DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Analysis of antique Statues using 3D Scans

Daniel Marius Haselbeck

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Analysis of antique Statues using 3D Scans

Analyse antiker Statuen anhand von 3D Scans

Author:	Daniel Marius Haselbeck
Supervisor:	Prof. Gudrun Klinker, Ph.D.
Advisor:	Dr. David A. Plecher, M.A.
Submission Date:	05.15.2024

I confirm that this bachelor’s thesis in informatics is my own work and I have documented all sources and material used.

Munich, 05.15.2024 Daniel Marius Haselbeck

Acknowledgments

I would like to thank the Museum of Casts of Classical Statues in Munich for their continued enthusiasm for this thesis and for always answering all of my questions.
Furthermore, I would like to thank the Open Source Community for all the work they put into the software that made this thesis possible.
I also want to thank my advisor, Dr David Plecher, for helping and providing me with many suggestions throughout this project.
A special thanks also goes to my Friends and my Family, who all stuck with and supported me throughout my studies and enabled me to write this thesis in the first place.

Abstract

This project involves Ancient Roman sculpture portraiture, Photogrammetry, 3D object manipulation, the development and testing of some simple algorithms for comparing 3D objects, and the creation of an android application using Augmented Reality (AR).

We compare two ancient statues to each other; one of them shows signs that it was reworked from a different looking original, and the other one is just an ordinary statue. The hypothesis we investigate is whether the second statue can serve as an accurate stand-in for the first statue's original look.
To do this, we analyze the statues directly in person and perform 3D scans using images from a smartphone camera. Then, after some postprocessing steps, we also analyze these statues digitally and look for a way to superimpose the ordinary statue on top of the reworked one.
We also test a hypothesis about the similarity of areas on the first statue that were possibly left untouched during its rework and the corresponding areas on the second statue using a simple algorithmic approach.
Finally, we create an AR phone application to showcase the results from this analysis and place them into historical context.

Contents

1 Introduction

1.1 Outline and Goals

In this thesis, we analyze ancient Roman portrait statues for abnormalities and similarities by comparing a portrait that had its face reworked into ththat of a different person to aother, preserved portrait of the former faces. For this, we partnered with the Museum für Abgüsse Klassischer Bildwerke ("Museum of Casts of Classical Statues") in Munich, who provided us access to their catalog, specifically some pairs of statues they had already picked out as appropriate targets beforehand.

The entire project is structured into multiple steps. First, we must create 3D scans of both statue portraits, after which we can then post-process and digitally analyze them for any peculiarities that may not be as apparent when examining the statues in the real world. For this, the 3D scans should resemble reality as closely as possible and capture not just the general head shape and texture but also represent the more intricate geometrical details from the hairline, ear shapes, and face structure.

Finally, we want to use these scans to create 3-dimensional **Augmented Reality (AR)** markers for use in an educational application for the Museum of Casts. This application should both combine the two scans in an estimate for the reshaped statue's original look and showcase the unusual features of the reworked statue. Additionally, making the application work in a browser would be preferable over a phone application, as maintaining such an application for the different app stores is much less feasible for smaller museums than a self-hosted website, and it should provide a more user-friendly experience if museum visitors did not have to download another app onto their phones just for a single visit.

1.2 Historical Background

In Ancient Rome, statue portraits were a way to honor and preserve the images of gods, ancestors, and military leaders. Many influential Roman families would decorate their mausoleums with replicas of their deceased, and statues of Roman emperors and gods were displayed in most public places for everyone to see. Naturally, all emperors were granted these statues once they came into power, which were then distributed

throughout the Roman empire. Oftentimes, the way these powerful people would have themselves portrayed changed during their lifetime, for example after reaching a military, political, or career milestone or simply due to personal preference.
Due to this, archaeologists now distinguish the portraits of most Roman emperors into several ’types’ [Arcb]. While different types are set apart by their facial expressions and features, jewelry, and hairstyles, statues of the same typus share many similar features among each other, frequently down to singular hair strands on their forehead. This is because sculptors, in order to meet the demands of the entire empire, were likely commissioned to copy from the same templates or manufacture in bulk.

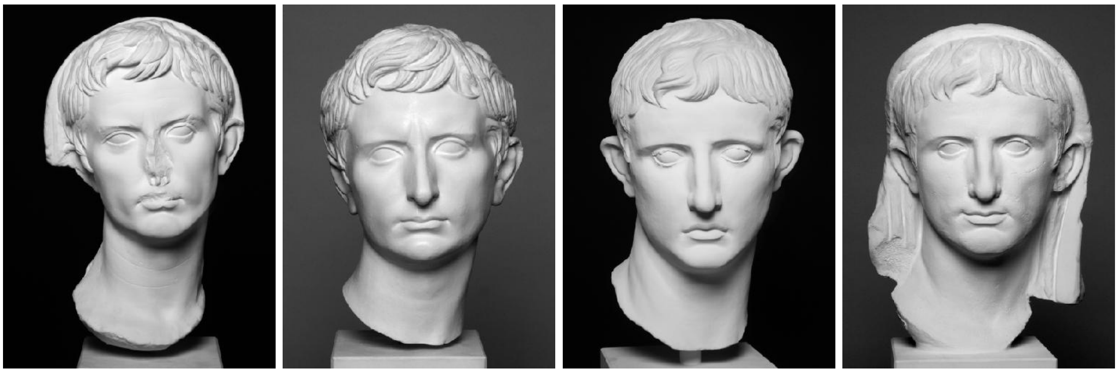


Figure 1.1: Two types of the Augustus statue: Octavian-Typus (left) and Prima Porta-Typus (right) [Arca]

Still, statues of the same type are anything but carbon copies of each other, and Romans likely viewed copying as an art form in itself [Ang14], with the sculptors introducing their personal flavor and style into each work, even the copied ones. Therefore, archaeological type classification is not as straightforward as it seems at first glance.

Another practice in Ancient Rome was the so-called ’damnatio memoriae’[Len20], Latin for ’condemnation of memory’, wherein disgraced influential persons had their actions removed from records and their effigies defaced. Whether the goal of this practice was to erase its targets from history or just to brand their legacy as shameful as well as how rigidly it was enforced are still debated [Kar19], and many depictions of the affected remain intact to this day. A possible explanation for the intact statues is that they belonged to private collections of wealthy individuals and that this practice, which mainly focused on public displays, did not reach these collections. Additionally, statues made from marble were frequently reworked into portraits of other, usually related, famous Romans instead of being destroyed outright, presumably to salvage the valuable raw material [Arcd].

On these reworks, we can often still observe marks from the revision process and remnants of the statue's original look, especially in non-front-facing areas.

These two facts mean that we can identify pairs consisting of one original statue and one reworked statue that, with high probability, once looked very similar. From these similarities, we should be able to better understand the processes that sculptors employed and how exactly the statues were reworked. It may also provide historical insights into the contemporary practices and politics of the Ancient Rome Imperial times. The specific portraitures we examine during this thesis belong to emperor Caligula, who had become a target of damnatio memoirae after his death, and to his successor Claudius, with his statue showing signs of being reworked out of one of Caligula's [Arcc].

2 3D Models of Real-world Objects

2.1 Tools used

A quick overview of the tools employed to scan, post-process, and analyze all models used in this thesis.

- Meshroom[Ali]
- Meshmixer[Auta]
- Meshlab[IST]
- CloudCompare[Ope]

2.2 Creating the 3D Models

2.2.1 Introduction to Photogrammetry

Before we can use **Augmented Reality (AR)** inside an application, we first need a way for it to recognize and track the objects we want to augment, which are the busts of certain Roman emperors in this case. To do this, we create digital 3D objects of these busts by taking many pictures of the statues from all angles and turning them into 3D scans using photogrammetry.

There are several important factors to take into account when taking photos for photogrammetric purposes. First, the object we choose to replicate should have a rigid and detailed surface. Photogrammetry software extracts feature points it can recognize on multiple photographs in overlapping areas; the more feature points it can find, the more detailed the final result will turn out. Rigid, colorful, and detailed surfaces usually make for the most accurate and easiest to recognize feature points. These qualities will also be beneficial when tracking said object for augmentation later down the line because, just as for humans, more unique objects are easier to identify for machines as well.
Unless we can take all photographs in the same instant, which would require an

expensive setup with multiple cameras, the scene around our target must change as little as possible in between shots. Especially the object itself may not have any moving parts, but even out-of-focus background objects, such as passers-by in public places, should be avoided if possible.
Besides their lack of color contrast, statues usually fit these criteria quite well. Therefore, it should be no surprise that they are commonly used as showcase examples for photogrammetry.

2.2.2 Capturing a first Photogrammetry Model

Now, it is time to create a first 3D model of a real-world object. We do not start with the statues themselves, as they are in the Museum of Casts' storage, and we will be unable to influence the conditions under which we perform these scans very much once we get to them. To get a better grasp of photogrammetry, we first capture a different subject as a trial run. Here, we can make many more adjustments to our workflow if necessary. For this, we took 16 pictures of a stuffed animal at eight different rotations and two different heights (see 2.1), simulating a turntable setup. We choose the stuffed animal as - despite it being made of cloth - it is quite rigid and resembles a statue closer than anything else available at the time. For turning our images into photoscans, we use the photogrammetry software 'Meshroom' [Ali]. After running them through the Meshroom photogrammetry pipeline, we get the following result, as seen in 2.2. We can quickly tell this is not at all what we expected; however, there still is some useful information to extract here. In the left third of the first picture, we can see all of the estimated camera positions - all 16 pictures could be represented, but they are all grouped together instead of evenly spaced around the rotated object. This means that the software did not use the stuffed animal for reference but rather its background,

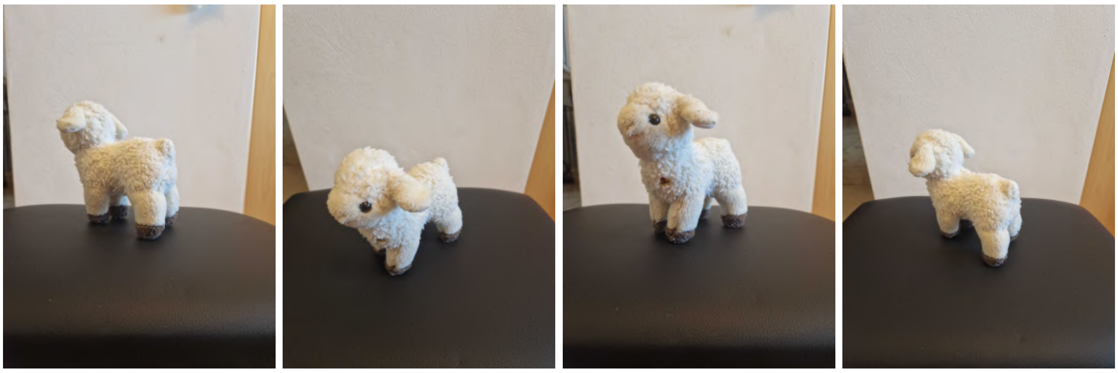


Figure 2.1: Stuffed animal photographed at different rotations and heights

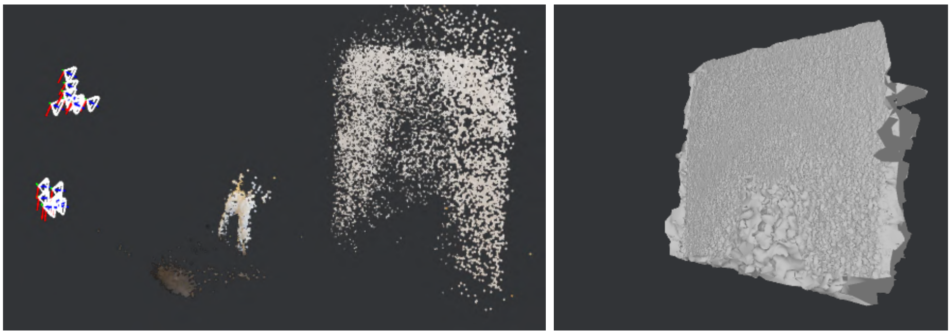


Figure 2.2: Structure from Motion with estimated camera locations, point cloud, and resulting mesh

which is made even more apparent by the results in the picture on the right: the software only managed to recreate the wall behind the sheep, and the few points of the sheep's surface were all discarded.

As this result is nowhere near satisfactory, we now retake the photographs and, instead of rotating the sheep, move the camera around after each image. We also end up taking 32 pictures instead of the earlier 16 by taking twice as many pictures around the target at both height levels. We also change backgrounds in favor of a color scheme that is more different from the sheep's. The resulting mesh is much better, but the Meshroom only managed to match 13 of the 32 photographs. Looking at the data of these 13 photographs, we find that their camera locations were estimated very accurately and that some of the background items were included in the mesh, namely parts of the piano and the edge of the desk seen in 2.3.

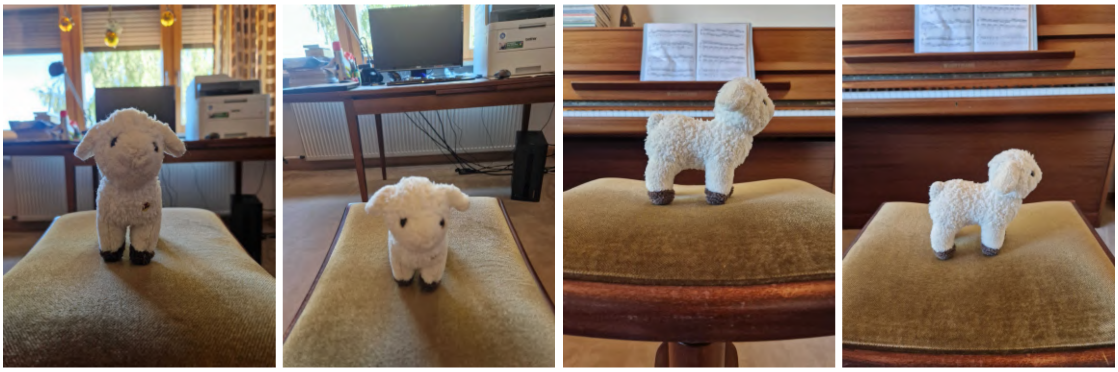


Figure 2.3: Successfully included photos for the second photogrammetry attempt

While we are not interested in representing background items in our final scans,

them being somewhat in focus seems to lead to better results, and we can conclude that a larger depth of field (distance between the furthest in-focus elements of a photograph[Nas]) will improve the estimation of camera locations.
Again, looking at the Structure from Motion(SfM) and Meshing nodes of Meshroom's output yields more information (see 2.4). Most apparent is the fact that all reconstructed cameras are from images taken in adjacent positions. This suggests that, besides image quality, the pictures were taken too far apart, as there seemingly was not enough overlap between neighboring pictures for the software to follow all the way around the stuffed animal. Furthermore, more photographs from the upper height level contributed to the result, which might be partly because the pictures from the lower angle have a worse depth of field. In addition, lighting was mostly provided from only one direction, which led to unwanted shadows. Shadows not only possibly lead to discoloration in the final product but can also be misinterpreted as surface artifacts, such as elevations or edges, in the meshing process, resulting in a less accurate 3D representation. For the final attempt with this subject, we switch backgrounds once more. This time, we obtain relatively diffuse lighting from at least a ∼180-degree angle as the room has multiple

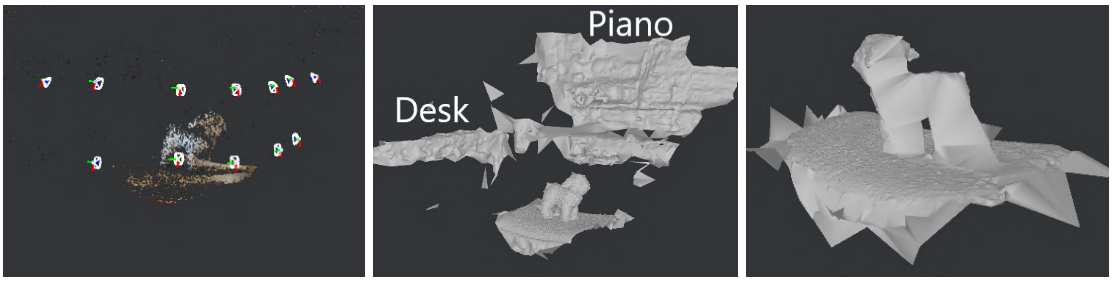


Figure 2.4: **SfM** and mesh with background items of the second try

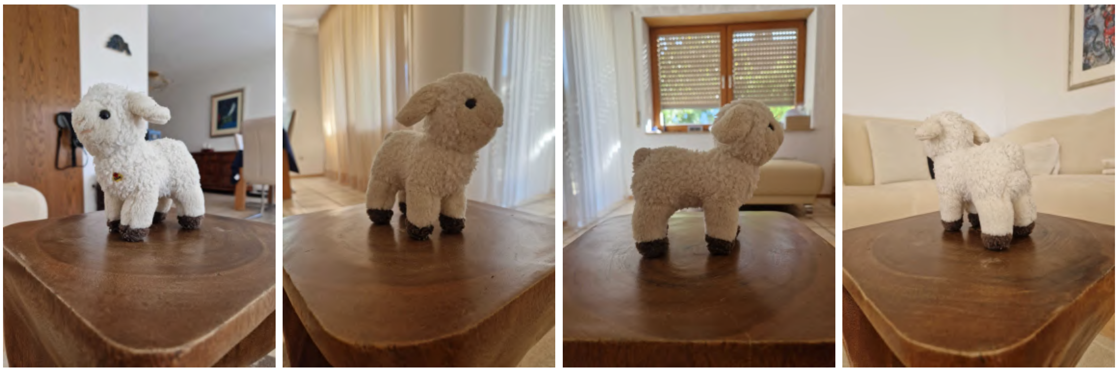


Figure 2.5: Images for the third photogrammetry attempt

windows and transparent curtains (see 2.5).
Of course, this still leaves a brighter and a darker side to the scene, but we can avoid shooting directly against a light source. We also add another horizontal layer whilst taking the photographs, yielding a total of 48 input pictures, 46 of which could be represented in the photogrammetry process. Again, the missing photos are adjacent and in the bottom layer, where larger parts of the picture are out-of-focus because they include more of the surface on which the stuffed animal was placed, leading to worse depth of field. This time, the mesh of the subject itself has details on all sides, with exceptions below the ears and in between the legs, where it is much sparser, as the camera did not capture these areas very well (see 2.6).

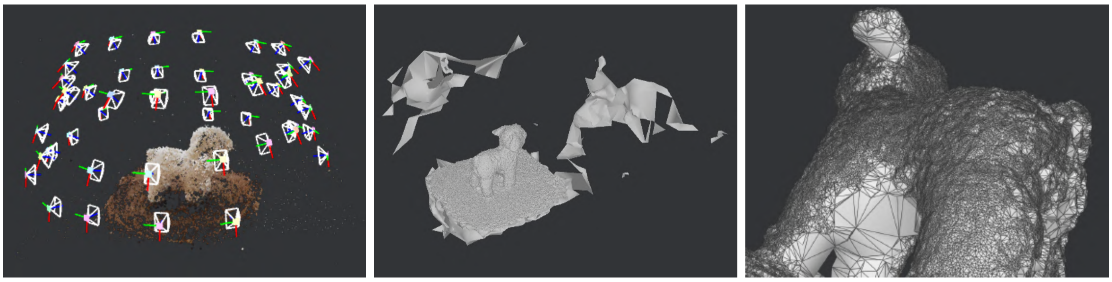


Figure 2.6: SfM and mesh of the third try

2.2.3 Scanning the Statues

We now move on to the subjects of this thesis, portrait sculptures of ancient Roman emperors. For this, we go to the "Museum of Casts of Classical Statues" in Munich, Germany, where we meet with Drs. Andrea Schmölder-Veit and Nele Schröder-Griebel to discuss certain pairs of statues that are likely quite similar to each other.
Here, we are let into the storage of the museum to scan a statue pair of Caligula and Claudius, which are the focus of this thesis, and one of Nero and Titus. Since these statues are both valuable, somewhat brittle, and also heavy, the simplest method is to perform the scans on-site. This comes with the drawback of not being able to control the lighting very much, and we had much better lighting on one side than on the other. The lighting also comes mainly from a wide window spanning across the entire hallway, but there is no direct sunlight influx.
Running the Meshroom pipeline with many images is a process that requires a significant amount of computational power and takes multiple hours even on high-end computers, so we cannot evaluate the results of the scans immediately and will have to return on another date to perform additional scans if the results are insufficient for our purposes. In total, we perform a scan of each statue on three different occasions,

trying a slightly different approach each time. For the first scans, We try to alleviate the one-sided lighting problem by performing the scans twice for each statue, once with the light source at the front and once at the rear, for a total of four scans, two for each statue. Afterwards, we may combine two scans of the same statue into a single mesh with the high details on all sides as an extra step.

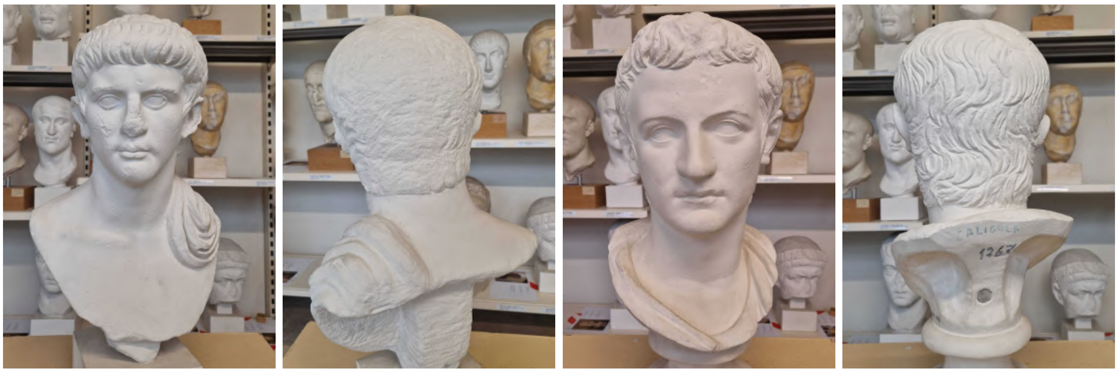


Figure 2.7: The Claudius and Caligula portraits at the Museum of Casts

On the other hand, shooting in the storage of the museum means that most pictures contain many unique and detailed background items, which makes it easier for Meshroom to figure out at which angle they were taken. Lastly, we make an additional effort to take multiple pictures behind each statue's ears in hopes of representing their details as accurately as possible, even though those parts are only visible in pictures taken from very few angles and otherwise blocked from view by the skull or the ears themselves. Each scan is comprised of around 60 to 90 images, and all of them have sufficient overlapping areas and unique details to be included in Meshroom's photogrammetry pipeline.

After evaluating the results from these scans, our worries about the lighting are confirmed; the well-lit surfaces of the statues were captured properly, yielding a highly detailed and dense mesh, but other areas show large holes (see 2.8). Thankfully, the fine geometrical details that Meshroom did manage to capture in the affected areas are quite accurate, so the idea of combining two scans into one is still feasible, and we do not have to worry about introducing unnecessarily noisy data from one mesh's bad areas into the complementary mesh's good areas and vice versa. For our second attempt, we want to avoid having to combine two different meshes of the same statue, so we take many more pictures of each statue, for a total of 350 to 400 photographs this time around. Again, all of the pictures make it into the pipeline, but for some reason, the 3D representation of Claudius is much better than that of Caligula (see 2.9(1 and 2)). Again, the lighting at the darker side opposite the museum storage hallway's

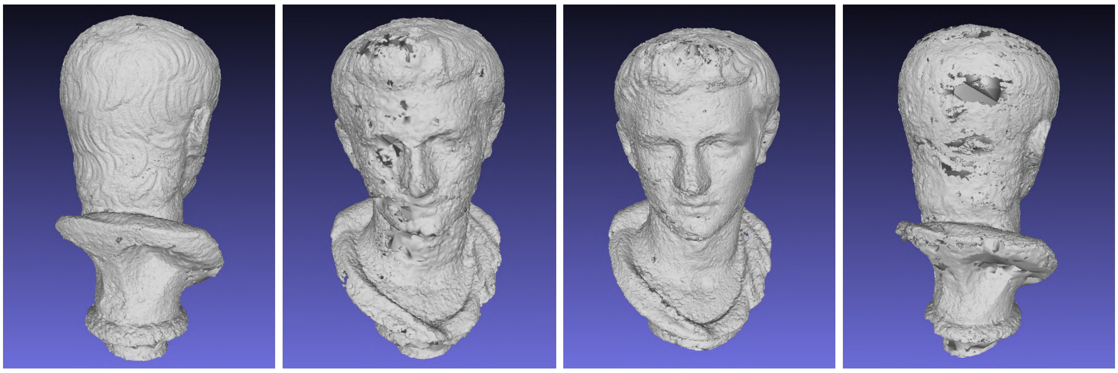


Figure 2.8: Incomplete scans from both sides

window seems to be the primary problem inhibiting the quality of our photogrammetry reconstructions, so, in the third attempt, we bring a spotlight and orient it against the wall. While still not optimal, the lighting here is much more diffuse than in the first two tries, and we wind up taking even more photographs of both statues, around 500 and 600 respectively. However, in 2.9(3 and 4), we observe that the meshes still show large holes in some areas. There are two possible explanations for this. We took many pictures, and the additional overlap caused the photogrammetry unnecessary complexity. At some point during the photo taking, the sunlight from outside the window was obscured by clouds, and at other points in time it was not. This certainly caused some texture issues in the final steps after the mesh has already been created, but likely also negatively affected other steps of the pipeline.

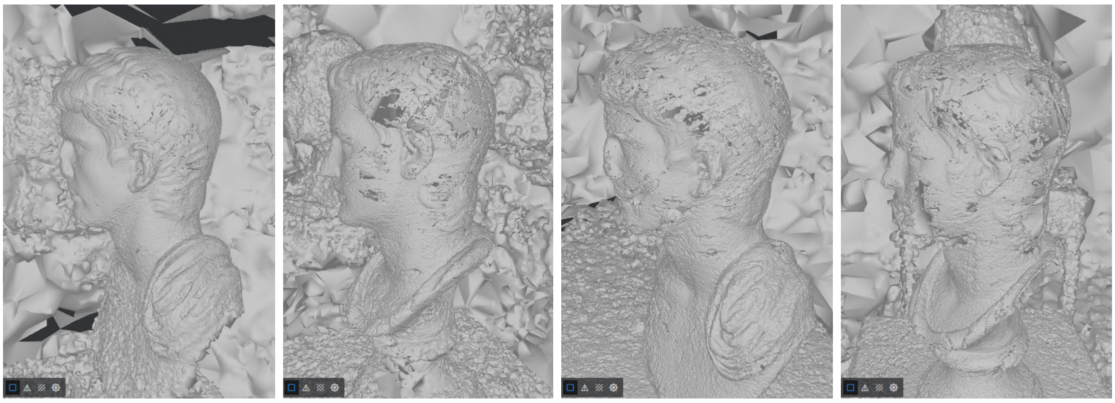


Figure 2.9: Mesh representation from the second and third scan attempts

2.2.4 Post-processing the scans

Before we may use the photoscans from the Meshroom output, we must first post-process them. As we have seen in earlier scans as well, Meshroom cannot tell that we are only interested in the statues, so it recreates any background items and details as well, if possible. Naturally, the further away from the center you look, the less accurate the scan data becomes, as the corresponding image data also becomes less dense and more likely to be out-of-focus. This is clearly visible in 2.10(2), where we can observe fine geometrical details on the statue's face, whereas the mesh becomes more and more warped as we look towards its neck and chest area, which are much smoother in reality. Even on the most accurately captured areas of the scan, we observe small lumps and dents in its surfaces that are also not present on their real-life counterparts. This is one of the main drawbacks of photogrammetry as opposed to Laser Scanning, which can precisely measure distances. Still, a more complete scene hereby points towards better results, as that comes complementary with sufficient and high quality data from the images.

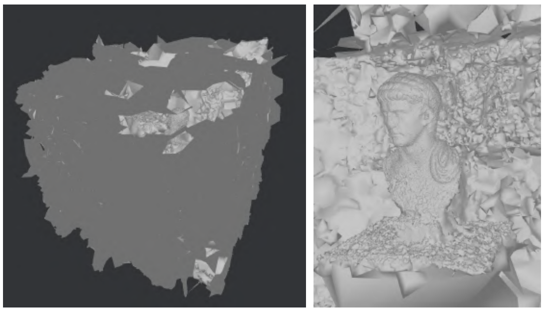


Figure 2.10: The entire scene with the statue in the center

To get rid of the mesh's unneeded parts, we first use 'Meshmixer' to manually extract the statues' mesh representations out of each scene. This is quite simple since the bases of each statue differentiate them clearly from the other parts of their meshes. As mentioned earlier, we plan to merge two incomplete meshes of the statues into a single, higher quality one as part of our first scanning workflow.

We can do so with the 3D mesh processing software 'CloudCompare'. First, we simply load both meshes into the software separately. Since the meshes come from different scans, they differ in scale and position within the 3D space. Thus, before combining, we must align them accurately on top of each other. The two scans represent the same object and there is significant overlap between them, so we can easily use the point-picking tool to find anchor and reference point pairs on the meshes. With sufficiently many such pairs, the tool will minimize the distances between every reference and anchor point, aligning the meshes in the process (see 2.11). Afterwards, we can combine

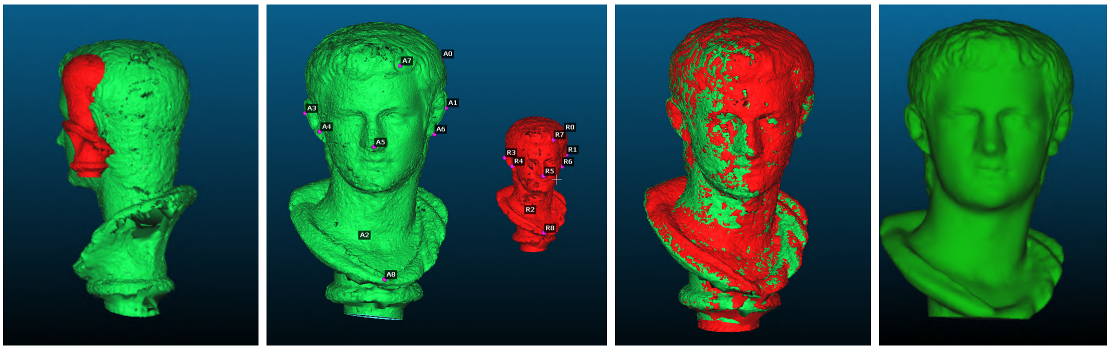


Figure 2.11: Aligning and combining the two meshes via point picking

the two meshes into one by merging their vertex **Point Clouds**. Now, we can calculate new surface details that include the data from both scans by reconstructing a mesh from their merged vertices. For this, we use CloudCompare's 'PoissonRecon' feature, where we can also specify how dense the reconstructed mesh will be. If we set the density too low, we may lose details; If we set it too high, the result will include interpolation noise from insufficient input data. Generally, we try to aim for as low a density as possible without losses, as having less data for the same quality is more efficient.
The reconstruction will also not rid us of all photogrammetry noise, so we manually apply some smoothing to the statue's base, face, chest, and nose in Blender [Ble].

2.3 Analyzing the Models

2.3.1 On-site

The quickest and most straightforward way to analyze the Statues is in person, which we were able to do when we went to scan the statues at the Museum of Casts' storage in Munich. Here, Dr. Schröder-Griebel pointed out the visible irregularities that distinguish the Claudius statue from ordinary ones.
On many reworked statues, their protruding ears are the most apparent giveaway. Ears are difficult for sculptors to process due to the same reason that they are often missing from statues in museums - the auricles are only connected to the head by a narrow part, making them prone to breaking off with just minimal force. Additionally, even if that did not pose a problem, it is impossible to reshape an auricle without adding new material to the old one because they are hollow inside. A different ear shape would have to be carved from the skull after shaving off the entirety of the old one, which would lead to a much smaller head than intended, so the original ears were usually

kept largely unchanged during reworks. With the rest of the head having material carved or chipped off, the original ears of reworked statues commonly protrude from their new shapes.

Because of the extra care with which the sculptors had to regard the statue's ears, much of the evidence for its rework and its original shape can be found distributed in close proximity around and especially behind them, as the latter parts are additionally shielded from view when looking at the statue from the front. At the place where the outer ears meet the cranium, we can see small holes that presumably stem from manual hand drills used during the revision process to shape the new hairline. While it is very common for ordinary statues to have a small, blank rim around their ears, such holes would usually be ground down and polished smooth until they disappeared. Their existence here gives more ground to the theory that statue revisions were commissioned as a way to save money and, in this case, not just the material but also the labor costs of sculptors.

Furthermore, these holes only appear at the middle and upper parts of Claudius' ears, suggesting that the hair detail behind and below them are leftovers from the original statue. We also take note of the erosion damage that both statues have gone through since ancient times. Caligula remains mostly intact, the only significant damage being at his right earlobe, which has partly broken off. For Claudius, the damage is more severe; the statue is missing most of its nose, both ears are missing material at their outermost parts, and his left earlobe is also missing.

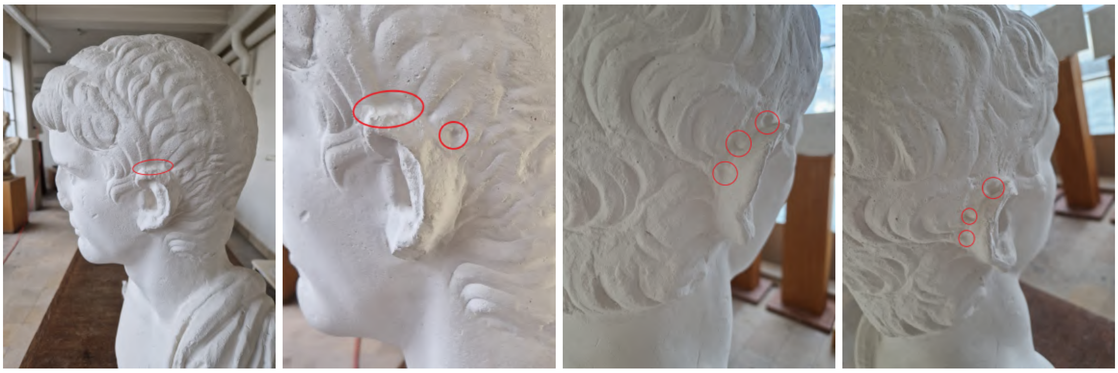


Figure 2.12: Visible drill holes on both sides of the statue

Claudius' portrait also lacks hair detail at the back of its head, not due to damage but rather because it was never created in the first place, providing further evidence for the theory that statues were reworked instead of replaced partly for cost-cutting purposes. We also notice this statue's narrow forehead and the unusually thick hair at the sides of its temples, but we cannot draw any immediate conclusions from these.

2.3.2 Digitally

The main advantage we gain by analyzing the statues digitally instead of in the real world is that we may scale and rotate the objects freely. This allows us to search for a fitting alignment of the two heads so that we can overlay the original statue on top of the reworked one. Such an alignment lets us draw conclusions about which parts of the statues were changed the most during the rework and which were left mostly untouched. Just as during post-processing, we again use 'CloudCompare' to superimpose the two scans onto each other via point picking, but the process is much more ambiguous for meshes that describe distinct shapes than for the same shape since we have to choose our anchor points in a different way.

There are three factors that play a role in finding a proper alignment, namely the relative size of the two scans, their position, and their orientation towards each other. First, we try to find out what scale best fits the two statues. Suppose the two scans were actually from the very same block of marble, captured once before and once after the rework. In such a case, we could simply take their real-world measurements as their relative sizes. However, since this is not applicable here, we need a different approach. As the ears of marble or other stone sculptures are very difficult to process - the auricles are only connected to the head by a narrow part, making them prone to breaking off easily - we can make the reasonable assumption that the sculptors did not remove much or any material at all from these parts. Therefore, resizing the statues in a way that equalizes the distance between both statues' ears or makes them the same size might yield an appropriate relative scale. As seen in 2.13, the two portraits' ears differ slightly yet visibly in shape, and there is damage on Claudius' left ear (far left) and Caligula's right earlobe (far right). This leaves some leeway for the exact place of the anchor points we choose and, therefore, also for the final scale between the sculptures. Choosing the ears as anchors for their scale already locks in the statues' relative positions as well, which leaves only the orientation of the two, so the next thing to consider besides just the ears' shapes is their orientation towards the rest of the head. If we choose to anchor our alignment at the ears and they are tilted in different ways, the two statues may end up malaligned at the neck, up to some rotation around the axis that connects both ears. While there is no exact way for us to obtain this rotation, the museum selected the two statues specifically because of their similar head shapes, so we make an additional assumption to also consider the statues' necks while looking for a proper way to superimpose the statues on top of each other.

Finally, we can come up with a best guess of how to overlay the Caligula portrait onto Claudius. From this alignment, a few details become apparent immediately. A few features of Claudius still protrude from Caligula's portrait, something that could not happen if they were a true pair of copies from the same template. As seen earlier, we

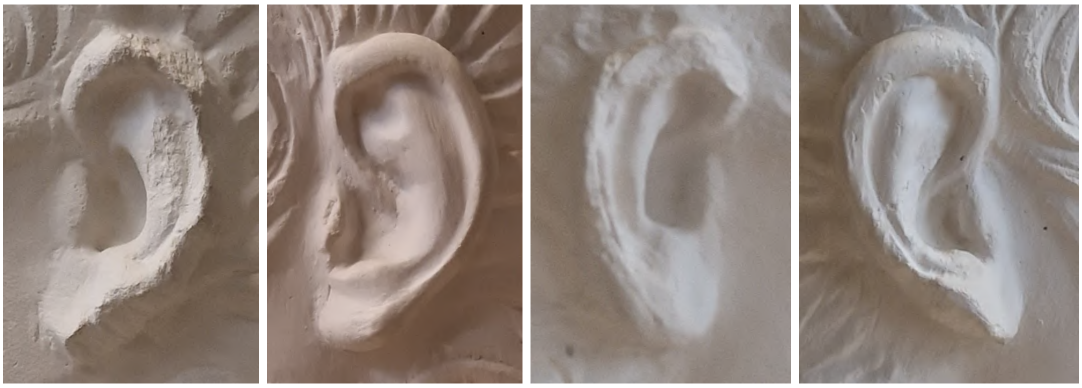


Figure 2.13: Closeup comparison of the statues' ears

already know that the ears do not share the same shapes, leading to some overlap there. Additionally, the top of Caligula's head stands at a steeper angle to its back than Claudius', which follows a rounder curve and in turn leads to some outward protrusions at the top (see 2.14).

Still, most features of the two statues are roughly aligned, except for their faces. Claudius has a slightly narrower face than Caligula, and his chin is much higher as well, making the entire statue seemingly lean forward. This becomes most visible when viewing both facial profiles from the side. From this, we can reason that revisions were done beginning at the top of the face with its hairline and then working downward towards the chin, shaving off more and more material the further down they moved in order to keep the face profile consistent.

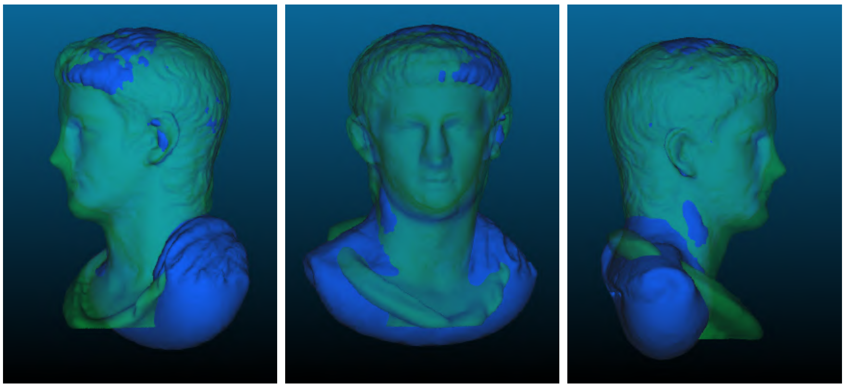


Figure 2.14: Alignment at the ears

On-site, we had recognized that this Claudius portrait's forehead ridge is much less pronounced than in other depictions of his but had chalked it up to coincidence as we already knew that the face must have been remade entirely during the rework of the statue. In hindsight however, we now know that the less dominant forehead ridge is not a remnant of Caligula's original face but rather a consequence of the rework that confined the forehead into a narrower space and left less room for such detail.

In fact, after measuring parts of both scans' facial features, we find that Claudius' nose, mouth, and eyes are about ten percent smaller than Caligula's with their ears scaled to the same size. Caligula's forehead also measures similar to Claudius' horizontally but is fifty percent larger vertically. If we additionally approximate their respective lower face boundaries by measuring how far both statues' earlobes are from where the areas below their chins transition into the adam's apples, it seems as though Claudius' face has been moved upwards by said vertical difference.

In 2010, William Storage published a very similar project comparing a possibly reworked statue of Augustus with another one of Caligula and arguing staunchly against the notion that the considered Augustus portraiture could have been recarved from any known Caligula portrait type [Sto09]. Their approach, however, differs from ours in that it disregards the ears as anchor points. They argue that, with so many ancient portraits missing their ears, it becomes much harder to verify results that rely on them as the number of suitable candidates for counterchecking decreases drastically.

The proposed reworking method that they argue against also makes the assumption that the main facial features - eyes, nose, mouth, and chin - were, at most, only slightly revised during reworks, with its main focus being on the hairline instead. Thus, they align the two statues at their eyeballs and quickly find that such an alignment is infeasible; the Augustus sculpture simply protrudes too far out from the face region of any Caligula portrait they compare it to.

We can also quickly verify the same for our own scans (see 2.15) and observe all the

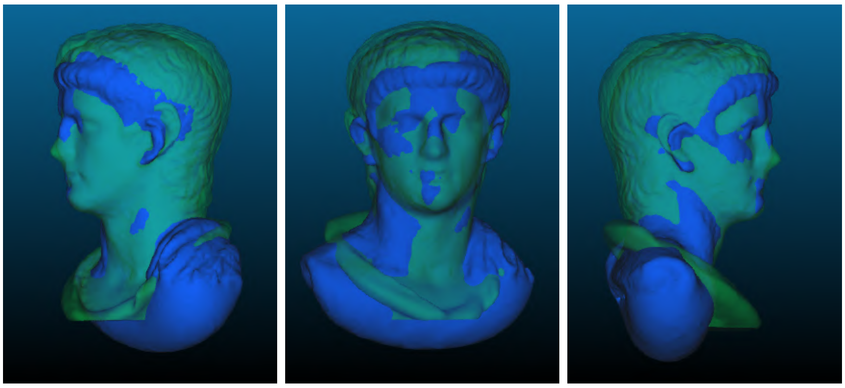


Figure 2.15: Alignment anchored at the scans' facial features

same issues as in Storage did in their project. The back of Caligula's head is much farther out than Claudius', whose hairline and forehead protrude outwards in the face region. However, our Claudius portrait's ears do not match up with Caligula's at all, whereas they coincidentally do so with the Augustus portrait in Storage's case. Therefore, while I agree with Storage's conclusions about reworking methods that would keep most facial features in the same place, a reworking method in which all facial features are recarved is still feasible for our statue pair.

2.3.3 Using Algorithms

In the final part of analyzing our scans, we try to develop a simple algorithm that can determine whether the hair patterns behind both statues' ears match up. We believe this may be possible since we have some evidence suggesting that said area was left untouched during the Claudius portrait's remodeling.

Before we can start developing algorithms that help us analyze the statues, we must first understand more in-depth how digital 3D models, specifically those of Photoscans, work. There are many ways to represent 3D structures digitally, one of the most common ones being the Wavefron OBJ format.

A Wavefront .obj file stores vertices in 3D space by their x,y, and z coordinates, which, on their own, constitute a point cloud [Mat]. Three or more of these vertices may be connected to each other as a polygonal face element and together represent a surface on the 3D object, which is also stored within the file. These faces also define edges that do not need to be directly stored, and together, these three factors are called a mesh. Finally, the orientation (the outward pointing side of the face) of these surfaces is determined by optional vertex normals. Wavefront .obj files may store many more kinds of information than those mentioned, but these are the ones that define a 3D object's shape[23].

All polygonal meshes are defined by their surfaces and do not store any information about volume, so they appear 'hollow' inside. This means that it is possible to create geometry that would be impossible in the real world, such as when an edge has no neighboring face (a 'hole') or when two faces are connected by a shared edge point in opposite directions with their normals. Such geometry is called 'non-manifold' and is important for us to avoid, as manifold objects offer some beneficial properties.

Photogrammetry scans frequently suffer from non-manifold geometry, especially holes, and ours are no exception. Additionally, photoscanning often creates unnecessarily large files, as it includes many points and faces that add little detail to the object's actual geometry. To fix this, we use the software 'CloudCompare' to post-process our 3D scans. CloudCompare offers a plugin called 'PoissonRecon', which recreates an

object as a manifold mesh consisting of roughly equal-sized triangles at a user-specified level of detail. So far, we have only applied this procedure when merging the results of our very first scans into a single mesh, but we now also apply it to the other ones as well. This process fills all holes with interpolated data, reduces filesize, and filters out noise, but that comes at the cost of also removing smaller details.
Filtering holes and making face sizes uniform is advantageous for us, as both holes and large faces are detrimental to calculations that compare vertices on the meshes. They represent gaps in data on the source mesh and lead to imprecise calculations if they are on the reference mesh.

We want to create an algorithm that assigns a certain score between 0 and 1 to each source mesh vertex and then visualize these scores by mapping each score to an RGB value on a green-to-red color scheme. While not part of the official Wavefront OBJ format, applications, including Meshlab [IST], will interpret up to three additional values between 0 and 1 next to a vertex's coordinates as RGB vertex colors and display them accordingly. This means that we do not have to generate custom texture files in order to implement any algorithms and can work directly with the files.
First, we consider an algorithm that considers the distance between two meshes by using the Euclidean distance between each vertex of a source mesh and the closest corresponding vertex on a reference mesh.
The distance between two meshes is an important measure of similarity because two properly aligned and scaled meshes that describe the same structure will have much lower relative distances between each other than if they described different structures. Here, if the reference mesh has a hole where the source mesh has vertices, those vertices will be matched to the border of said hole and treated as if they had a larger distance to the reference mesh than if the hole was filled with interpolated vertices. This applies not just to holes but also to large faces, which are also filtered out during the reconstruction.
Comparing two meshes based on their vertex-to-vertex distance instead of their actual surface-to-surface distance saves on computational expenses, as we do not need to sample points from the source mesh's surface nor calculate plane intersections to get a distance. Before we can do this, we must must first make sure that the meshes are dense enough for us to still extract useful information from our results. Namely, if the maximum error introduced by matching a source mesh vertex with its closest corresponding reference mesh vertex instead of the closest reference mesh surface is significant compared to the average distance between the two meshes, we cannot tell whether a higher distance is valid or stems from an error. On the other hand, if the calculated distance of a point to the reference mesh is much larger than this maximum error, we can conclude safely that that point is indeed roughly that far away from the reference mesh.

An easy way to tell whether source and reference mesh are dense enough is running a pure distance scoring on them. If the source mesh is dense enough, it will generate a smooth color gradient; otherwise, the scores/colors on neighboring vertices may differ drastically. If the reference mesh is too sparse and the errors outweigh the actual distances, the source mesh's gradient will still be smooth but form unexpected circular 'wave' patterns around these sparse points. As seen in 2.16(a), where we compared two dense square-shaped planes, one of which was rotated around the horizontal axis, the distance scoring accurately computes the expected results, as the distance errors due to vertex-to-vertex comparison are insignificant to the actual distances. Figure 2.16(b), shows the same dense plane compared to a sparse plane with vertices only in the 4 corners and looks much more like 2.16(c), in which the two planes are exactly congruent. 2.16(d) also shows the result of comparing two congruent planes, both dense in this case. Because the average plane-to-plane distance is precisely zero, any non-zero vertex-to-vertex error is significant even for very dense meshes, but such patterns are highly unlikely apart from specifically tailored examples and can be easily identified.

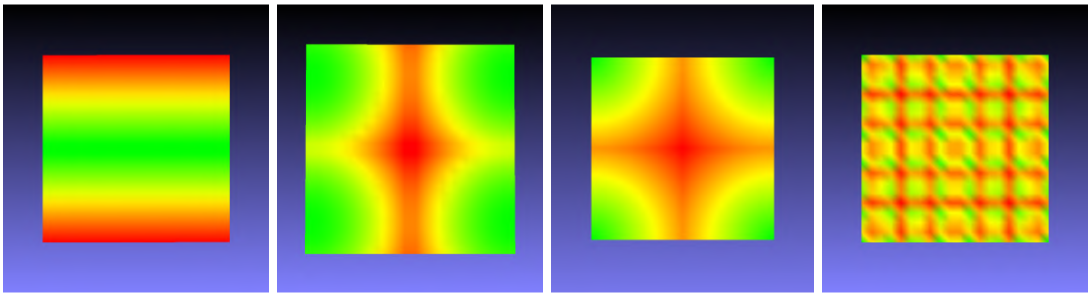


Figure 2.16: Accurate vertex scoring (a) and possible errors (b,c,d)

On dense enough meshes, distance measures also perform well in recognizing global differences, i.e., if you trace a line along the source mesh that slowly moves apart from the reference mesh, its visualization will be a smooth transition from green to red. Because of this, an algorithm based purely on distance cannot recognize local similarities well. If two structures are identical, a distance scoring will not necessarily grade them well but rather give every vertex the same score, and whether that score is good or bad will be determined by the distance between the two. Since we want to recognize local similarities in our meshes, we need a different approach.

One such approach is to segment the mesh into smaller parts, changing what would be local differences in the entire mesh into global differences of these smaller parts. For our case, we compare the hairline behind both Claudius' and Caligula's ears, as we already know that the Claudius statue's face has been reworked extensively and the back of its head has no detail. In the earlier analysis, we already realized that this area possibly had been spared a rework, as the still visible hair detail seems atypical for Claudius and does not bear the same reworking hallmarks of the other parts of the head.

We can further boost the local aspects of the comparison by introducing an additional component. If we consider the immediate surfaces surrounding each source mesh vertex and compare their orientations to the immediate surfaces surrounding the corresponding reference mesh vertex, we can recognize local changes much better. If the surfaces point in the same or similar directions, we know that the immediate neighborhood of both points describes the same shape, and if they face in different directions, we know that the two meshes describe different shapes in the area around the respective vertices. This approach is much more rigid than using the distances between two corresponding vertices and highly sensitive to noise. Of course, the mesh distance still plays a significant role, as it determines which of the surfaces we compare to each other.

We then calculate the scores of each vertex once, accounting only for distance to its closed reference, then only the orientation of the immediate neighborhood around said closed reference, and finally a weighted aggregate of the two. We make sure that the reference segment is larger than the source segment in order to avoid outliers (see 2.17(1)). In 2.17(2), we observe that the distance scoring method creates a smooth color gradient that primarily follows the surface structure and not a concealed pattern. As is expected for such dense meshes that are not perfectly aligned, the average surface-to-surface distance outweighs the maximum vertex-to-vertex distance error. The distance scores in the center and ear sections of this segment are also generally better than those at the boundaries, but that does not necessarily mean that the meshes are more similar in these central areas than in the outer areas. Instead, since the alignment of the statues

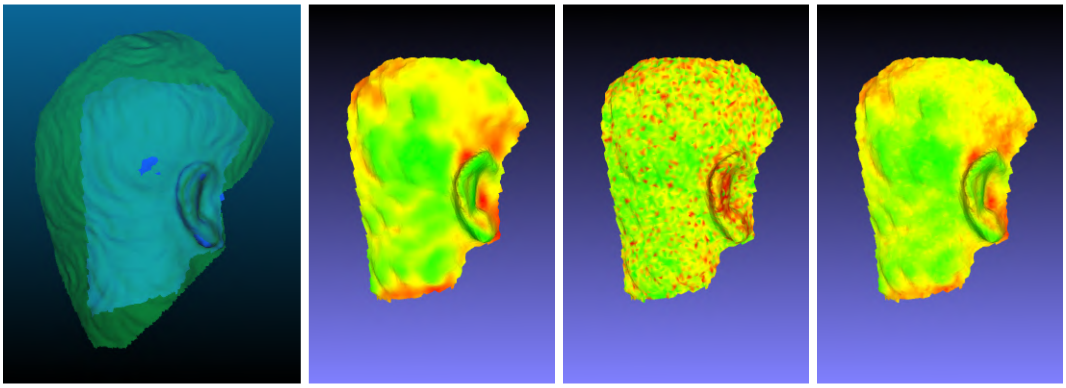


Figure 2.17: Different scoring methods on Claudius' ears

was anchored at the ears, these automatically have the least distance to each other. We have already seen earlier that the Claudius and Caligula portraits have slightly different shaped skulls, which means that, as we move away from the anchor points, the surfaces of the two scans will drift apart as well.

In 2.17(3), we see that the scoring is indeed very rigid, placing very good scores right next to very bad ones. This means that the algorithm is considering too little information at once and responds too quickly, even to minor changes. This erratic pattern also persists after applying smoothing operations to each mesh, suggesting that it is not just caused by noise caused by inaccuracies from the photogrammetric process.

Collecting a larger neightborhood for each vertex would make scoring smoother and potentially more meaningful. Currently, our algorithms perform calculations on vertex lists because these can be read directly out of the 3D objects' files, but collecting bigger neighborhoods in this way becomes grows the runtime exponentially, so we propose using a different data structure for this purpose.

Unfortunately, even if we considered a larger area around both the source and reference point, the increasing global error we see in the distance scoring makes it so the algorithm chooses worse correspondences for points that are further away from the anchors, meaning that the comparison for those points becomes less meaningful, up to almost arbitrary, and this approach most likely cannot capture what we are looking for. A better comparison would have to somehow account for and equalize the different head curvature of both statues while preserving the finer hair details, but that would most likely involve deforming the meshes. At that point, results would become much harder to interpret and verify.

3 Creating an Educational Augmented Reality Application

3.1 Tools used

All software that was used to create the application itself, the markers for tracking the statues, and the sprites that were used inside said application.

- Unity Game Engine[Uni]
- Vuforia Engine Unity Package and Model Target Generator[PTC]
- GNU Image Manipulation Program[GIM]

3.2 Augmented Reality in Cultural Heritage

There are many efforts to employ **Augmented Reality (AR)** in museums and other culturally or historically significant sites. AR lends itself very well to such application because its strength as a medium lies in combining multiple different visual elements into a single scene. We can augment artifacts with relevant information and highlight key details on the object itself. This allows users to interact more directly with the pieces themselves instead of splitting their attention between the historical object and an informative sign.

Moreover, since most artifacts used to look much different back when they were created compared to whatever was preserved of them until now, we can now use AR to recreate their original shapes and colors and even interact with them without further wearing them down or having to create expensive replicas.

3.2.1 Expert Interview

As the de facto commissioners of this project, we asked the Museum of Casts' staff for their perspective on how AR can help enrich the average museum visitor's experience with our culture and history through the unique lens it provides. Thus, we sat down with Dr Nele Schröder-Griebel, one of the museum's conservators, for an expert interview. As a museum conservator, Dr Schröder-Griebel places high importance on science communication, and she thinks that AR can greatly improve the average museum visitor's experience.

One of the main benefits that AR offers to its users, she says, is agency. The user may seek out the parts that interest them freely and at their own pace. This gives them a more integral role during their museum visit, making the user an actor themselves rather than just a listener or spectator. Coupled with the more direct interaction between user and historical object that was already mentioned in the previous section, these factors hopefully create a more memorable visit.

Take for example the Arch of Constantine in Rome, which is embellished with numerous reliefs that come from different timeframes (see 3.1). Some of these reliefs have their replicas on display in the Museum of Casts. Annotating such an object with plaques containing all of this information would quickly become chaotic, which may confuse a potential target audience. With a well designed AR application, however, users could choose by themselves what parts they would like to know more about and how elaborately that application should present the historical context to them. This also makes it easier to strike a balance between different types of museum visitors; those who are less interested in a specific object may choose to view a less extensive version of its notes, while the others can get a more complete picture.

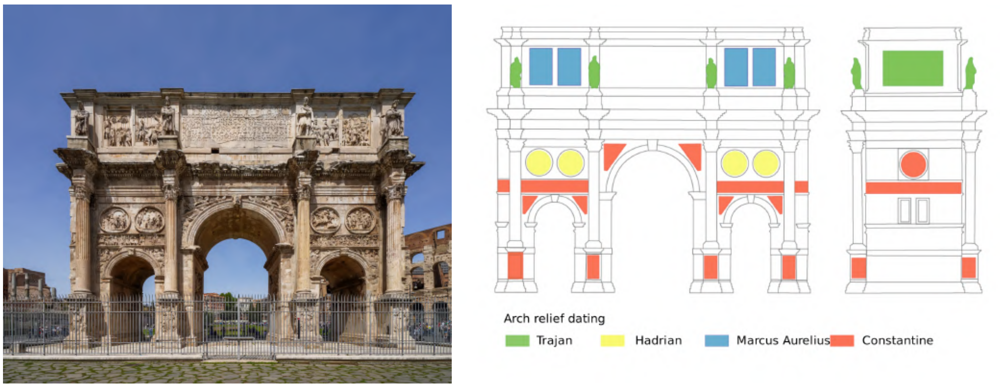


Figure 3.1: Various reliefs on the Arch of Constantine [War99]

In the past, the Museum of Casts in Munich had already offered a smartphone application for its visitors, but the upkeep for multiple operating systems and constantly changing guidelines proved too much expense for its worth, which is why the app was taken down eventually. Since then, the museum has preferred to host its projects on their own website whenever possible. However, with web-based AR technology still in its infancy, such projects would still require a separate phone application.

Making a phone application that is more easily maintainable therefore remains a worth-

while problem to figure out, and we explore some ideas of how to realistically achieve this. Other smaller museums currently also face the same problem, and a union of multiple museums for a shared phone application could greatly reduce the associated workload on each participating party. Each museum could comprise digital guide tours, newsletters, and AR as sub-applications hosted on their own website and make them available for on-demand download in a centralized application. This way, only the centralized application that is shared among multiple museums would need to be kept up to date for the app stores, and with sufficient abstraction in the main app, it should theoretically be possible to keep older sub-applications active after updates through backward compatibility.
It might also be worth having such an app managed by the Ministry for Science and Art, but such an undertaking still seems quite a long way off for now. As of now, the Museum of Casts is already collaborating with a few other museums in order to make 3D scans of their catalog available on their website via a custom 3D-viewer, which is the next point we discussed.

The digitization of historical artifacts also carries with it several benefits. In the past, archaeologists have always relied on comparing and examining artifacts mostly by eye or based on pictures alone. Oftentimes, the relevant artifacts to any specific research project would also be scattered all across the globe, which would in turn lead to high traveling expenses as well as delay progress. Having greater access to these artifacts would speed up that process and perhaps even inspire new discoveries.
The Museum of Casts already serves as a collection of historical objects from many different places, and 3D scans of these objects serve as another important avenue of documentation. Dr Schröder-Griebel recalls that working on 3D scans as opposed to images is a big step up in terms of her research, as some of the more intricate details/structures are very difficult for images to capture. We ourselves noticed some details about the Claudius and Caligula portraits from their scans that we had missed initially thanks to the 3D viewer's different perspective and the ability to directly compare these after aligning them on top of each other.
Additionally, these 3D scans make up the groundwork for algorithmic analysis such as the one we discuss in this thesis and possibly even automated approaches that could eventually save archaeologists much time.
Since there are so many artifacts to digitize, a big focus also lies on the scanning process itself; it should neither be too difficult to perform nor too time consuming whilst also not relying on very expensive hardware such as laser scanning. Accounting for all of these factors and still producing high quality results is also quite a challenge that the museum must face as it extends its online catalog. They have been working with university students from diverse fields of study for quite some time now, and none

of the options come without. So far, they have been using a tablet with Polycam [24], but that captured the finer geometry of its targets not as well as photogrammetry with images when we tried it ourselves initially.

Besides for direct research purposes, having 3D scans of historical artifacts helps archaeologists communicate their findings better. We already discussed that images do not capture certain details as well as scans can, and the same is obviously also true for conveying such details to another party.

The reason that the Museum of Casts in Munich considered AR for this project specifically is twofold. For the visitors, Dr Schröder-Griebel hopes to present them with interesting insights into the archaeological process and research methods by showing them the oddities of the Claudius and Caligula portrait. While they do seem rather inconspicuous at first glance, we have seen how they tie into the concept of damnatio memoriae and the economic incentives of ancient times.

If the user can follow along with this process and retrace the steps from noticing a peculiar detail over forming a hypothesis to explain said detail, testing this hypothesis via comparison, and finally reaching a conclusion all by themselves, they may experience it as an 'aha moment'. The knowledge gained from such an experience may stick with them or even inspire them into further interest in history and archaeology.

Secondly, we also discussed this project from a scientific standpoint. Dr Schröder-Griebel wished to find out how feasible it is to compare these kinds of different yet in some ways similar statues and explore some of the challenges that come up while working on this problem statement. The museum's staff had obviously noticed these portraits for their head shape beforehand, but without 3D scans of them, it was hard to tell how they would look aligned on top of each other and how good of a fit it would be. Therefore, the application might also be useful for any researchers who want to explore similar ideas, and it could be beneficial if we included more information about our analysis within it.

The limited information from ancient times that is still available to us nowadays makes it difficult to produce results with absolute confidence, as, usually, there are many assumptions involved and almost always another possible explanation. This makes the fact we know so much about these times all the more impressive, and we should not be discouraged if a project unearths more new questions than it answers them, as that is simply part of the scientific process and especially true for archaeology.

3.3 An Augmented Reality Application for the Claudius Portrait

In the course of this thesis, we have discussed many of the details about the Claudius portrait that differentiate it from an ordinary portrait. Part of this work was to create an AR application that allows its user to examine the portrait by themselves.

After analyzing the scans, we must now apply further post-processing steps before we use them in our application. Recall that, in chapter 2, we applied the 'PoissonRecon' feature of 'Cloud-Compare' to fill in the holes of these scans and make them less demanding on memory, both qualities we want to preserve for for the 3D objects we want to use in the application. However, this process does not transfer the textures onto its results.

Digital Textures were unimportant for the algorithm or analysis parts of this project, but they are necessary if our application is to provide a proper user experience. Therefore, we must now transfer the texturing from the original, faulty 3D scans to their retouched versions.

One way to do this is with the software 'Blender' [Ble], where we can transfer textures onto another model by 'baking' them.

On digital Models, textures are essentially just 2D images that have been given some sort of mapping to the model - a so-called 'UV-map' [Polb]. We have to create one such map for our untextured models before we can bake over the textures from the textured scans, as the software would otherwise have nowhere to place the results from its calculations. We cannot simply reuse the UV-maps from the original, already textured scans because their geometry is entirely different, which also applies to their UV-maps.

Therefore, the first step for us is to UV-unwrap the model in Blender. It offers an automated way to do so via the 'Smart UV Project' option. This leaves us with a texture map on our rework scan, and we can start 'baking' the textures over from the original scan. The software does this by casting rays from the surface of the textured model towards the untextured one and then calculating the surface detail at the point where the ray intersects the latter. This surface is then saved to the UV-map of the untextured model [Pola].

In our case, unfortunately, the 'Smart UV Project' option in Blender had created a faulty UV-map for the reworked scan that does not match every triangle on the mesh to a part in the texture and, in some cases, multiple different triangles to the exact same location in the texture. These mismatched singular triangles are scattered seemingly uniformly across the entire scan, and because of the incorrect texture details for them, they simply show up as tiny white spots on the scan (see 3.2).

While looking for ways to fix the pixel artifacts, we found a different way to apply texturing to our made-over scans directly within Mehsroom. The Meshroom texturing

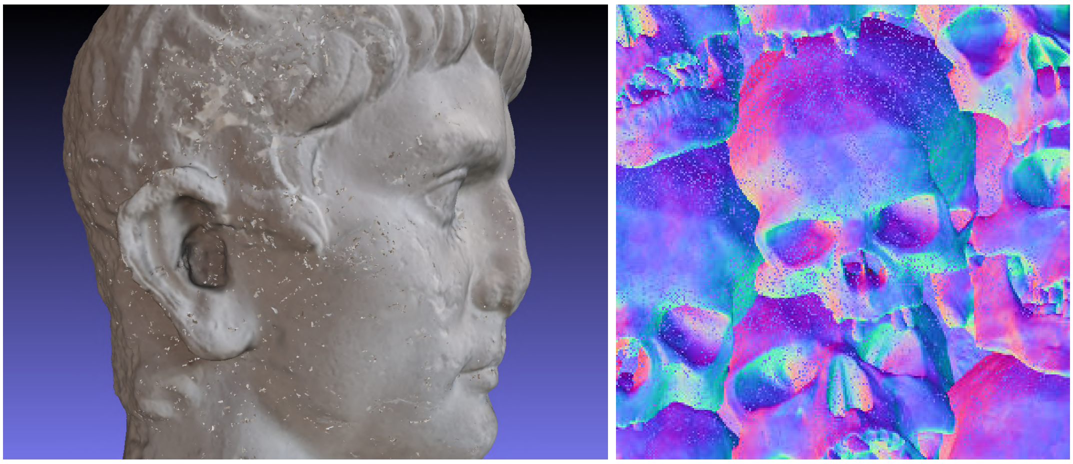


Figure 3.2: Pixel Artifacts in our scans and showcase example (image: Eric Chadwick)

node in the pipeline requires the processed image data, which we already have from initially creating the scans, and an input mesh. If we inject the post-processed scans into the texturing node instead of the original scans, Meshroom can take care of the rest for us.

Still, we must be aware of two things. First off, the injected mesh must be in the exact position of the original scan's mesh, but we thankfully already know how to align two meshes using CloudCompare [Ope]. The other problem only appears if the original scan has larger holes in its surface. In these cases, the textures in the affected areas are likely faulty, which manifests in off-colored specks (see 3.3).

It should be possible to combine the results of two different texturing outputs into a single one to compensate for the specks, but that would probably require the user

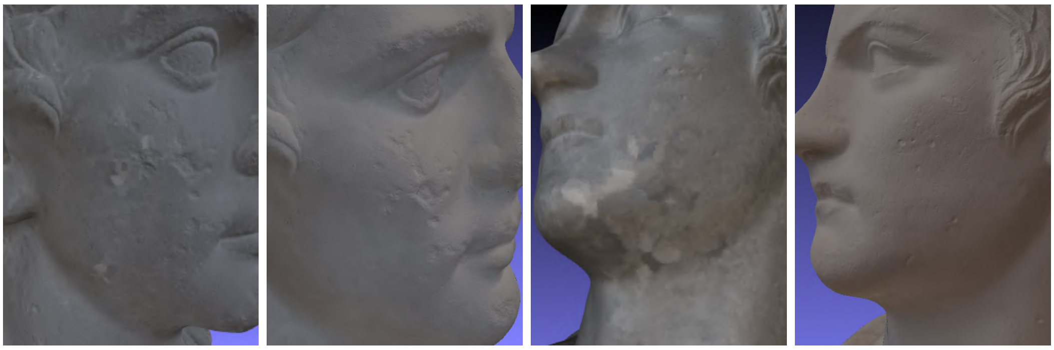
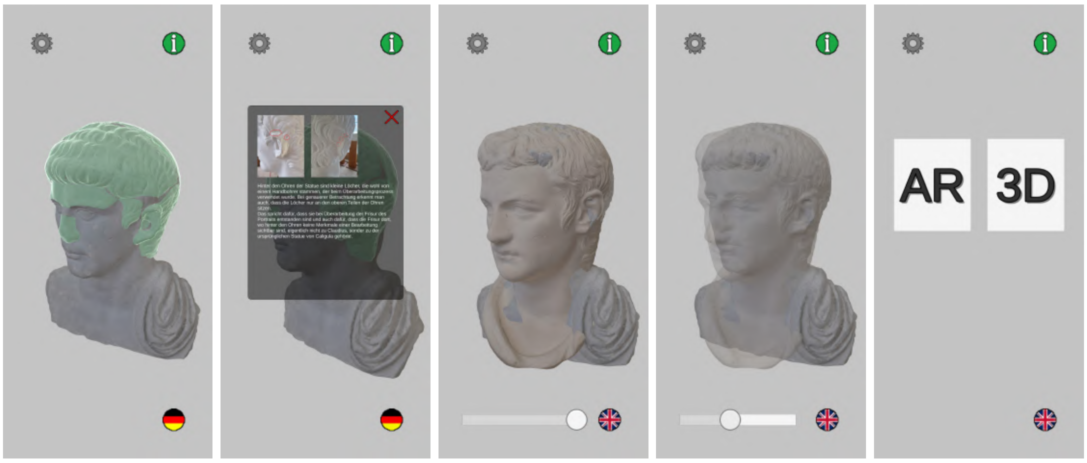


Figure 3.3: Faulty and proper textures on the edited scans

to equip the input mesh with a UV-map beforehand, and the transitions between the texture from the different outputs will also need smoothing to account for differences in lighting. However, this was not necessary for our scans.
For the tracking, we use the Vuforia Model Target Generator, which lets us convert ordinary 3D meshes into trackable objects. Since we have multiple meshes of each statue from the different post-processing stages we performed, we create several distinct model targets, both with and without textures, and test them on the Claudius portrait at the museum. All of these targets worked very well at the time, so we ultimately settled on the model target that was made from the final version of the scan, which we also use for the non-AR part of the application as stand-in for the real statue.

With appropriate 3D representations of the statues and a way to track them in AR, we now build an android application using the 'Unity Game Engine' [Uni]. The user may choose to view the statues in one of two different modes. The first mode concerns only the Claudius statue. In this mode, the statue is augmented in a way that highlights distinct features, showcasing the evidence we found for the statue's rework and some of the damages it received since ancient times. Tapping on these areas then displays information about the highlighted parts.
In the other mode, the user may display a see-through version of the Caligula portrait augmented on top of the statue and see for themselves how an ordinary statue would have to change in order to become a different face. This disables the highlights on the Claudius statue to avoid cluttering. They may also change the opacity of the augmented statue freely with a slider and the language of the application between English and German(see 3.4).



Hinter den Ohren der Statue sind kleine Löcher, die wohl von einem Handbohrer stammen, der beim Überarbeitungsprozess verwendet wurde. Bei genauerer Betrachtung erkennt man auch, dass die Löcher nur an den oberen Teilen der Ohren sitzen.
Das spricht dafür, dass sie bei Überarbeitung der Frisur des Portraits entstanden sind und auch dafür, dass die Frisur dort, wo hinter den Ohren keine Merkmale einer Bearbeitung sichtbar sind, eigentlich nicht zu Claudius, sondern zu der ursprünglichen Statue von Caligula gehörte.

Figure 3.4: The application in both modes and main menu

The see-through statue coincides with the (subjectively) best alignment of the two portraits we obtained during digital analysis, and this application will hopefully allow users to understand the thought process behind said analysis.
Should the Claudius portrait be unavailable, we also made it possible to view the application in a 3D scene instead of AR. Since we already obtained scans of both portraits, the real-life Claudius statue is replaced with its textured and post-processed mesh representation.
We only include our own images for the information panels and UI sprites to avoid any potential copyright issues, but we would still need to acquire a license for the Vuforia Engine in order to release this application publicly.

4 Related Work and Future Work

4.1 Proxy Painting: Digital Colorization of Real-world Objects

Contrary to the common belief during the Renaissance that ancient marble sculptures were purely white, we now have ample evidence that these sculptures used to be painted with various pigments and other colorful adornments [Abb07]. There are multiple projects that seek to revive the ancient polychromy of now white marble statues with the help of Augmented Reality.
In their article about proxy painting, Lange et al. use projection mapping to display non-permanent and dynamic color on normally uncolored statues [Lan+20]. Projection mapping directly affects real-world objects by projecting details onto them using a beamer or something similar. Unless the projection target is a flat surface or has very simple geometry, doing so accurately requires a digital representation of that target and a 3D-capable video setup with multiple cameras or a depth camera. This makes it possible to 'map' the state of the digital representation onto the target and augment its surface with patterns and colors.
Contrary to other AR methods using screens such as handheld or head-mounted displays, projection mapping can be observed by multiple people at once. Of course, there must be a minimum distance between visitors and the projection-mapped exhibit since the beamer must keep a clear line of sight to the augmentation target, limiting user interaction and possibly making fine details more difficult to spot. This is where Lange et al.'s proxy painting method reintroduces the interactive parts of AR by allowing users to directly interact with a smaller, 3D-printed clone of the statue. This clone is also scanned and lets the user simulate painting the actual statue with an Augmented Reality paintbrush almost in real time. Their projection mapping setup can also be used to capture pre-painted painted copies of the statue and transfer their colors onto the real statue, also allowing for more accurate coloring methods than painting with the AR brush.

For our own project, while projection would not be able to recreate the missing facial structure of Caligula on Claudius' reworked portrait, we may be able to use this approach to AR in order to recreate the original hairstyle of Caligula at the back of its head. Since the back of Claudius' head is blank, it should serve well as a projection

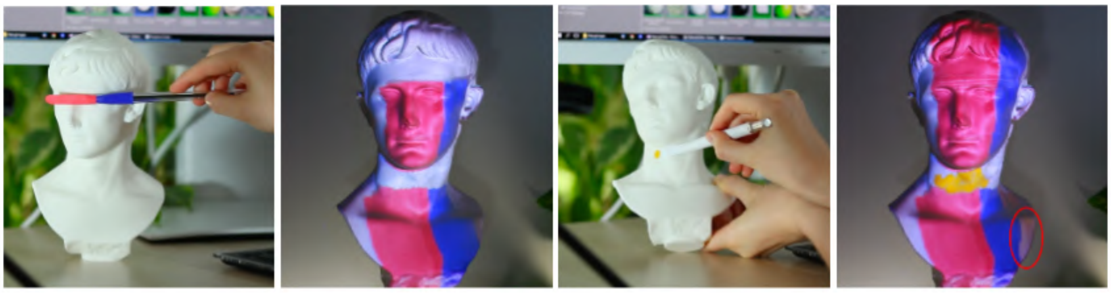


Figure 4.1: Proxy painting example with AR brush [Kle+20]

surface. Such an approach may be able to show where the leftover hairstyle of the original statue blends into the projection and where the new hairline clashes with the old details.
Alternatively, one could develop an application that allows the users to examine a 3D printed copy of the portrait via touch and reacts whenever they reach an area of interest.

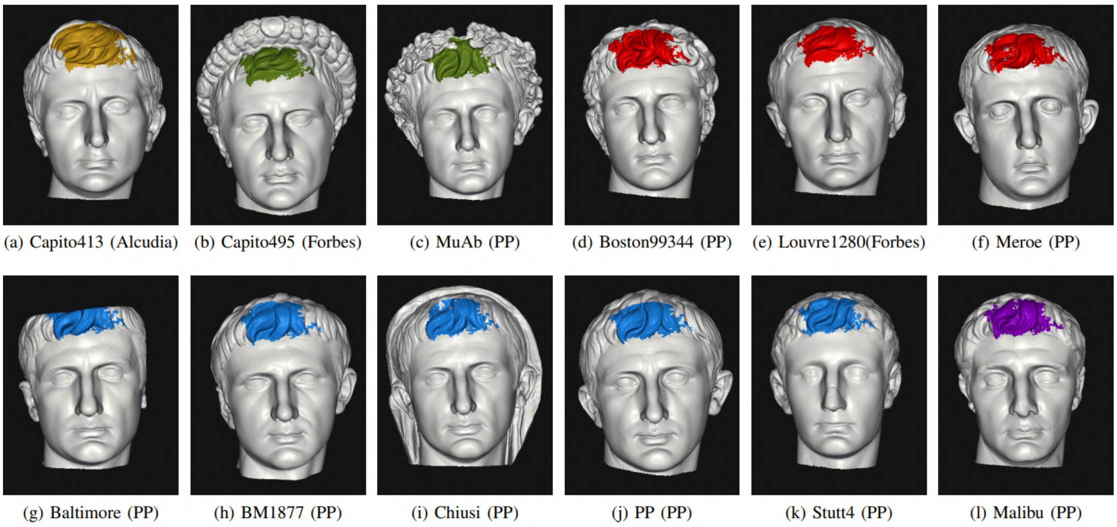
4.2 Projective Augmented Reality in a Museum: Development and Evaluation of an Interactive Application

In 2020, Plecher et al. presented another way of using projective AR with the goal of enhancing a museum experience [Ple+20]. Here, the users can examine the museum exhibit, a cast of the Tombstone of Xanthippos, on a tablet and view different kinds of information by tapping on the corresponding part in the application. This part is then highlighted by the projection mapping setup on the real exhibit as well. The application also allows the users to colorize the exhibit with a brush tool on the tablet, which is then also transferred by the projection mapping setup. After they have finished, their drawing is then saved to a gallery to be viewed later, and the saved drawings are projected onto the exhibit while the application is idle in order to catch attention.
Nowadays, this approach of combining knowledge transfer with gamified elements is being adopted by more museums with hopes of providing their visitors with a lasting experience. In Plecher et al.'s case, they could show that, under equalized conditions, users of the application showed a better level of knowledge retention than visitors who were only given the same information on a sheet of paper.

4.3 Future work in algorithms for comparing statues

In section two, we describe how difficult it is to find a proper alignment for two heads that look so different from each other. With as many assumptions as we have to make, our conclusions become less clear. For an algorithmic approach, it should be better to start with statues that are more similar to each other.

In 2013, Lu et al. examined twelve sculptures of the emperor Augustus on laser scans also obtained at the Museum of Casts in Munich [Lu+13] for a very similar project to ours. They tried and tested an algorithmic approach to categorizing these into different known 'types' as explained in 1.2.
Their laser scans were much more accurate than our photogrammetry scans, and they had no need for post-processing the scan data's structural inconsistencies such as holes before being able to use them. Their algorithmic approach is much more sophisticated, and they are able to deform and make their scans uniform via non-rigid transformation onto a shared template. They argue that type categorization depends mainly on the frontal regions of statues and consider only the face and hair above the forehead for that reason. They then segment the scans into multiple regions and categorize the twelve statues into different types using a local shape comparison method that compares each segmented region of one statue to the corresponding region of another. This scheme gives the entire segment a single score, and the segments with the most similar scores are grouped together into a type category (see 4.2). Their results show that it is possible


(a) Capito413 (Alcudia) (b) Capito495 (Forbes) (c) MuAb (PP) (d) Boston99344 (PP) (e) Louvre1280(Forbes) (f) Meroe (PP)

(g) Baltimore (PP) (h) BM1877 (PP) (i) Chiusi (PP) (j) PP (PP) (k) Stutt4 (PP) (l) Malibu (PP)

Figure 4.2: Type categorization of Augustus statues in [Lu+13]

to reproduce the archaeological categorization algorithmically by looking at the state's facial regions.

As we have already seen, alignment is a big issue for our project because it is very difficult to obtain accurate correspondences on the two different scans, and we cannot rely on their facial features nor their hairlines due to the rework. Non-rigid transformation besides changing the global scale is not an option for the same reason, at least not for any of the facial regions.
We also see from the Augustus portraits that statues of the same type retain quite some variation even in their main features. It is only reasonable to assume that this is also the case for the less significant areas and that these might show even more of a difference between statues of the same type. Since less significant areas are, by the same logic, also more likely among those overlooked during a rework, we have to deal with another degree of uncertainty. For example, a good match between two statues in an insignificant area might be more indicative of the workshop or artisan they come from than the type and person they represent, whereas a bad or impossible match might not necessarily yield sufficient evidence against the rework hypothesis.
In the future, if we continue the work on algorithmic statue comparisons, we propose starting with looking for similarities in statues that were not reworked. If we take a similar approach to Lu et al. and compare multiple statues of two different emperors to each other, we can find out if it is possible for an algorithm to tell them apart based on their insignificant areas.
If we can find an algorithm that confidently assigns statues to the correct emperor based on these areas, we may be able to apply that algorithm to a reworked statue. A confident placement amongst either of the two emperor's statues should be sufficient evidence for deciding whether that particular area was or was not modified accordingly during its rework, while an unsure placement may suggest that the statue's original look was of another emperor altogether.

4.4 Methods for accurate and quick photoscanning using affordable hardware

In the Expert interview, we mentioned that many museums have great interest in the digitization of historical artifacts, for visitor experience and scientific purposes alike. For our project, we used a simple phone camera to take many images of our targets from all degrees and put them into the free photogrammetry software 'Meshroom' [Ali]. As of April 2024, RealityCapture [Gam24], another photogrammetry software, is also available for free for smaller projects. The industry standard, Agisoft's Metashape

[Agi24], is still paid software but is commonly said to produce the most consistent results. What these three programs have in common is that they offer a highly customizable photogrammetry pipeline, allowing experienced users to make the most out of their images.
As we have seen, photogrammetry is somewhat tricky to perform, and even with some experience, scans may always go wrong, requiring a retake. It also struggles with smooth surfaces as it does not include any distance measurements and can therefore not guarantee the accuracy of its output. We observed these inaccuracies ourselves in this project and had to put quite some effort into 'cleaning' up our scans.
On the other hand, we could rely on applications such as the already mentioned Polycam's photogrammetry option [24] to produce 3D scans, which is much quicker and easier. These usually capture the rough geometry of objects quite well and generate very realistic-looking scans, but many of the fine structural details come from the textures and not the structure of the mesh itself. This likely comes from a high degree of automated smoothing operations performed on the original mesh data. As we are already aware, the computational cost of photogrammetry pipelines is quite high, and these applications do not compute them locally on the phone but upload them onto their servers instead. This usually also means that the in- and outputs are compressed, and the pipeline is not as extensive as it could be [Kat23]. Such applications should work fine if we just want to view the artifacts digitally, but we should aim for the maximum possible structural accuracy for any algorithmic analyses.

A relatively recent development in the mobile market is Apple's 'LiDAR' scanners, which are available on some of their newer models. These scanners can very accurately measure the distance between the camera and surfaces, and combining them with photogrammetry images should be able to create highly detailed, textured, and precise 3D representations of real-world objects.
LiDAR data does not include textures and is separate from image data, so in order to make use of it in applications that do not natively support and handle both types of data at once we need to align LiDAR and photogrammetry point clouds from two separate scans first.
Polycam supports LiDAR scanning as well, but it processes the result locally and in real-time on the phone or tablet we use for scanning as opposed to its photogrammetry options. The data being processed locally also means that this feature is available for free on all the devices that support LiDAR, as it comes with no additional costs for the application's publishers. Of course, this option is great for beginners, as they obtain live feedback for their scans. However, for our purposes it means that the benefits from LiDAR go mostly into performance and not into improved accuracy. This could still be a great option to quickly scan many artifacts in a row, but the meshes might not be

detailed enough for use in algorithms.

There is also the option of using structured light optical scanners for our captures. This method may be particularly interesting to the museum of casts, as structured light performs very well on uncolored objects such as casts. Photogrammetry, in contrast, will have a harder time finding feature points on these casts. Similar to LiDAR, structured light scans are quite fast as well while still offering high accuracy and level of detail and only struggle with reflective and dark surfaces, neither of which should be an issue when trying to scan casts. Just like the LiDAR options, they can also capture an object's texture and generate a preview in real time, but the processing is done on a connected computer instead of the device itself, so there is more computing power available for more accurate scans in the same or similar timeframe.
Structured light scanners start in a similar price range to the LiDAR-capable Apple hardware, so they could certainly be considered as an alternative to such devices if their main purpose will be creating scans of statues. Of course, they also come with their own drawbacks. Unlike with photogrammetry where we can just reuse the images for multiple applications, structured light scanners are usually bound to a specifically tailored software of the manufacturer, making it less flexible. As already mentioned, they additionally struggle with dark colors and especially the cheaper versions have a hard ceiling in the resolution they can achieve.

Finally, we should also mention laser scanners. These are very expensive, but they do come with accuracy guarantees in the $\sim 40\ \mu$m range by the manufacturer [21]. Now, while some of them do support the creation of textured scans via some other method, laser scanning itself captures only structure and not colors directly. Therefore, before using their scans in a digital library/guide or an application, we may first need to obtain these details via a different method. Still, the resolution and accuracy of laser scanning stays unmatched, and especially the algorithmic approaches for analysis could be greatly improved with this data.

5 Summary and Outlook

During this project, we managed to create high enough quality photoscans of the statues that this thesis concerns to track them with Augmented Reality. After digital post-processing, these scans look very close to their real-life counterparts, which allows us to accurately highlight even the smaller details of the statue in an AR android application.

Our digital analysis also produced some specific evidence for the statue's alteration that was much less apparent whilst viewing it in person. Furthermore, we proposed and reasoned for a reworking method that may have produced this Claudius portrait from a Caligula portrait that originally had a similar shape to the one we scanned.
Still, much of our analysis hinges on assumptions about the statues' ears, which are generally not the most consistent parts between different statues of the same type, and the damages to these may also affect the quality of our alignment and scale finding as well.
In order to increase the confidence of our results, we would require more in-depth knowledge and more statues for comparison.

During analysis with algorithms, however, we could not produce proper results. This is due to three factors. Firstly, without any thoroughly tested ways of verifying our results, the only way for us to verify these was by eye, meaning that they would, at most, only be able to confirm the information already available to us. Secondly, due to the many factors differentiating the two statues we wanted to compare and the highly local nature of our algorithms, this analysis quickly falls short. Finally, because our scans were obtained using photogrammetry and not with a laser scanner, we had to perform many post-processing steps in order to obtain workable 3D representations, potentially introducing many small inaccuracies and losing the finest of details throughout.

Finally, we managed to create an Android application that allows its users to scan and augment the reworked Claudius statue with both the information we were provided by the Museum of Casts and what we found out ourselves by analyzing the photoscans. This application should allow for a more direct and gamified interactive experience between visitors and the exhibition pieces.

By now, AR technology has become an established technology, but lots of resources are still being poured into its further development. Especially for 3D objects, both tracking itself and even more so the creation of their markers are very expensive computationally. This unfortunately means that there are no tried and true methods for working 3D object tracking in browsers available just yet, and we could not make the AR part of our application usable without installing an application onto any device beforehand. However, AR development platforms are becoming accessible more widely each year, and with its high marketability in e-commerce, it seems inevitable that AR for the web will eventually both provide all current state of the art features and be available for more than just big enterprises.

In the same vein as with AR, the inclusion of machine learning has also made photogrammetry and other 3D scanning methods more accessible and efficient, and many big companies are interested in improving these processes further. For most of the projects like this one, revisiting them in a few years' time with new and improved technology could lead to considerable improvements.

6 Glossary

Augmented Reality (AR) describes the practice of placing virtual objects into real world environments, specifically in order to simulate an interaction between those two. Some of the most prominent examples would be face filters.
AR falls on the ’Mixed Reality Spectrum’, which describes how much of a certain experience relies on the real world versus a virtual environment (see 6.1).

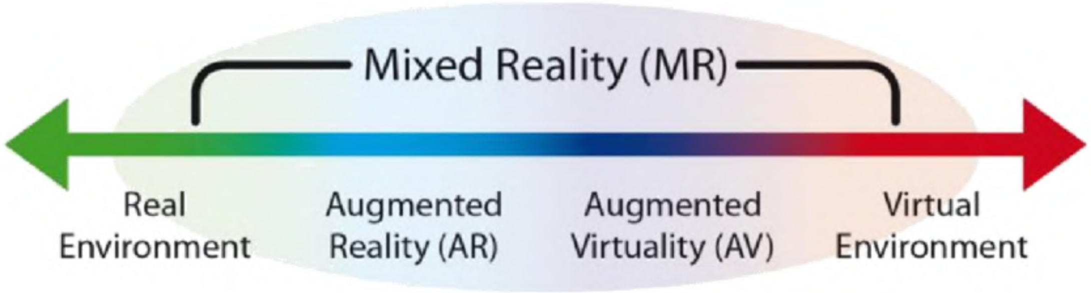


Figure 6.1: The Mixed Reality Spectrum ([Ash+23])

On this spectrum, AR is commonly used to refer to experiences that rely mostly on real environments, with only few virtual ’augmented’ parts.

Photogrammetry is used when trying to extract 3D information from photographs. Matching overlapping areas in these photographs against each other makes it possible to estimate the corresponding camera positions at the time the photos were taken. Finally, the camera and pixel data from the overlapping pictures are used to extract unique feature points and use these to create different kinds of 3D objects[Autb].

Structure from Motion (SfM) is the process of estimating the 3-D structure of a scene from a set of 2-D images. Here, ’Motion’ abstractly refers to the different camera positions and how they stand in relation to each other, which, for an SfM made from a video or many ordered images around an object from a single camera, coincides with the motion that said camera made around the object in the real world.
It is a specific kind of Photogrammetry that produces a cluster of colored points in 3D

space, a so called **Point Cloud**.
By treating each point as a sphere with some user-defined radius, dense Point Clouds can be made to look like solid 3D objects such as this reconstruction of a cultural site destroyed in the Syrian civil war (see 6.2).

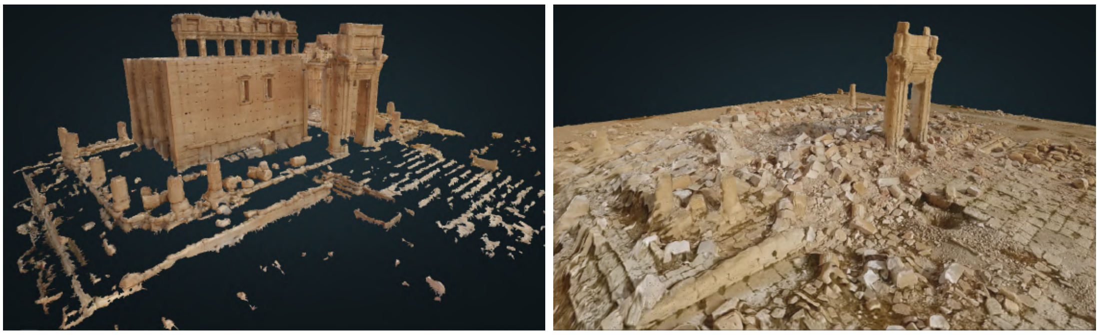


Figure 6.2: Point Clouds of the Temple of Bel before and after its destruction, Palmyra Syria [Cal]

They can also be refined into more complicated 3D objects such as polygonal **meshes** with surface information. In addition to storing points as vertices, meshes also include edges and faces connecting these vertices to each other. Polygonal meshes commonly used in 3D applications because of their efficiency and ease of use [Gup23].

List of Figures

Bibliography

[21] *Accuracy of EinScan HX Handheld 3D Scanner*. 2021. URL: https://www.einscan.com/applications/accuracy-of-einscan-hx-handheld-3d-scanner/.

[23] *Wavefront .obj file*. 2023. URL: https://en.wikipedia.org/wiki/Wavefront_.obj_file.

[24] *Polycam*. 2024. URL: https://poly.cam/.

[Abb07] M. B. Abbe. “Polychromy of Roman Marble Sculpture.” In: (2007).

[Agi24] Agisoft. *Metashape*. 2024. URL: https://www.agisoft.com/.

[Ali] Alicevision-Association. *Meshroom*. URL: https://alicevision.org/#meshroom.

[Ang14] A. Anguissola. *Classics Greek Originals and Roman Copies*. 2014. URL: https://www.oxfordbibliographies.com/display/document/obo-9780195389661/obo-9780195389661-0213.xml.

[Arca] Archaeologisches-Institut-Goettingen. *Augustus*. URL: https://viamus.uni-goettingen.de/fr/e_/uni/e/02/01/index_html.

[Arcb] Archaeologisches-Institut-Goettingen. *Die Typologie römischer Kaiserbildnisse*. URL: https://viamus.uni-goettingen.de/fr/e/uni/d/03/01.

[Arcc] Archaeologisches-Institut-Goettingen. *Gaius "Caligula" (37 - 41 n. Chr.)* URL: https://viamus.uni-goettingen.de/fr/e/uni/e/02/03.

[Arcd] Archaeologisches-Institut-Goettingen. *Memoria damnata - das Löschen des Andenkens einer Person*. URL: https://viamus.uni-goettingen.de/fr/e/uni/d/03/05.

[Ash+23] E. Ashgan, N. Moubarki, M. Saif, and A.-M. El-Shorbagy. *Mixed-Reality-continuum-spectrum*. 2023. URL: https://www.researchgate.net/figure/Mixed-Reality-continuum-spectrum_fig4_366879982.

[Auta] Autodesk-Inc. *Meshmixer*. URL: https://meshmixer.com/.

[Autb] Autodesk-Inc. *Photogrammetry*. URL: https://www.autodesk.com/solutions/photogrammetry-software.

[Ble] Blender-Foundation. *Blender*. URL: https://www.blender.org/.

[Cal] U. of California San Diego Library. *Temple of Bel, featured on https://www.capturingreality.com/temple-of-bel-in-palmyra-as-vr*. URL: https://pointcloud.ucsd.edu/Palmyra_Temple_of_Bel_Viewer.html.

[Gam24] E. Games. *RealityCapture*. 2024. URL: https://www.capturingreality.com/realitycapture-14.

[GIM] GIMP-Team. *GIMP*. URL: gimp.org.

[Gup23] K. Gupta. *Polygon Mesh in Computer Graphics*. 2023. URL: https://www.geeksforgeeks.org/polygon-mesh-in-computer-graphics/.

[IST] ISTI-CNR. *Meshlab*. URL: https://www.meshlab.net/.

[Kar19] P. S. Karol Kłodziński. "Domitian's damnatio - A Critical Case Analysis." In: (2019).

[Kat23] D. Katsafouros. *How good is Polycam?* 2023. URL: https://www.youtube.com/watch?v=IuaWCfNpTlg.

[Kle+20] V. Klein, P. Kurth, B. Keinert, M. Boss, M. Stamminger, and F. Bauer. In: *Journal on Computing and Cultural Heritage* 13 (May 2020). DOI: 10.1145/3377145.

[Lan+20] V. Lange, P. Kurth, B. Keinert, M. Boss, M. Stamminger, and F. Bauer. *Proxy Painting: Digital Colorization of Real-world Objects*. 2020.

[Len20] J. Lendering. *Damnatio memoriae*. 2020. URL: https://www.livius.org/articles/concept/damnatio-memoriae/.

[Lu+13] M. Lu, Y. Zhang, B. Zheng, T. Masuda, S. Ono, T. Oishi, K. Sengoku-Haga, and K. Ikeuchi. "Portrait sculptures of Augustus: Categorization via local shape comparison." In: Oct. 2013, pp. 661–664. ISBN: 978-1-4799-3170-5. DOI: 10.1109/DigitalHeritage.2013.6743812.

[Mat] MathWorks. *Point Cloud Processing*. URL: https://www.mathworks.com/help/vision/point-cloud-processing.html.

[Nas] Nashville-Film-Institute. *What is Depth of Field?* URL: https://www.nfi.edu/depth-of-field/.

[Ope] Open-Source. *Cloud Compare*. URL: https://www.danielgm.net/cc/.

[Ple+20] D. A. Plecher, A. Ulschmid, T. Kaiser, and G. Klinker. "Projective Augmented Reality in a Museum: Development and Evaluation of an Interactive Application." In: *ICAT-EGVE 2020 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*. Ed. by F. Argelaguet, R. McMahan, and M. Sugimoto. The Eurographics Association, 2020. ISBN: 978-3-03868-111-3. DOI: 10.2312/egve.20201258.

[Pola] Polycount. *Texture Baking*. URL: `http://wiki.polycount.com/wiki/Texture_Baking`.

[Polb] Polycount. *Texture Coordinates*. URL: `http://wiki.polycount.com/wiki/Texture_Coordinates`.

[PTC] PTC. *Vuforia Engine*. URL: `https://developer.vuforia.com/#`.

[Sto09] W. Storage. *Augustus Once And For All*. 2009. URL: `https://www.rome101.com/Portraiture/Augustus/IsItRecarved/#Varner`.

[Uni] Unity-Technologies. *Unity*. URL: `https://unity.com/download`.

[War99] Ward-Perkins. *Arch of Constantine datation*. 1999. URL: `https://commons.wikimedia.org/wiki/File:Constantine_arch_datation.svg`.