and Head Improvements

QuantaCorp's previous body space has been created with the extraction of specific measurements in mind, rather than visual attractiveness. Certainly, the quality of hands, feet and head reconstruction were ignored since no measurements are extracted in these areas.

Table 4-2: Level of detail near head, hands and feet.

Problem Area	Comments
	The hands are missing fingers and are reduced to a chunked flat disk.
 	Facial features are almost completely gone. We see some relief where the nose is supposed to be.
	Model shows incomplete feet. Chunks are missing in the front or in the side.

Within the context of this project the reconstruction quality of hands, feet and head does not suffice because the model is used for visualization purposes. We considered two different approaches to improve on this: (a) directly repairing meshes and (b) exploring alternative body models that better model hands, feet and head.

Automatically repairing meshes in an efficient way is a challenging subject. When modifying a mesh, it is important to avoid the introduction of self-intersecting primitives, non-manifold issues and other requirements regarding the mesh topology. This is achievable by editing the point cloud rather than the mesh and later fully reconstructing the mesh [4.1]. However, the actual point cloud/mesh modifications as well as surface reconstruction require a computationally intensive processing. Additionally, it results in a new mesh topology so also the UV-layout of the texture becomes invalid. This requires an automated way to generate a new UV-layout. In an attempt to overcome the listed problems, we started exploring alternative body models.

Instead of applying fixes directly on the mesh, we have decided to switch to a new body space. By directly matching bodies that do not exhibit these artefacts, the visual quality is higher and the processing time is considerably lower than applying the heavy mesh operations. More details on the specific artefacts and the solution by switching body spaces are described in chapter 5.

4.5 Texturing

QuantaCorp's processing original pipeline produces scans without any colour information included. Via an internal tool we do have the possibility to visualize a textured representation by projecting the front and side image onto the model.

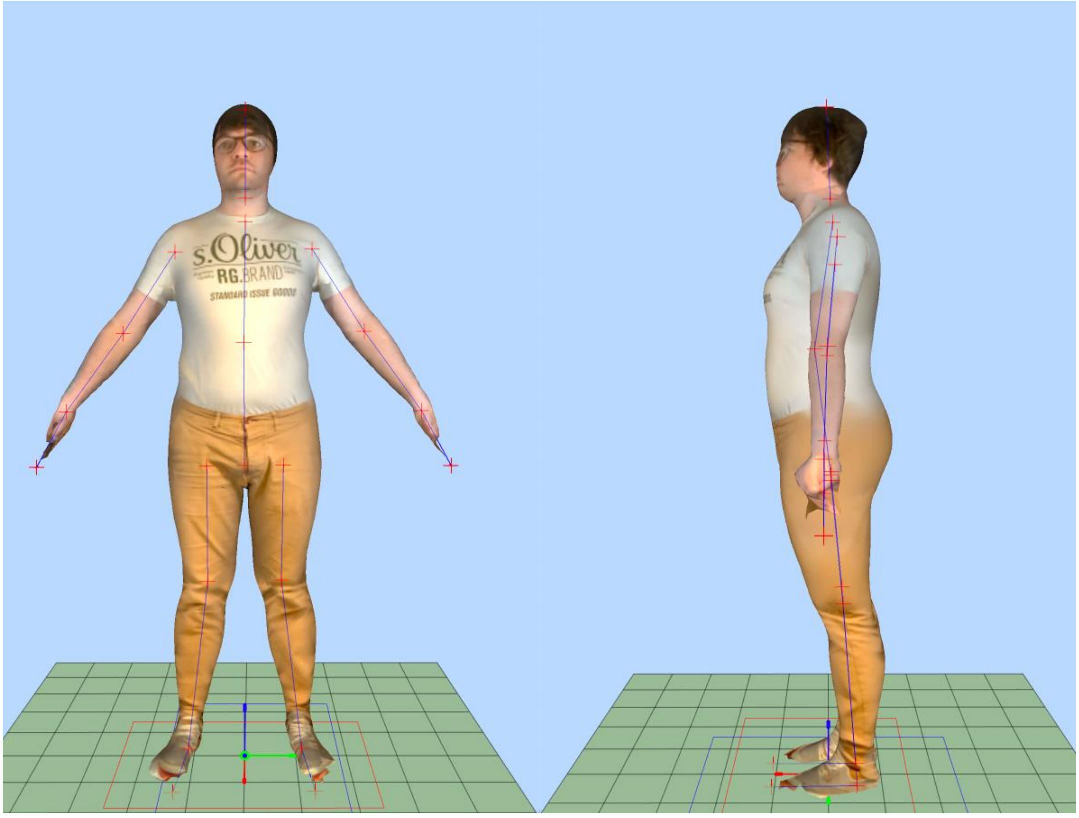


Figure 4-3: Projected textures of a scan taken under lab conditions.

However, there are two challenges. First, there is no method available to export the represented model into a texture map. Second, the reprojected colours coming from the front and side image might not match due to pose variations and deviations in camera calibration. These challenges together with the proposed solution, will be presented in this section.

4.5.1 Exporting Projectively Texture Mapped Models

In the current processing pipeline, no colour information is exported. For development and evaluation purposes, an internal tool is available that visualizes a coloured model by projecting the captured images directly onto the model and blending areas of overlap. A projective texture mapping approach is taken to achieve this. This is possible since we have the all the relevant data available:

- digital model of the person
- front and side image
- camera calibration parameters (intrinsics)
- position and orientation with respect to the digital model of the person (extrinsics)

Directly exporting the colours coming from the projective texture mapping is not possible. We would need to find a way to (a) decide which colour blend should be assigned to a 3D location and (b) a way to store the results. Storing the results can be achieved by building a UV-layout for the model and storing the colours in a single texture image. Automatically creating a UV-layout can be difficult. Fortunately, in our case this can be achieved manually because this step has to be executed only once for the model we use. Changing the shape and pose parameters of the model does not require an updated UV-layout. Hence, in this paragraph we will focus on how to determine the right colour for each pixel in the texture, also referred to as a texel.

The goal is to fill a texture given a 3D model with UV-layout, pictures taken from different views together with their corresponding extrinsic and intrinsic camera parameters. The proposed steps are as follows:

1. Create a texture where each pixel gets a unique RGB value. Since there are 256 x 256 x 256 = 16 777 216 different values possible without even using the alpha channel, a texture atlas of 4096 x 4096 can be created. See figure 4.4.
2. Render the textured model with the coded texture map to all the views where a picture is taken. Use the camera extrinsics and intrinsics to have correct projections. Disable texture filtering to avoid blending different texels. Each pixel in each view should show a single value of the texture, not a mixture of multiple texels. In case the texture resolution is too high compared to the pictures, the resolution of the rendering should be increased.
3. Render the model again with a purely white material, diffuse shading and a single point light source in the camera. White pixels indicate that the orientation of the triangle is perfectly perpendicular to the viewing directions. Black (foreground) pixels indicate that the orientation is perfectly parallel to the triangle. These values can be used as weights to blend colour values of different views.
4. For each render, a map can be created to map a colour code to a list of pixel coordinates.
5. For each texel in the texture, search for the corresponding pixel coordinates in the different views. Calculate the texel colour by taking the weighted average of the corresponding colour values in the different views. Weights are the intensity values of the diffuse shading. Note that a single texel can have multiple corresponding pixels in a single view. Preferably the average is first calculated per view and then averaged out over the different views. The average weight per view should also be

taken into account when blending multiple views (since we do not want an equal contribution per view).

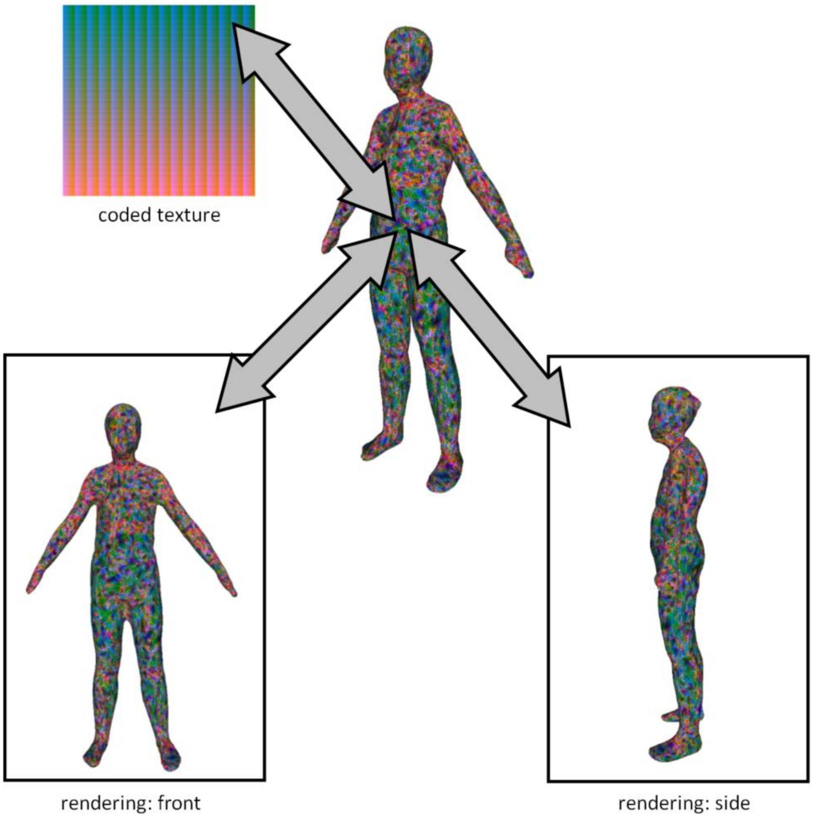


Figure 4-4: Illustration of a digital 3D model, textured with a coded texture and virtually rendered from the front and side. This is used to find the relation between captured pixels and texels in the texture.

Potential challenges are (a) limited colour codes limiting the maximal texture resolution and (b) invisible texels, such as the back of the person, that cannot be filled in. The texture dimensions limit to 4096 x 4096 can be solved by including the alpha channel. By employing the additional channel, the number of codes can be increased by a factor of 256 (factor 16 for width and height separately), yielding a resolution of 65536 x 65536. Alternatively, or in combination with the former method, we can also assign only colour codes to texels that are actually referenced to boost the texture dimensions even more.

The proposed method has been conceptually created but has not effectively been implemented because of the findings described in the next section and the alternative solution agreed with the project partners.

4.5.2 Projection Artefacts

Even with the previous techniques allowing the projectively texture mapped results to be exported to a commonly used file format, such as OBJ and a PNG containing the texture, there are more fundamental issues. More in particular:

1. Material properties that exhibit view-dependent reflections does not allow for blending colours.

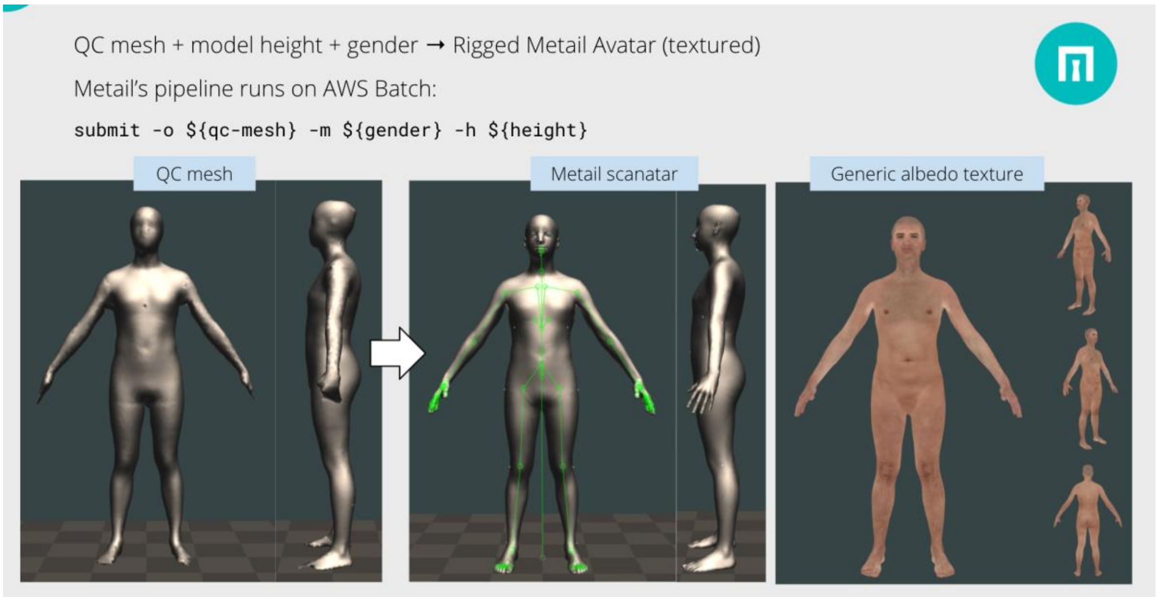
2. The dynamic nature of the person between views can cause misalignments. For instance, pose changes by muscle fatigue, breathing and facial expressions can change between different image captures.
3. Hair is very hard to model correctly. We capture a mixture of foreground and background colour, so a person with black hair and white background will get white and grey spots in the hair.
4. Illumination changes due to uncontrolled light sources can throw off the results very easily. Uncontrolled light sources must be interpreted broadly as it covers flickering of light sources, clouds moving in front of the sun, changing shadows and many more. This can only be partly prevented without using a professional photo studio.

Based on our findings and expertise, we highly doubt if generating photorealistic texture maps can be done with the current setup. A specialised studio would be required to achieve the aspired photorealism, which is outside the scope of the project. The solution to this is described in the next paragraph.

4.5.3 Solution

The main reason for rejecting this proposed texturing method is the lack of detail in facial features like hair, eyes, nose and lips. On the other hand, a better solution may be found in the use of a hand-made texture. This artistic interpretation will keep it accessible for the broad public and will not require a specialised studio filled with expensive hardware. It will be the closest we can get to generating photorealistic texture for the avatars. The solution of using artistic renders will be developed and put to the test in a later stage of the project.

Apart from being textured, the final avatar requires to be rigged so it can be posed and used for dressing. The obtained mesh is fed to the rigging service developed by Metail, which registers the mesh against a template model. This registration, as shown in figure 4.5, adds some visual features to the mesh, like detailed hands and feet, and a generic texture. The skeleton of the rig matches the body shape of the source mesh.



QC mesh + model height + gender → Rigged Metail Avatar (textured)

Metail's pipeline runs on AWS Batch:

```
submit -o ${qc-mesh} -m ${gender} -h ${height}
```

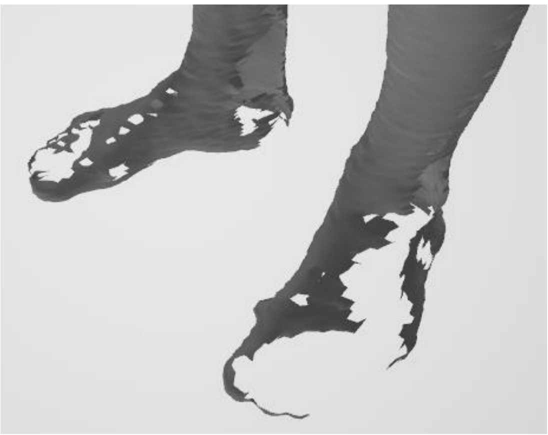
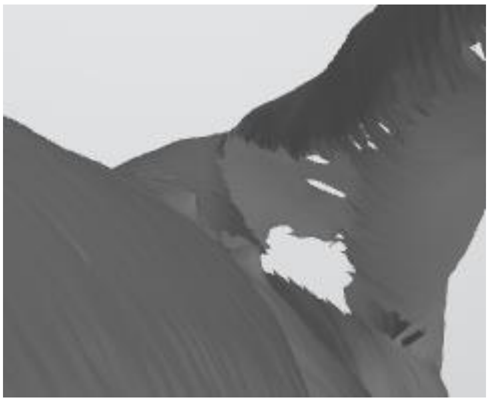
Figure 4-5: The QuantaCorp mesh gets processed by Metail's pipeline resulting in a rigged avatar.

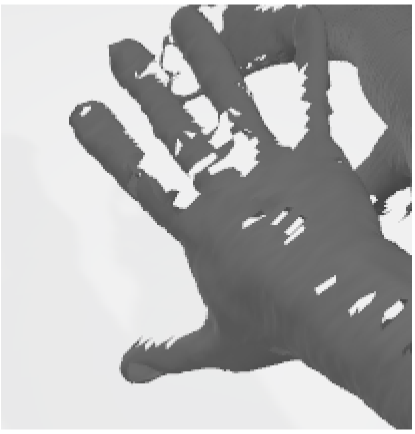
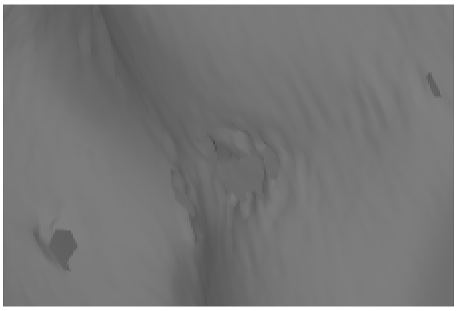
5 Integration of New Body Space

5.1 Introduction

In chapter 4 the attractiveness of the avatar was discussed. Chapter 5 will give a detailed explanation on how to solve these problems. In order to improve the current QuantaCorp body space (all possible bodies generated by the current QuantaCorp PCA model) the input data would have to be examined. Once all problems with the input data are fixed, a new body space could be created. The current QuantaCorp body space is generated by using scans from the CAESAR database [5.11]. Some of the more obvious problems are detailed in the following table 5.1.

Table 5-1: This table shows the shortcomings in our CAESAR scans.

Problem area	**Comments**
	Feet reconstruction
	Hole filling is needed in the armpits

	Hand reconstruction
	Chest holes

Observing that all the input scans show these types of problems, we would have to fix the complete CAESAR set in order for us to create a body space where these problems are less prominent. Note that in literature there are several ways to do this like *A survey on algorithms of hole filling in 3D surface reconstruction. Visual Computer* [5.5], *A robust hole-filling algorithm for triangular mesh* [5.2] and *Template-Based Mesh Completion* [5.1].

Considering that fixing all our input data would take a long time, and even if we did it in an unsupervised way we would still have to check every avatar, we decided to first investigate alternative models. In recent years *SMPL: A skinned multi-person linear model* [5.3] has been gathering a lot of traction. A new iteration on the SMPL model was released in 2020, namely *STAR: Sparse Trained Articulated Human Body Regressor* [5.8]**.** After doing initial experiments with the STAR-model, several observations were made. First of all, the resulting avatars did not have any unwanted holes or artefacts. Second, the attractiveness of the avatars was also increased.

In section 5.2 a further comparison between the STAR-model and our current body space will be made, while answering important questions; can we integrate STAR without any real hiccups? What are the implementation problems we encountered? Does STAR offer the same accuracy in measurements than our current body space?

Section 5.3 will detail several alternatives for our current matching approach. In this section we will compare results from several matching methods, taking into account the state of the art.

5.2 Body model

Given the limitations that were described above with the current QuantaCorp body model, we investigated the state of the art for new body models. The two contenders that we found were SMPL and STAR. We choose to continue with the STAR model for the reasons outlined in the STAR-paper. The paper claims there are three main advantages of STAR versus SMPL:

First, SMPL has a huge number of parameters resulting from its use of global blend shapes. These dense pose-corrective offsets relate every vertex on the mesh to all the joints in the kinematic tree, capturing spurious long-range correlations. ~Abstract, STAR

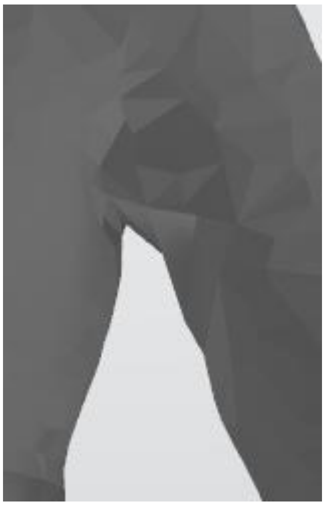
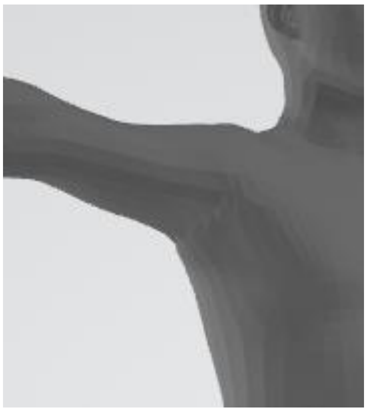
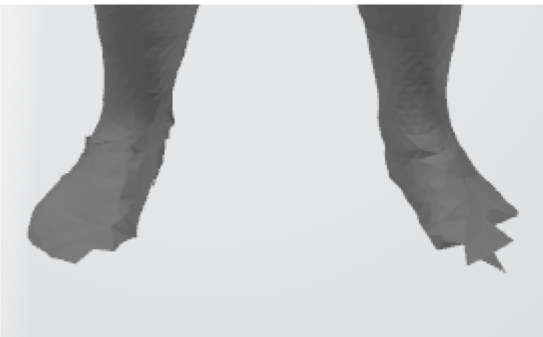
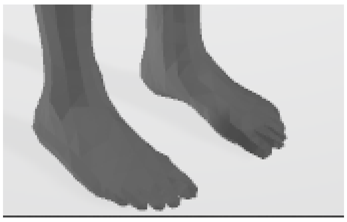
Second, SMPL factors pose-dependent deformations from body shape while, in reality, people with different shapes deform differently ~Abstract, STAR

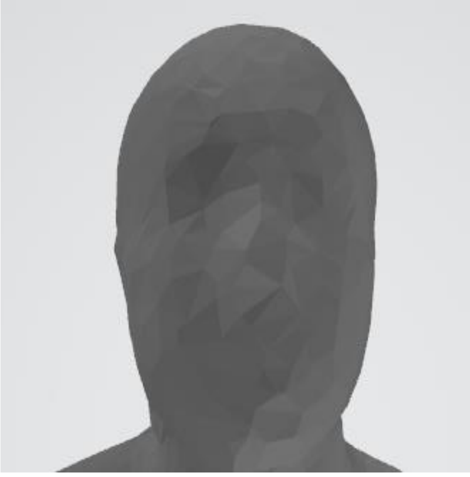
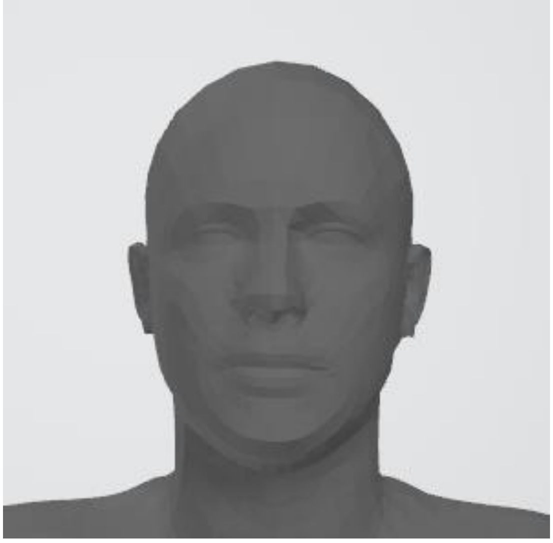
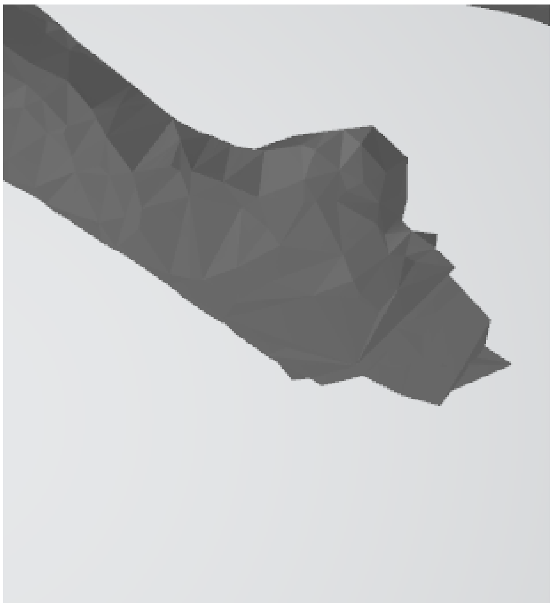
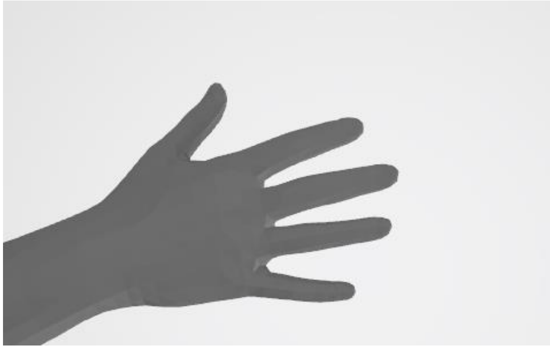
Third, the shape space of SMPL is not rich enough to capture the variation in the human population. We address this by training STAR with an additional 10,000 scans of male and female subjects ~Abstract, STAR

Given that our current input data for our body space is the CAESAR set, the resulting avatars will have the same problems as our input data. In any data driven method the resulting method will only be as good as the data it is trained on.

From the paper we conclude that from a theoretical perspective we expect the STAR-model to outperform our current body space. Both in attractiveness and measurements. In the next table you can see the relative improvements of the STAR avatars.

Table 5-2: This table visually compares the current QuantaCorp-body space with the STAR-model.

QuantaCorp's Current body space	STAR-model
	
	

Next comes up the question: How do the body measurements compare versus our current body space? In order to give a good answer on this question, several steps have been undertaken. First, we had to integrate the new body space in our existing code base. In section 5.3.2, we will take a closer look at this integration.

In our results comparing the latest version of the original body space with the first implementation of the STAR-model, we found that the STAR-model has a positive impact on the results when comparing all torso measurements. The comparison happened on an internal validation set for which we have ground truth 3D scans. The validation set was evenly distributed between men and women. These measurements start around the trouser waist and go up towards the chest area.

5.3 Silhouette-based matching

5.3.1 Introduction

The digitalization of human bodies is a difficult task. As a first option, there is the possibility to 3D-scan the human body. This scan would make use of one or more sensors that are capable of capturing depth information. In particular structured-light stereo techniques come to mind and more recent LiDAR technology. The captured images then have to be registered in a global space and the human body can be reconstructed. However, these setups have a couple of downsides. Multiple high-quality sensors are expensive, moreover they have to be set up, calibrated and the data has to be processed.

As a second option, body matching tries to infer human bodies from RGB pictures. The main idea here would be to start from a predefined parametric body model, like STAR, SMPL or SCAPE [5.10]. In a next step we would have to define the amount of available input data. For the method we currently use this is two pictures and the height of the person. From those pictures we derive silhouettes and it is those that we will try to fit to the parametric body model. From a more general point of view, this is called the prior-information. As part of the scope of eTryOn, we have also looked at expanding the prior information we currently use. For instance, BMI, 2D joint points, extra views are all candidates to improve our current methodology.

The main question here is how do we fit our prior information to these parametric body models. There are three main approaches that could be followed for this, and we will look at two of those more closely in this chapter.

The first one is an optimisation-based method. In this case, traditional optimization techniques are used and a certain energy function is either minimized or maximized. A few techniques that are very well described in the literature are iterative methods (Newton’s method, gradient descent, line searches, trust regions, etc), while alternatively there are also heuristic-based methods (simulated annealing, tabu search, genetic algorithms, swarm optimization, etc). These techniques can be a differentiator, since they all have upsides and downsides. How the prior information is used in the objective function and thus how the objective function is defined, is very important for these methods.

The second option to infer human bodies from prior information is by using CNN’s. Recently, the state of the art has been expanded with a number of very interesting papers which all use some sort of prior information combined with different loss functions to build a neural network that can infer both the shape and pose parameters of the parametric models. Although many methods exist, here we take a closer look at *HS-Nets: Estimating human body shape from silhouettes with convolutional neural networks* [5.4] and *Towards Accurate 3D Human Body Reconstruction from Silhouettes* [5.6]. Both methods define a network which is then trained with artificial data.

A third option would be to not use a parametric body model at all. In this case, there are also multiple options possible, but mostly the prior information is used to derive new data. As an example, a CNN might be trained to infer depth information from RGB’s. In *Moulding humans: Non-parametric 3D human shape estimation from single images* [5.7] point clouds describing human bodies are estimated from RGB images. These methods are typically limited by the resolution of the output grid and cannot produce detailed estimates, and as a result, will not be further explored in this document.

In the scope of the eTryOn project multiple papers and methods were evaluated. The goal was to improve our current methodology and obtain better results. Better results are defined as human bodies who have realistic shapes and for which the measurement error falls under or equal to 1 cm. In the next subsections, those efforts will be explained.

5.3.2 Generating and Comparing Silhouettes

For silhouette-based matching, a scoring function is needed to compare the expected silhouette with respect to the silhouettes of the 3D model. Therefore, given the 3D mesh of the model, virtual silhouettes need to be generated. We have achieved this by building a rendering tool that employs the GPU to increase the processing speed.

The silhouette rendering tool that has been developed creates binary masks containing black and white pixels, indicating respectively background and foreground pixels. The input parameters are the 3D model, camera position and orientation with respect to the model (extrinsics) and camera and lens properties (intrinsics).

Besides the required parameters listed above, we have also added an optional vertex selection mask. This has been integrated to indicate which vertices of the model should and should not be visualized. The selection is performed directly on the GPU. We have included this mask to exclude upper limbs of the model when producing virtual side pose silhouettes. Since the arms of the person should be close to the body when scanning the side pose, they should not be visible in the side pose silhouette. However, during the matching process it is possible that deviations in arm poses would occlude the projections of the body. We have solved this by excluding these areas in the 3D mesh during rendering.

When the virtual silhouettes are created, they are compared to the silhouettes extracted from the captured images. In the current implementations we use Intersection over Union (IoU). IoU is calculated as follows:

- Intersection of the two masks by applying the AND operator
- Union of the two masks by applying the OR operator
- Count the number of white pixels in the intersection mask and divide by the number of white pixels in the union mask

In addition, we have built an adapted version of IoU that punishes overshoot. This punishment is used as an incentive for the loss function to stay within bounds of the segmentation masks.

5.3.3 Genetic Algorithms

In this subsection we start from a genetic algorithm as an optimization technique. A genetic algorithm is a metaheuristic which is part of the bigger group of evolutionary algorithms. The base idea being to mimic natural selection in software. There are a couple of base operators which have many different implementations but all have the same concept. Examples of operators are selection, crossover and mutation.

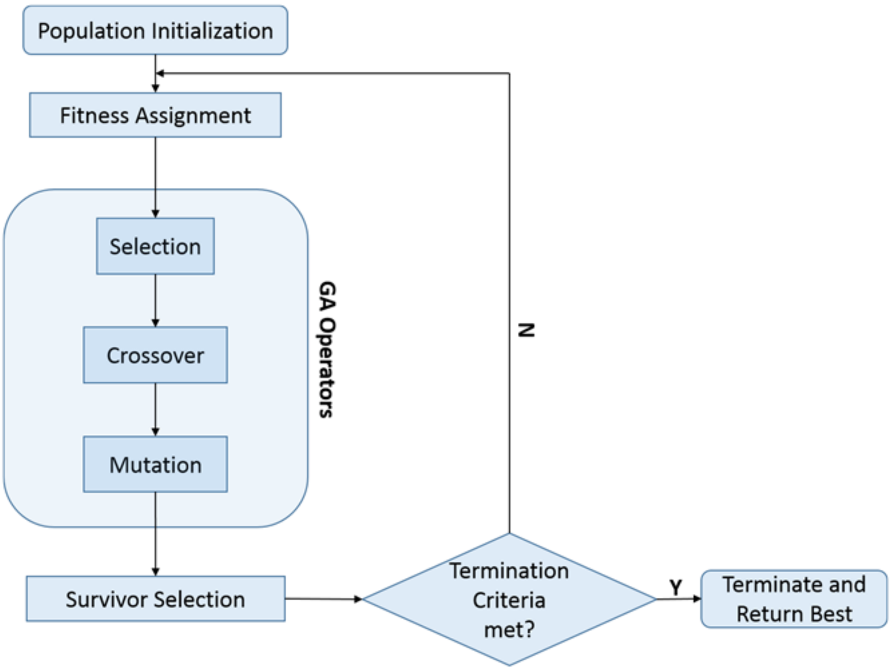


Figure 5-1: This diagram shows the base building blocks for a GA-algorithm.

Applied to our problem we start by defining a couple of variables.

- Number of generations
- Number of shape and pose variables
- Number of solutions (or bodies) per generation

First, we start by generating our first population, random bodies are made which are evenly distributed over the search space. Each body has been given a random shape which is evenly distributed between [-3, 3].

The pose solutions are further restricted by the pose bounds. This is a JSON file where per STAR-model joint the allowed angles are defined by using a lower and upper bound. For each pose coefficient we then generate an even distribution. The combination of these two variables gives us our initial generation. The population of this generation can be anywhere from 128 - 5012 bodies. The number of solutions per generation is only limited by the amount of memory the GPU can feasibly handle.

Per generation, a fitness assignment has to be made. This is the equivalent of an objective/loss function that results in a number being assigned to each solution in the generation. In the current implementation there are two loss functions implemented, the IoU and IoU with extra punishment for overshoot as described before.

First, we need to select the “parents” of this generation. In this case, we select the best scoring solutions in the whole generation. Next, crossover will use these parents to make offspring. This offspring represents a merger between two parents. There are multiple techniques of crossover, currently we use fixed point crossover with the fixed point being the middle of the array of shape or pose coefficients.

The offspring that was made in the last step with the crossover operator, will also get mutated to introduce new solutions to the population. Currently the mutation is dependent on the index of the current generation. If we are at the start of the run, multiple values will be selected for mutation, while at the end of the run this will be reduced to one value. The magnitude of change also differs at the start versus the end of the run. Both shape and pose coefficients are mutated.

After mutation we use the offspring and parents to create a new generation and we continue with that generation in the next step. Then we go back to fitness calculation with the newly created generation.

In the first version of this algorithm, IoU scores of 94% were achieved. This is rather *good* since we can never expect a 100% match, due to real life attributes like clothing. In the following image you can see the match results indicated in light blue while the ground truth for this scan is indicated in white.

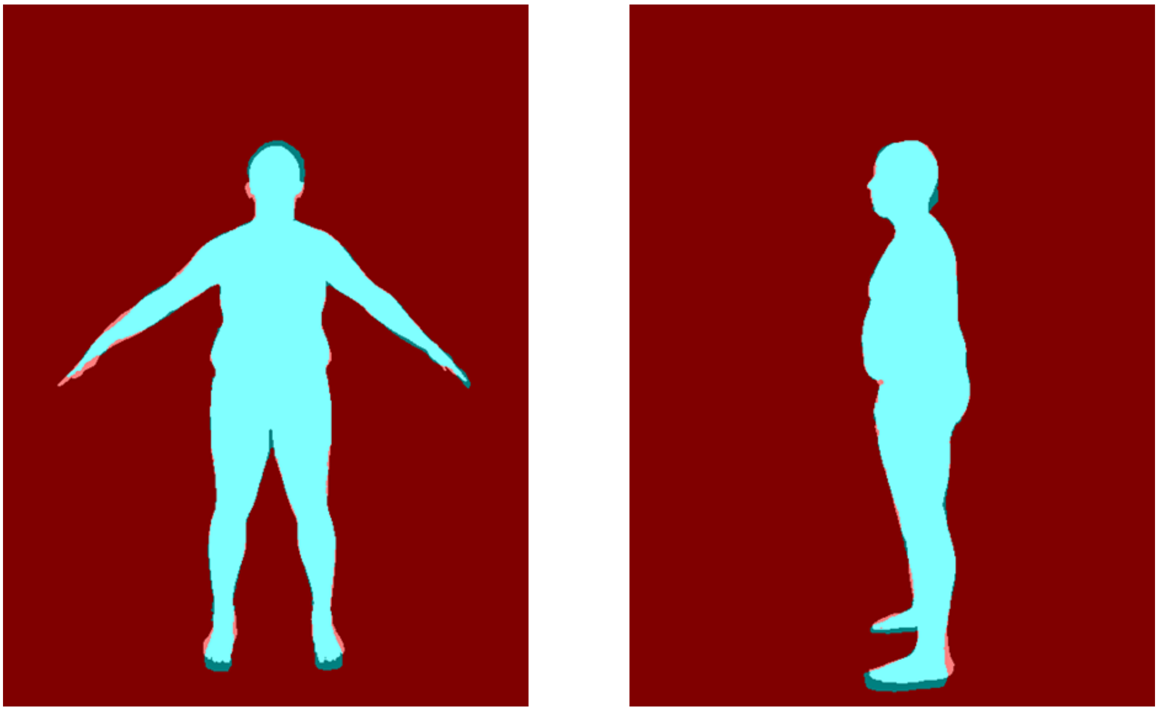


Figure 5-2: This image shows a matched result achieved by the first implementation of our GA algorithm.

The measurements for this scan are validated against the same person being scanned in a 3D-booth. The silhouettes for matching are reprojections of this 3D scan, so as to avoid any sort of pose variation or segmentation issues.

The method was further refined and several avenues of future work were documented and will be further explored.

5.3.4 Machine learning

As mentioned in the introduction to this chapter, two papers, which both utilize CNNs for inference of human body shapes were explored. In this subsection we will discuss the achieved results achieved by both papers.

5.3.4.1 HS-Nets

Seminal work in the field to estimate shape and pose parameters from silhouettes are the HS-Nets [5.4], which is presented by Dibra et al. The results claimed in this paper are very promising and thus we decided to further explore this within the eTryOn scope.

The network architecture for the HS-Nets is shown in the following figure (image from [5.4]). The paper explores several ways to input images to the network both scaled and unscaled. One view or two views, the traditional A and I poses. The result from the network is an estimation of the shape parameters of our parametric model.

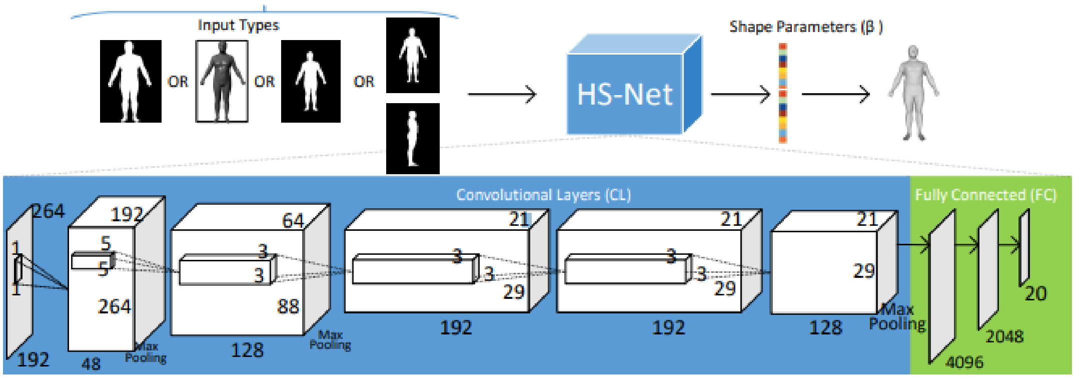


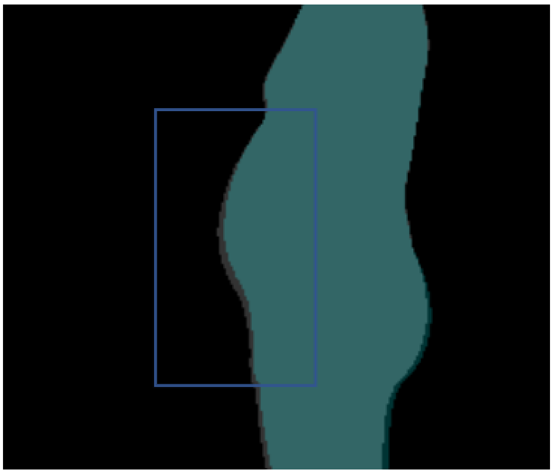
Figure 5-3: Block diagram of the proposed architecture in the Dibra paper.

We want to note that the results in this paper do not use the STAR-model as source of training data. It is our hypothesis that using the STAR-model, which is capable of generating more realistic human shapes, should only further improve the results.

In order to train the model, a lot of data had to be generated. The first iteration of this method used 40 shape parameters, 1 pose (consisting of 72 pose coefficients) and 1 camera pose (both rotation and translation vector). Around 100k male bodies were generated using the STAR-model. This was then trained and our initial observation was that the training was very slow. This is to be expected due to the amount of data used and the nature of the network architecture.

The definition of the loss function we used, can be found in the paper. A loss of 0.5 was achieved in our experiments. The results were validated by using a validation set (which is part of the 100k initially generated bodies, these bodies were not used in training). The inference time is relatively fast <= 1 sec (i7, RTX 2080TI). Although we stopped training early due to time constraints, the results are rather promising. The next images will show some of the results.

Table 5-3: This table shows the results of our implementation of the Dibra paper.

HS-Nets	Comments
	Belly is undermatched
	Shoulder is undermatched
	Pixels are missing around the chest area

Given more training time, it should be feasible to generate a model which does not have these problems. Nevertheless, it still seems to be a long shot to use this approach in a production environment due to varying camera poses. It is not always realistic, especially not in the eTryOn context, to force one camera pose or variations around this camera pose. Future work for this method may include further training, introducing more camera poses, shape parameters, pose parameters and evaluating silhouettes taken from real scans.

5.3.4.2 Towards Accurate 3D Human Body Reconstruction from Silhouettes

The method presented in [5.6] was implemented and evaluated as well. Initial results show that this model also scores well when trying to match inputs generated by the STAR-model. Furthermore, efforts are made to match against our validation sets. The evaluation of this approach is still work in progress and will be further explored in the future.

5.3.5 Conclusion

In section 5.3 there are two main innovations put forward. First a new body space was tested, implemented and evaluated against current QuantaCorp systems. This body space has generated better results than our current body space. Going forward QuantaCorp will use the STAR-model for any eTryOn related activities.

Secondly several novel methods for human pose and shape estimation have been explored. These methods differ significantly from the current methods used at QuantaCorp. At the time of writing, a decision has not yet been made on which method will be used for the eTryOn project.

6 Measurement Accuracy

6.1 Introduction

In chapter 6 we present the issues encountered so far in regards to the accuracy of measurements.

We dedicate section 6.2 entirely to segmentation as it is a very important topic in regards to getting pixel-perfect silhouettes, which translates into increased accuracy.

In section 6.3 we discuss the issues with the upper limb measurements, and we explain why this is a hard problem to solve.

6.2 Segmentation

In this section we want to stress the importance of pixel-perfect segmentation. To do this, we demonstrate the sensitivity to missing pixels in the segmentation when using the QuantaCorp method. There are many variables at play here and before continuing, we want to stress that the findings in this topic are only valid for these variables. Specifically, we will assume certain camera parameters. Obviously for other camera parameters the results will be different.

In this scenario we are going to introduce the following variables:

- FOV: 60.54 degrees
- camera distance to person: 221 cm
- camera height: 110 cm
- image width: 480 pixels
- image height: 640 pixels

These variables may differ per scan, depending on the method, so these variables might be an approximation of the real values. (and thus introduce errors). We further restrict this example by picking a scan which has near perfect matching and then we look at one particular circumference at a certain height on the body. From this we are able to derive a linear pixel per cm rate.

- pix_per_cm = 2.518 [pix]/[cm]
- cm_per_pix = 0.397 [cm]/[pix]

In further research, pixels were added and subtracted to the segmentation to see what the impact is on the measurements. This happened within the scope of the eTryOn project and the broader scope of improving our measurements.

6.3 Upper Limb Measurement Refinement

An issue that aroused by the evaluation of the ODLO fit models scanning is the inaccurate measurement of the upper limbs. The biceps measured on the avatars of every model was off by more than the accepted accuracy tolerance of two centimetres. Correctly measuring the arms is a particularly tricky task. When creating the black-and-white silhouettes, we lose the information necessary to properly match the arms. Though this is a known issue, up until now this was never brought up by our existing clientele. ODLO, being a brand with sports clothing, is more concerned about tight fits. We usually deal with distributors and manufacturers of workwear clothing, which tend to be loose fits. ODLO informed us on the importance of the biceps measurement for the garment's sleeve fit. Although we already briefly looked into these reported issues, further investigation will be needed in future work.

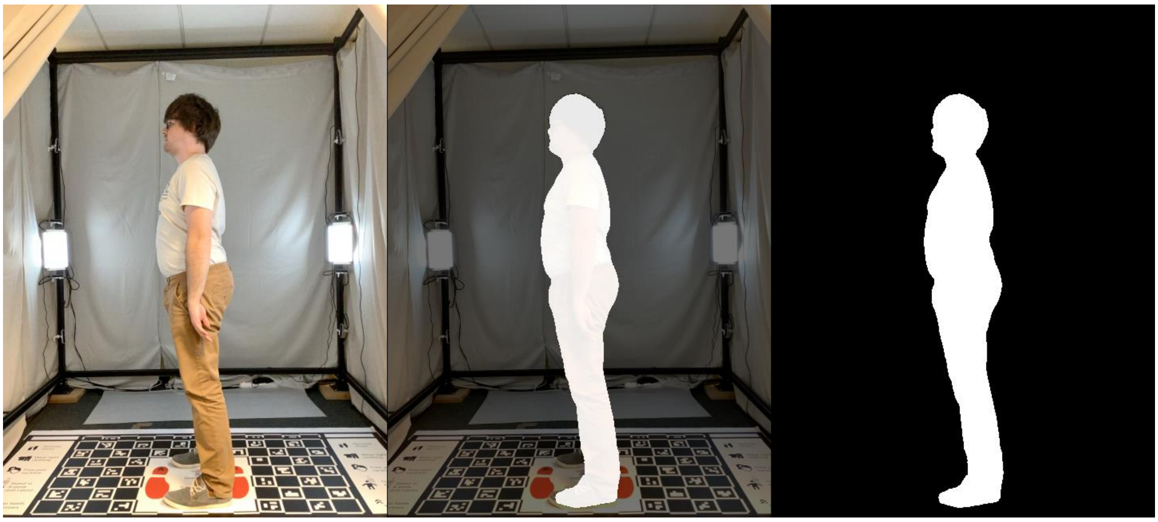


Figure 6-1: In the image on the left, we can clearly see a distinction between the person's arm and torso due to the difference in colour and shading. Moving to the right we can see this information is lost.

One approach could be to expand our segmentation algorithm so it can deliver more information to the matching algorithm. This would entail segmenting only the limbs and would result in the creation of a third black-and-white segmentation image, just for the limbs. We would then need to leverage the information of this new segmentation image during matching. The details of this initial idea still have to be worked out, but the underlying idea is expected to give better measurements for the upper limbs. It is also important to note that this suggestion is very time and labour intensive. To be able to expand our algorithm, we would have to manually segment thousands of pictures.

7 Silhouette Segmentation

7.1 Introduction

In the previous chapter 6 in section 6.2, segmentation was brought forward as one of the main detractors of accuracy with the current QuantaCorp method. The basis of the method assumes that in the input silhouettes only the human body is segmented. To make the problem even more challenging, we will have absolutely no control over the background against which the images are being taken. As a result, the need for accurate, almost pixel-perfect segmentation arises that is very robust against many different backgrounds.

In general, there are several methods that are available for background removal like the more traditional methods including grab cut, watershed or graph-based segmentation. Although these methods might perform well, they sometimes require extra attention based on the exact situation. In the current state of the art, these tasks are all completed by neural networks. Specifically, for binary segmentation, U-Net succeeds in reaching very high accuracy combined with a certain level of robustness.

At QuantaCorp we have been using a U-Net-based method (*U-Net: Convolutional Networks for Biomedical Image Segmentation* [7.1]) for several years. This method has been proven to deliver high accuracy in many cases. That being said, the network is not perfect. Sometimes it is possible to have “artefacts” in these segmentations. These artefacts can be seen as both false positives and false negatives. In order to produce as accurate avatars as possible, it was decided to further improve our segmentation methods.

In any scenario, performance constraints are applied. For us this means that the inference time of our networks has to be low enough so we have plenty of time to do body matching. There is a direct relation between inference time and resolution. The same can be said of the relationship between number of available pixels and quality of the result. Resolution is thus very important and in the next section 7.2 we will tackle this topic. Section 7.3 will deal with the very delicate problem of edges. From a more technical point of view, the following question arises: When is a pixel considered to be background or human? Especially if that pixel is around the edge of both borders. In section 7.4, we will present the achieved results with these proposed methods.

7.2 High Resolution

While resolution is very important in order to increase the quality of the scanned models, the scanning process has to perform under very strict time constraints. Not only that but we also need to be mindful of the bandwidth used by our customers who only have access to mobile data (4G, 5G). For achieving the highest possible accuracy, we would like the highest available resolution. In addition, we also want to know everything there is to know about the camera with which the pictures were taken. These are modelled with the use of extrinsic and intrinsic camera parameters.

In our experience it is impossible to fulfil all the needs lined out in the previous paragraph. Take into account that there are also extra variables over which we have very little to no control. These variables include but are not limited to lighting, reflecting objects, very complex backgrounds, unusual poses and others. We try to solve these problems directly but some of them are solved indirectly.

Currently we have very tight control over the cameras used with our platform and we know all the relevant camera parameters. This means we also know what resolutions are available and against which framerates. These pictures then have to be uploaded to our services. On average this ends up being a ~ 30 mb payload for two 4K pictures.

We had to make several changes to our backend, in order to process multiple incoming requests, especially if these requests all contain 4K-scans. In the next paragraphs these changes are detailed. Given the requirements laid out earlier, we have decided to make our platform more dynamic. Users can and are recommended to upload high resolution images but can still use low resolution pictures if they want. This also introduces the need for a decoupled architecture. For high resolution images we will want to do a refinement step, for low resolution images the service will stay the same.

In chapter 3, it was already made clear that when images are sent to our service a “responder-daemon” does an initial processing of these images. The current segmentation flow is very much in sync with the current responder. This holds us back to scale the inferences made by our models. We need a decoupled architecture so we can scale both horizontally and vertically. In a general software architecture approach this means introducing queues and messages between services.

In this aspect we are converting the segmentation to a micro-service which is then scalable. At the end of the responder call, a message is published with the relevant UUID to a “SegmentorQueue”. A daemon is introduced that constantly polls this queue for new messages. If one appears, inference starts. It is possible to spin up multiple daemons, polling the same queue. The same will happen with whatever our refinement step is after segmentation. We introduce a refinement queue on which messages are placed and polled by a refinement daemon.

7.3 Edge Refinement

In the state of the art there are several novel ways to approach the problem of edge refinement of existing segmentations. Aside from improving upon the existing models by introducing new architectures, existing networks have taken on the challenge of improving existing segmentations. In particular we considered *RefineNet: Multi-path refinement networks for high-resolution semantic segmentatio*n [7.3] and more recently *CascadePSP: Toward class-agnostic and very high-resolution segmentation via global and local refinement* [7.5] and SegFix: Model-Agnostic Boundary Refinement for Segmentation [7.4].

In this section we will focus exclusively on CascadePSPs, its implementation and evaluation within the QuantaCorp environment. The following image shows the potential of the CascadePSP architecture and is taken directly from the paper.

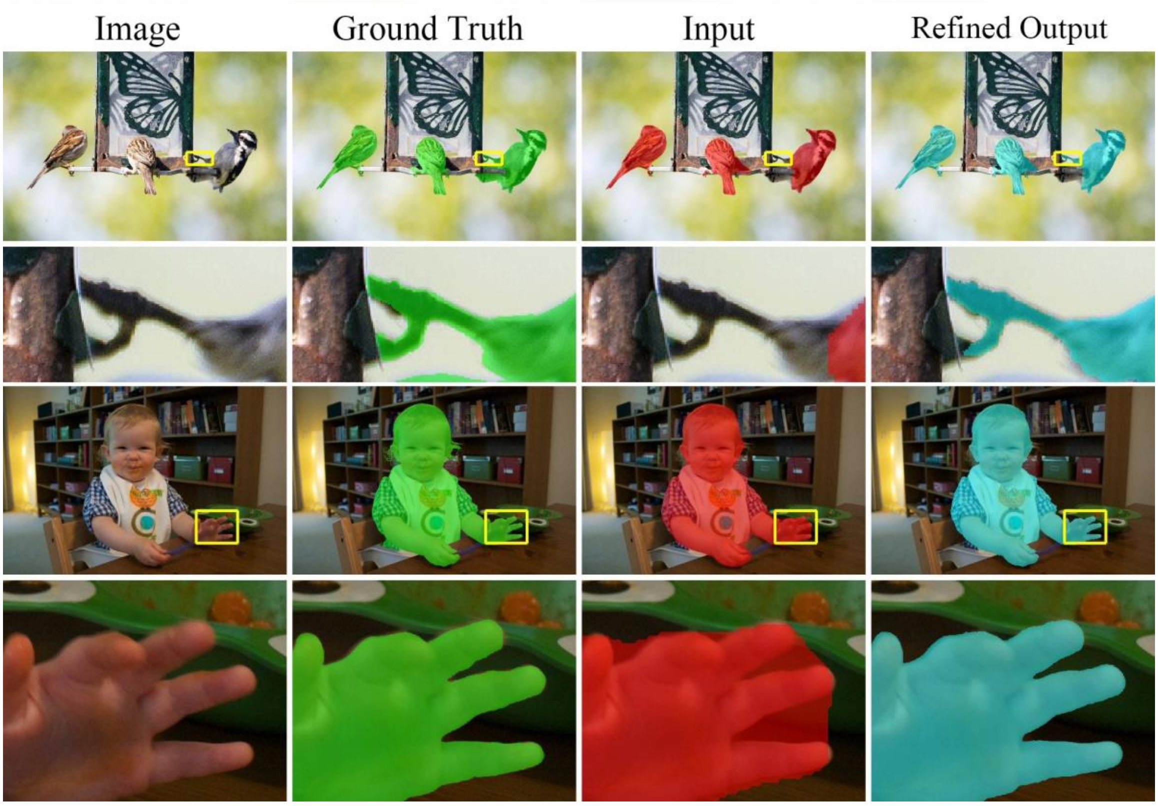


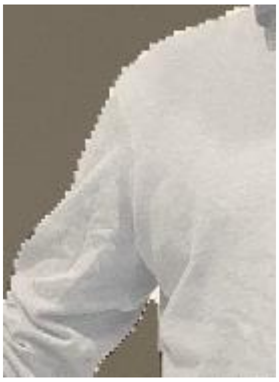
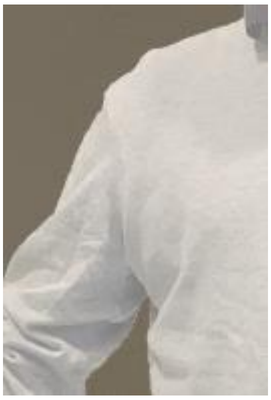
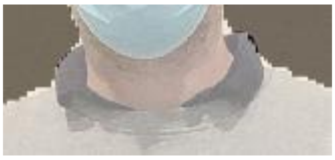
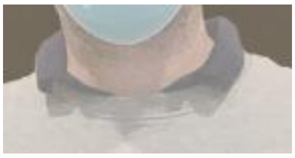
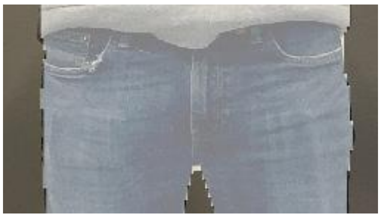
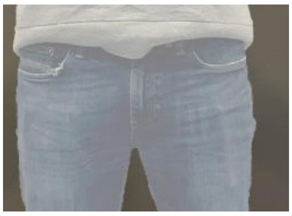
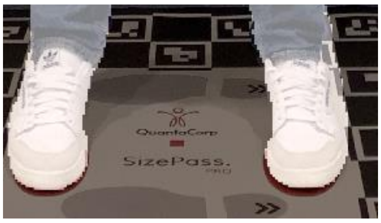
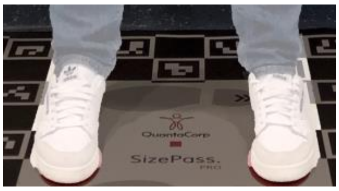
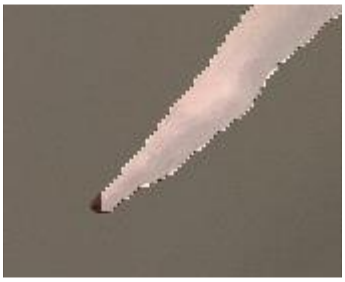
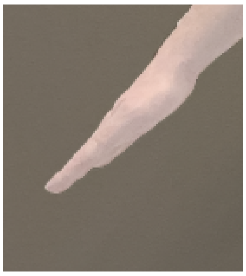
Figure 7-1: This image is provided by the authors of [7.5], showing the improvements.

As you can see the goal of this paper is very clear: improve existing segmentation for which you have high-definition input images. Given the new architecture introduced in section 7.2, we now have the 4K-images for each of our scans. This enables us to apply this extra refinement step after initial segmentation has happened.

The code and pre-trained models for this technique are publicly available and can be found at https://github.com/hkchengrex/CascadePSP. In this section, we will look at the off-the-shelf performance of the PSPCascade. While the performance seems already rather good, if it proves inadequate, we will consider modifications or re-training of these techniques in the future for further improvement.

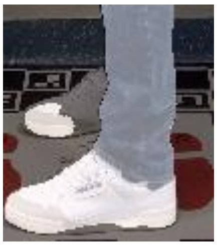
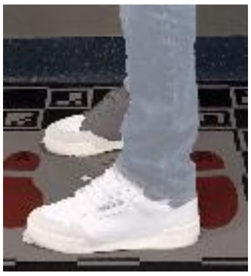
The next table shows the comparative difference between a front scan that has left our current segmentation pipeline and one that is refined with the PSPCascade method.

Table 7-1: This table compares the result before and after refinement, for the front pose.

Original	PSPCascade	Comments
		This is the zoomed-in right-arm of a colleague. Notice how the sharp edges that are clearly visible in the original segmentation are completely gone. The armpit errors are also completely solved.
		
		This one is of particular interest to us, specifically for the measurement crotch_height or inseam.
		Here we notice a problem with the CascadePSP and our mat. This will have to be fixed in post-processing.
		

The next table shows the comparative difference between a side scan that has left our current segmentation pipeline and one that is refined with the PSPCascade method.

Table 7-2: This table compares the result before and after refinement, for the side pose.

Original	PSPCascade	Comments
		Notice how an error in the original segmentation propagates in this case to the refined image as well.

7.3.1 Segmentation Problems

As you can see in the previous two tables, while this method provides a significant performance boost, it does introduce certain problems. We ran this method on several datasets and noticed that we needed to further improve the refinement step. Due to privacy, we cannot share any images taken by customers. The area around the feet seems to be a “problem” area. As a recent addition to our scanning technique we introduced a new mat which already has feet placed upon it. It seems the network confuses those fake placeholders for real feet.

Other problems included false positive pixels, which were segmented fine in the original input segmentation but almost always went outside of the contour of the person. In tacking these problems, we will introduce several post-processing steps in order for this method to be usable in production.

7.3.2 Integration

The CascadePSP network has a very “slow” inference time. Depending on whether only the global or also the local optimization is used. Not only that but uploading two 4K-scans to our services is highly dependent on the available internet connection and bandwidth. In order for us to deploy this model to production we opted for GPU-based inference on AWS.

7.4 Conclusion

In this chapter 7, we started with laying out several problems related to high resolution segmentation. A trade-off between bandwidth, speed and user experience has to be made. An internal proof of concept (PoC) was made as part of the eTryOn project. This PoC significantly changed several of our backend services towards a microservice approach. Moreover, a refinement step was added after our current segmentation. This refinement step greatly increases the edge accuracy of current segmentation.

8 Software Development Kit

8.1 Introduction

In this chapter, we first take a look at how the QuantaCorp scanning module will be integrated in the eTryOn architecture. Next, we take a look at the API clients and finally we go over the scan component.

8.2 Architecture

The description of work package 1 task T1.4 describes the intent to expand on QuantaCorp's existing mobile scanning application for the eTryOn project so as to enable people to take their own scans. One option is to create a standalone application that will handle the scanning. However, in order to allow for a frictionless interaction of the users with the eTryOn applications, we will focus on the development of a software development kit, which can be integrated directly in the eTryOn apps.

The main reason why the SDK approach is the better choice here, is the fact that a single scanning application does not fit in all of the use cases. Depending on the use case, the users may not always be scanning themselves, as explained in the paragraphs below. The SDK will allow the eTryOn application developers to integrate our scanning interface into their own apps, making the need for a separate scanning application obsolete. By removing the separate scanning application, and implementing an SDK, we reduce the complexity of the eTryOn architecture for both the developers and the end-users.

In use case 1 there is a focus on garment designers. It requires the development of a virtual reality (VR) application in which designers can see how clothes fit on their models, the VR Designer app. The VR application will be developed in Unity and will require a VR headset to operate. To populate the VR Designer application with garments and 3D models, a separate back-end application is required. The VR Designer Backoffice application will be a web application developed in ReactJS. In this use case the designer will use the SDK's scanning component integrated into the web application to create 3D models of their size models, not of themselves.

Use case 2 is focused on influencers. A second web application, the Dress Me Up app, will allow users to upload a short video of themselves and in return get that same video with a digital garment simulated on top of the user's body. This web application will also be developed in ReactJS. Before they can upload a video, the influencers will have to digitise themselves using the integrated scan component of the SDK.

For use case 3, a mobile application will be developed for the fashion consumer, i.e. the Magic Mirror app. The mobile application will allow users to virtually dress themselves. Just like in use case 2, users will be digitising themselves with the scan component of the SDK before they can use the eTryOn applications core functionality. The app will be developed in Unity and will initially target the Apple platform, due to the lack of support for 3D body tracking on Android (see deliverable D5.1).

Given all the use cases above, we will be developing an SDK that will contain the scan component and a wrapper for the QuantaCorp API for every platform used by the eTryOn project. For ReactJS this means we will develop in JavaScript or a sibling programming language like ES6 or TypeScript, and deliver the scan interface as a React component. For Unity, SDK development is done in the native programming language of the targeted platform. During the export of the Unity-developed application as a mobile application, the entire project is turned into a project in the native development language of the targeted platform. In use case 3, the output of the Unity export will be an XCode project for the

Apple mobile applications. This means the SDK components for iOS and iPadOs will be developed in the Objective-C programming language and the scan component will be a UIViewController. Should we expand to Android devices in the future, the SDK would need additions in the Java programming language.

In conclusion the SDK will contain per platform two components (a) the API clients, a wrapper for the QuantaCorp API in the native development language of the platform and (b) a scan interface in the platform's most commonly used UI component. The SDK will also include sample code and documentation. At QuantaCorp, we have the most experience in iOS development. The JavaScript components however form a new challenge. Up until now, even though we have a proof of concept using web technologies, this has never been tested in our production environment. We may run into restrictions where we will not be able to extract enough information from the camera (camera intrinsics for example) in order to guarantee the same accuracy as we do in native applications. Should we encounter such issues, it will take some time to introduce the extra features required to extract that information from the camera. Such extra features may include adding a calibration step in the SDK or creating a list of supported devices. In figure 8.1 we give an overview of the SDK and indicate with colour coding the risk involved.

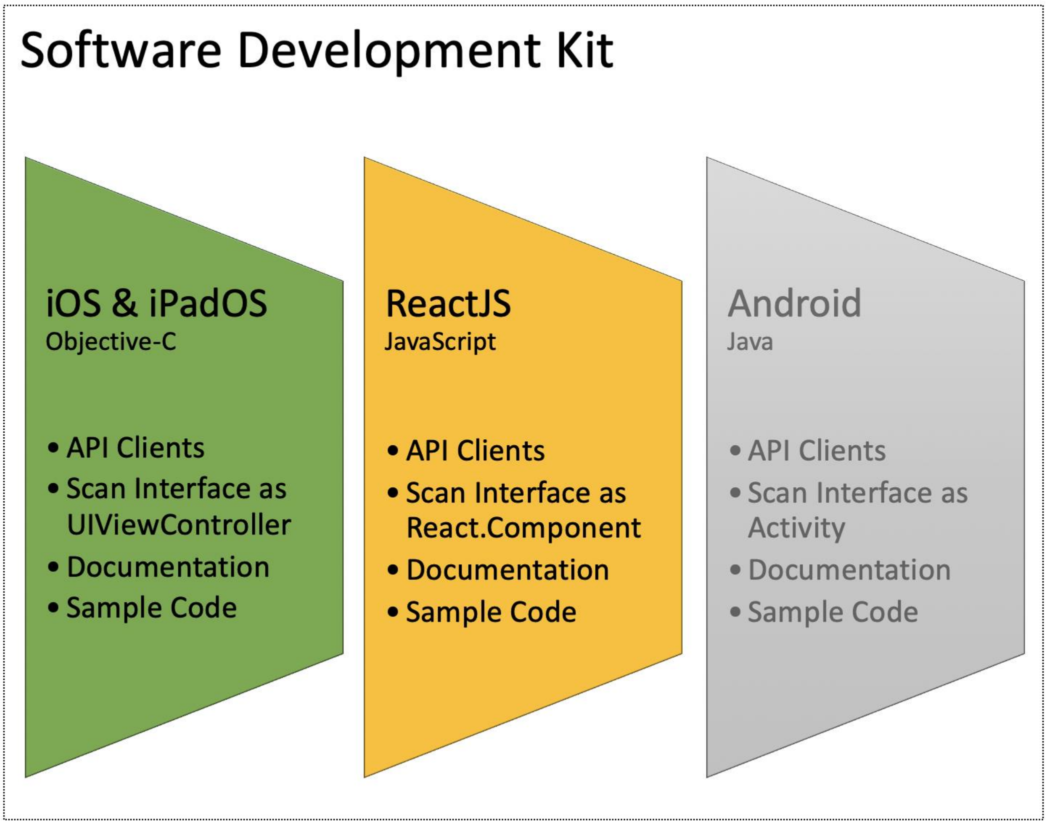


Figure 8-1: Software Development Kit components risk assessment.
The green colour indicates QuantaCorp sees no risk for implementation.
The yellow colour indicates QuantaCorp sees risk for implementation.
The grey colour indicates QuantaCorp has not looked into implementation risks.

8.3 API Clients

The SDK's API Clients are HTTPS clients for the QuantaCorp API endpoints in the native development language for a given platform. By providing these wrappers, third party developers save precious time not having to go over the details of the QuantaCorp API documentation and not having to write their own HTTPS clients.

In the eTryOn project, the number of API Clients is fairly limited. The only endpoints that require wrapping are the ones related to creating a scan and this actually boils down to a single POST request, which is custom made for eTryOn.

As mentioned in chapter 3, we work on a fork of our existing platform for the entirety of the eTryOn project. For the project, we will be adding a new endpoint to our public API. The new endpoint is a POST request for the creation of a new scan resource but specifically for eTryOn. What is different in this call compared to our regular scan POST request, is the addition to the request body of signed URLs and a JSON Web Token (JWT) to verify the authenticity of the caller. With the JWT the QuantaCorp Cloud can verify the authenticity of the caller with the eTryOn Cloud and the signed URLs are used to upload the 3D model and a thumbnail image to the eTryOn Firebase Cloud Storage.

To better understand the communication between the eTryOn application, the QuantaCorp Cloud and the eTryOn Cloud we will briefly go over the eTryOn pipeline.

Every eTryOn application will authenticate their users using Firebase Authentication. Once a user is logged in, he or she can start creating an avatar by requesting the eTryOn app to present the scan component. To show the scan component of the QuantaCorp SDK, the eTryOn app first instantiates a new scan component. How the scan component works will be discussed in section 8.4, for now it suffices to know that the scan component will capture the images necessary for a scan upload, and it also generates the metadata required for that upload. Once the pictures are captured and the metadata is generated, the scan component returns this data to the eTryOn application. Next, the eTryOn application generates a JWT for user authentication and validation and it generates two signed URLs, one for the 3D model and one for a thumbnail. The eTryOn application then forwards the data it received from the scan component, along with the signed URLs and the JWT to the SDK's API Client responsible for the creation of a new scan resource for eTryOn. The upload will happen asynchronously and through pub/sub the eTryOn application will be kept in the loop on the upload progress. Once the scan is uploaded, the QuantaCorp API will first use the JWT to validate if the caller is an authentic eTryOn user. On a negative validation, the request is discarded. On a positive validation, the public API forwards the images to the segmentation server. The generated silhouettes are then sent to the matching server, along with the colour images and the metadata. Once the matching is complete, a 3D model is generated and gets uploaded to the eTryOn Cloud Storage using the signed URL destined for the model. Finally, a thumbnail image is generated, and gets uploaded using the signed URL destined for the thumbnail.

That concludes QuantaCorp's part in the eTryOn pipeline. The eTryOn Cloud will have a Cloud Function running that checks for new file uploads and will automatically trigger the next step in the eTryOn pipeline.

8.4 Scan Component

The scan component of the QuantaCorp SDK is responsible for the capture of all the data required to generate a 3D model.

The first thing the implementer of the QuantaCorp SDK has to do after the scan component is instantiated, is initialisation of the scan component with the data it is unable to capture. During the initialisation of the scan component, all required data is sent to the scan component. The data then gets validated. Should there be data missing, the scan component will prompt the user for this data when it is presented. So possibly, the first view shown to the user is a view that requests all required information. Currently, the required data that cannot be captured by the scan component itself is the height of the person. In the future we may require additional data such as gender or weight.

After all required data is received, the user will be shown introductory pages. These pages will explain how to use the scan interface as succinctly as possible. The instructions could be displayed in the form of short video clips or well-designed graphics. The user will be given the option to skip these pages, because we should not force this on the user every time he/she takes a scan. Using a local setting we will remember if the user prefers to skip seeing the introductory pages or not. The user will be able to consult the introduction pages again if needed from within the scan interface.

After the introduction, the user will land on the scan interface. The scan interface will most likely feature some sort of silhouette in which the user has to fit the model. For right now, besides a silhouette we also expect to see some visualisation of the device tilt. We want to make sure the user is holding his/her smartphone in an upright position when capturing the photos. Once both pictures are taken, the data together with the captured metadata are sent to the implementer of the QuantaCorp SDK. What happens next is described in the previous section 8.3.

9 Conclusions

In this deliverable we showed some of the issues with our current body space in relation to the 3D model's appearance. We explained that the conclusions drawn from the various artefacts and missing level of detail put us on the path of evaluating our existing body space. We also discussed an approach to generate a photo-like texture to give our 3D models a photorealistic appearance, but explained why this approach was let down, in favour of an artistic-interpretation of photorealism.

Further in the document, we explained the weaknesses of our current body space. These problems were illustrated in detail with corresponding figures. It was decided that improving the current body space would be too costly and too time consuming to get done within the eTryOn project. Alternatives were looked at, like SMPL and STAR. We moved forward by testing, implementing and evaluating the STAR-model. Several different matching algorithms were also implemented and evaluated. The STAR-model shows promising results both in measurement accuracy and visual fidelity. As a result, it was decided to proceed with the STAR-model for the eTryOn project.

In chapters 6 and 7, the importance of accurate segmentation was brought forward. Within the scope of the eTryOn project the current method was evaluated and we concluded that our method is sensitive to big deviations in segmentation. Thus, it was decided to improve our current segmentation pipeline. Given the progress in the state of the art in terms of refinement, we decided to implement a refinement step to complement our existing segmentation. This method was then evaluated and significantly improved the results. If high-resolution images are available, refinement will be applied.

In chapter 8 we presented the SDK and where it fits in the eTryOn project. We decided the SDK will contain a scan component and QuantaCorp API wrapper for every supported platform by eTryOn. We explained how the SDK, the QuantaCorp Cloud and the eTryOn Cloud communicate with one another. We discussed how the scanning component will work and gave an idea of what it will look like.

10 References

4.1. Kazhdan, M., & Hoppe, H. (2013). Screened poisson surface reconstruction. ACM Transactions on Graphics (ToG), 32(3), 1-13.

5.1. Kraevoy, V., & Sheffer, A. (2005). Template-Based Mesh Completion. In Eurographics Symposium on Geometry Processing.

5.2. Zhao, W., Gao, S., & Lin, H. (2007). A robust hole-filling algorithm for triangular mesh. Visual Computer, 23(12), 987–997. https://doi.org/10.1007/s00371-007-0167-y

5.3. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: A skinned multi-person linear model. ACM Transactions on Graphics, 34(6). https://doi.org/10.1145/2816795.281

5.4. Dibra, E., Jain, H., Oztireli, C., Ziegler, R., & Gross, M. (2016). HS-Nets: Estimating human body shape from silhouettes with convolutional neural networks. Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016, 108–117. https://doi.org/10.1109/3DV.2016.19

5.5. Guo, X., Xiao, J., & Wang, Y. (2018). A survey on algorithms of hole filling in 3D surface reconstruction. Visual Computer, 34(1), 93–103. https://doi.org/10.1007/s00371-016-1316-y

5.6. Smith, B. M., Chari, V., Agrawal, A., Rehg, J. M., & Sever, R. (2019). Towards Accurate 3D Human Body Reconstruction from Silhouettes. Proceedings - 2019 International Conference on 3D Vision, 3DV 201

5.7. Gabeur, V., Franco, J. S., Martin, X., Schmid, C., & Rogez, G. (2019). Moulding humans: Non-parametric 3d human shape estimation from single images. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 2232-2241).

5.8. Osman, A. A. A., Bolkart, T., & Black, M. J. (2020). STAR: Sparse Trained Articulated Human Body Regressor.

5.10. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., & Davis, J. (2005). Scape: shape completion and animation of people. In ACM SIGGRAPH 2005 Papers (pp. 408-416).

5.11. Robinette, K. M., Daanen, H., & Paquet, E. (1999, October). The CAESAR project: a 3-D surface anthropometry survey. In Second International Conference on 3-D Digital Imaging and Modeling (Cat. No. PR00062) (pp. 380-386). IEEE.

7.1. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. http://lmb.informatik.uni-freiburg.de/

7.3. Lin, G., Milan, A., Shen, C., & Reid, I. (2017). RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. Proceedings - 30th IEEE Conference on Computer Vision and Pattern

7.4. Yuan, Y., Xie, J., Chen, X., & Wang, J. (2020, August). Segfix: Model-agnostic boundary refinement for segmentation. In European Conference on Computer Vision (pp. 489-506). Springer, Cham.

7.5. Cheng, H. K., Chung, J., Tai, Y. W., & Tang, C. K. (2020). CascadePSP: Toward Class-Agnostic and Very High-Resolution Segmentation via Global and Local Refinement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8890-8899).