TU WIEN Informatics

Anatomical Entertainer

Physikalisierungen im Gebiet der medizinischen Visualisierung

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Marwin Schindler
Matrikelnummer 01627754

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Dr Renata Georgia Raidou

Wien, 24. April 2020

Marwin Schindler | Renata Georgia Raidou

TU WIEN Informatics

Anatomical Entertainer

Physical Visualization in a Medical Context

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Marwin Schindler
Registration Number 01627754

to the Faculty of Informatics

at the TU Wien

Advisor: Dr Renata Georgia Raidou

Vienna, 24th April, 2020

Marwin Schindler   Renata Georgia Raidou

Erklärung zur Verfassung der Arbeit

Marwin Schindler

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 24. April 2020

Marwin Schindler

Kurzfassung

Visualisierungen sind essenziel, um der allgemeinen Öffentlichkeit einen Einblick in die menschliche Anatomie zu ermöglichen. Traditionelle Visualisierungsmethoden konzentrieren sich auf 2D und 3D Informationsrepräsentationen, entweder in digitaler oder gedruckter Form. Physikalische Visualisierung ist ein Teilbereich der traditionellen Visualisierung, hier werden die Daten als physikalische Objekte dargestellt. Es wurde gezeigt, dass physikalische Visualisierungen, bei den interagierenden Personen zu einem besseren Verständnis der Informationen führen. Viele der Verfahren, um physikalische Visualisierungen zu erstellen, sind jedoch nicht zugänglich für die allgemeine Öffentlichkeit. In dieser Arbeit wollen wir ein Verfahren präsentieren, um den Erstellungsprozess von physikalischen Visualisierungen aus Papier zu vereinfachen. Das vorgestellte Verfahren kann dafür genutzt werden zwei verschiedene Arten von anatomischen Visualisierungen zu erstellen. Einerseits können 2D Visualisierungen erstellt werden, welche zusätzlich mit analogen Farbfiltern untersucht werden können. Um die Darstellung von mehreren überlappenden Schichten der Anatomie zu ermöglichen, wird eine spezielle Methode der Farbmischung verwendet. Diese ermöglicht es, mit der Hilfe von Farbfiltern, trotz Verdeckung, die verschiedenen anatomischen Strukturen zu untersuchen. Andererseits können 3D Papierfiguren erstellt werden, welche ebenfalls mit Farbfiltern untersucht werden können. Die anatomischen Modelle werden dafür entfaltet und können gedruckt und dann zusammengebaut werden. Die fertigen Papierfiguren sind sehr einfach zu erstellen, sowie zu verteilen und können als Lernspielzeug oder zur Unterhaltung genutzt werden. Wir präsentieren einige 2D und 3D Beispiele des Verfahrens anhand von Modellen für den Anatomieunterricht.

Abstract

Visualizations are essential for anatomical education of the general public. Traditional visualization methods focus on 2D and 3D information representations, either digital or printed, but visualizations also have a physical form. Physical visualization is a subdomain of the traditional visualization domain, where the data is represented by means of a physical object. Physical visualizations have been reported to lead to greater information insights for the interacting user, but a lot of the fabrication methods to create physical visualizations of the anatomy are not accessible for the general public. In this thesis, we present a workflow to ease the process of creating physical visualizations, made out of paper. The proposed workflow can be used to create two different types of anatomical visualizations. First, we generate 2D visualizations, examinable with color filters that enhance the interactivity of the visualization. To encode multiple channels of information from the anatomical structures, a specific method of color blending is used, which enables the users to access the different anatomical structures selectively, without occlusion. That way the users explore the single layers of the printed visualizations using color filters. Second, 3D papercrafts are generated, which are also examinable with color filters. The anatomical model is unfolded on the paper sheet, can be printed and the user can assemble it and examine it under the color lenses, similarly to the 2D case. The papercrafts may be used as an educational toy in school teaching or for entertainment, since they are very easy to produce and to distribute. We present several 2D and 3D examples of the workflow of the Anatomical Entertainer on models for anatomical education.

Contents

CHAPTER 1

Introduction

Visualizations are embedded in our everyday lives in various forms. They are an important tool to make abstract information easier to comprehend and present. Therefore visualizations can also be very useful in an educational context. They may function as a major instrument to explore data and to get additional insights into research relevant questions. Physical visualization [Phy] is a subdomain of traditional visualization, where the data is represented by means of physical objects. This comes in contrast to the classic procedure of visualizing data on the screen or printing the representations on paper. These objects are often tangible, to further enhance the interaction possibilities. This facilitates focus on the visualization and, therefore, on remembering the embedded information better [JDF13], [SSB15]. This effect can even be enhanced by offering additional interactivity, for instance through shapeable objects. The manipulation of the object helps to engage the user playfully and also increases the information content of the visualization [Stu15].

Among other applications, physical visualizations are used within the field of medicine. Examples of such medical physicalizations can be generic, such as the well-known model kits to showcase the human anatomy or otherwise contextual 3D printed models [SAS+19]. However, the major drawback related to these examples of physical visualization, is that the manufacturing process is rather complex and costly, for most individuals. Even with the recent advances in the field of digital fabrication, these models remain not easily accessible for a lot of people due to expensive machinery and material. As opposed to 3D printing, a traditional 2D approach would only require cheap materials that are available to everyone, such as a common printer and appropriate paper. There are already physical visualizations consisting only of paper [Phy] and a lot of papercrafts in general, that embed information to a certain level. The usage of paper in this context has the advantage to be very cheap and accessible to the general population. Paper can also be folded to easily construct 3D structures. Yet, using paper as fabrication material also entails some drawbacks, for example concerning the representation of nested or overlapping materials. To increase visibility, smart strategies that take advantage of the properties of our physical world need to be employed.

The aim of this thesis is to *design a workflow for the easy, accessible and affordable generation of physical models within the context of human anatomy.* The outcome of this workflow is an easy to produce, tangible and interactive physical visualizations for anatomical education purposes of the general population, such as schoolchildren.

The first step in this workflow is to acquire models of the human anatomy. Anatomical models are loaded into an easy to use application, which helps the user to select adequate visual properties of the different structures, so that under colored lenses or colored light, specific hues of the visible spectrum change with respect to the color of the light or the lens, as showcased in this tutorial [CFT]. This can be useful in the context of the human anatomy, to help the user focus on certain areas or parts of the body. The implemented stand-alone application, built using the VTK framework [VTK], enables the users to load in multiple 3D models as objects and lets them decide whether they want to create a 2D or 3D visualization with the colored models. The 2D visualizations are easily printed and can be explored using the filters described above. To increase the engagement of the user, we also provide additional 3D visualizations. The 3D visualizations rely on a mesh unfolding step, to ensure that they can be assembled to a 3D figure afterwards. Similary to the 2D case, they can be explored using the filters.

Anyone who has access to a computer and a printer is able to create visualizations in our proposed way. These visualizations only need to be created once and can easily be copied afterwards. This makes them a cheap and accessible tool for educational use. The physicality of the created papercrafts adds a playful element to the visualizations and they could be especially useful for the younger population.

The thesis is structured as follows: Chapter 2 gives a quick overview of previous work and other sources of inspiration for this project. Chapter 3 describes the applied methodology. In Chapter 4 the implementation is explained in detail. Chapter 5 shows the obtained results as well as a discussion of the limitations in the current pipeline. The thesis concludes with a short conclusion and possible future extensions in Chapter 6.

CHAPTER 2

Related Work

The first study that evaluated the possible benefits of physical visualization, compared to an on-screen visualization was conducted by Jansen et al. [JDF13]. Their aim was to find out if physical information visualizations have a positive benefit on the memorability since upcoming technologies like digital fabrication make it continuously easier to produce them. The authors also described the disadvantages of 3D visualizations on-screen, like occlusion and navigation issues. To have perceptually similar representations on both mediums, the authors decided to use 3D bar charts for their user study, visualizing country indicator data. The questionnaire that the participants had to answer based on the visualization, was partitioned into three different types of questions, to evaluate the effect of the tangible visualization on varying problems. They additionally observed the effect of a 2D visualization and a stereoscopic monitor in their user study. Their results showed that the physical visualizations outperform the on-screen visualizations, which is related to the haptic aspect of the physical object, that facilitates focus and engagement of the user.

Later, Jansen et al. [JDI+15] conducted a more general study about data physicalization. They emphasized the importance of tangible objects for information representation, by highlighting that physicalizations of data existed long before the written language. They also mentioned a lot of contemporary examples for data physicalization, like three-dimensional maps, where the height encoded the population density. They also mentioned much simpler examples, such as Lego bricks on a board, used to encode the project progress and current tasks. The authors discretized a few key benefits of physical visualizations, for example active perception and natural depth perception and also inherent support for our haptic sense.

In another study, Jansen and Dragicevic presented an adapted information visualization pipeline, for the creation of physical data representations [JD13]. Their aim was to describe the several steps relevant to generate physical visualizations. From the raw data to the physical representation, the pipeline consisted of the following steps:

1. Data transformation, to transform the raw data to suit the visualization.
2. Visual mapping, to map the processed data onto graphical primitives.
3. Presentation mapping, to get the final producible version of the visualization.
4. Final rendering, to bring the visualization into the physical world.

Swaminathan et al. [SSJ+14] showed existing problems and gaps in the creation pipeline of physical data visualizations, which make the automated creations of such data structures complicated. They introduced MakerVis, an application that creates physical visualizations in a semi-automatic process. The user had to choose the data, the mapping method, the visualization type and set the fabrication machine and parameters, from a set of predefined digital fabrication processes, like laser cutting or 3D printing. The application supported simulations of the resulting object and manual assembling and tweaking of the data structure. The enabled creational process took much less time with the application, compared to a process consisting mainly of a manual workflow.

Stusak et al. [SSB15] also wrote about the potential memorability benefit that can result from physical visualizations. They conducted a user study with 40 participants, where they observed the differences in memorability between a digital and a physical visualization, both in the form of bar charts. The digital chart was displayed on a tablet, which brings a bit more interactivity into the perception, compared to a desktop display. The physical bar charts could also be split up into separate parts. After the participants gathered information they filled out a questionnaire, split up into three types of questions concerning extreme values, numeric values and general facts. After two weeks, the participants had to fill out the same questionnaire again. Their results showed that physical visualizations can significantly increase the memorability of information, compared to its digital counterpart. Even though the results of the participants with the digital visualization in the immediate questionnaire were better (digital: 62,8% correct, physical: 59,9% correct), the physical visualization group did better in the second questionnaire (digital: 49% correct, physical: 50,7% correct), especially with extreme values. Stusak et al. [Stu15] also created a paper-based visualization for large events, to see if building your own visualization changes the perception of the data. Both times the feedback they got was very positive, but the actual number of completed visualizations was very low.

In their study of interactive lenses for visualization, Tominski et al. [TGK+17] compared multiple different types of lenses in the context of digital and analog information visualization. They described lenses as a lightweight tool, to alter a visualization and therefore enrich the information content, without changing the base visualization permanently. They mentioned a broad field of digital lenses, from casual magnification to complex transfers consisting of multiple steps. Most of these complex lenses were implemented in an application. The analog alternatives were simpler and less in numbers, but they still mentioned some interesting projects. They introduced tangible interaction or tangible views as concepts besides the mouse/keyboard and touch/multi-touch interaction. An example was the PaperLens project [SSD09], where the visualization gets projected from a beamer onto a piece of paper which can be moved along three dimensions, to load different views of the data. They showcased this type of interface with an anatomical dataset, where coherent bigger structures of the anatomy like the bones, muscles, veins and nerves were viewed separated on different layers, according to the height of the paper lens. The conclusion of the research team was that lenses are widely applicable in the context of visualizations. They can be used in almost every step of the visualization pipeline to enrich the information content and they inherently create an interactive perception of the visualization.

Stoppel and Bruckner presented a printable interactive volume visualization, named Vol2velle [SB17]. They created the effect of transfer functions and different rendering styles, in hardcopies of visualizations, to enhance their interactivity. For this purpose, they created an application, that produced renderings of medical volume data and projected them onto a paper template of a Volvelle 2.1. A Volvelle is an old paper version of a circular slide chart, made out of stacked and rotating pieces. It is therefore very easy to produce. They also presented a version using transparent foil as the print medium for the medical data renderings, to enhance the possibilities of the visualization in exchange for access. The resulting circular chart can be used for educational purposes, as well as a tool within the medical context.

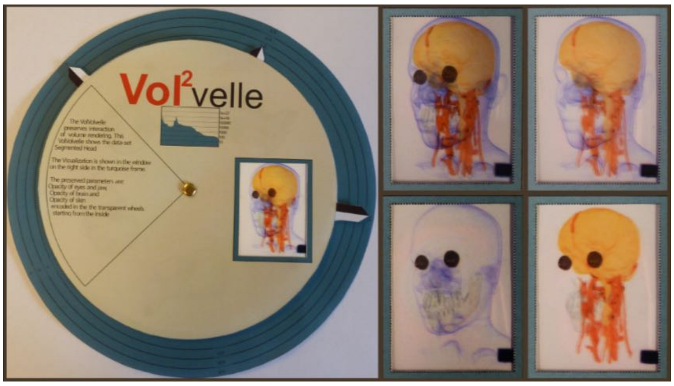


Figure 2.1: A Volvelle from [SB17].

Finally, an important source of inspiration for this project has been the book "Illumanatomy" by Carnovsky [Car], shown in Figure 2.2. They used red, green and blue filters in combination with printed visualizations, colored in cyan, magenta and yellow on white background. Each lens lets a different gamut of light pass and masks away the left-over wavelength range, thus showing the masked color in black and all the others in white and gray. For bigger exhibitions they showcased their illustrations on a large scale, masking between different colors is achieved via colored light.

There is a lot of previous work in 3D printing for medical applications, [Ven14], [API19], [ASS+19], but we consider digital fabrication out of scope within this thesis, as our goal

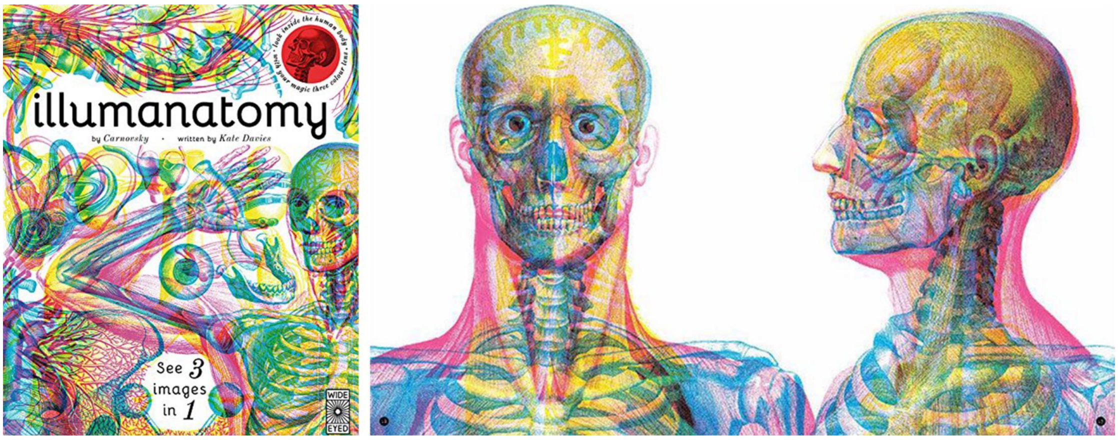


Figure 2.2: The book "illumanatomy" [Dav17] featuring illustrations from Carnovsky.

is to introduce affordable and accessible physicalizations, which are not possible with 3D printing.

CHAPTER 3

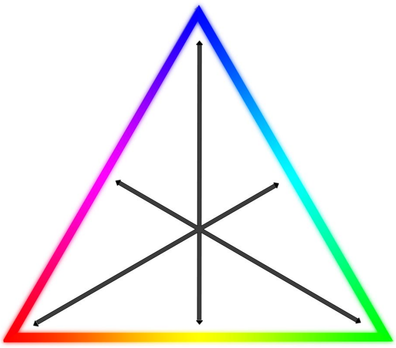
The Anatomical Entertainer

The focus of this thesis is the design and development of a workflow for the generation of tangible physical visualizations of anatomical models, to take advantage of the additional benefit of haptic interaction into the visualization. There are also other senses like smelling or hearing that could be utilized to enhance the degree of information encoded in a physical visualization [JDI+15], but these are not addressed in this project. An extra objective is the accessibility of the resulting physical visualization, as it should be easily producible by laymen without a lot of manual post-processing while keeping the necessary materials, as well as the costs, as low as possible. The following parts of this section describe the methodology, starting with the two main concepts for the physical visualization of the anatomical models. Afterwards, the applied visualization pipeline is explained in detail, followed by a presentation of the implemented application and its possibilities.

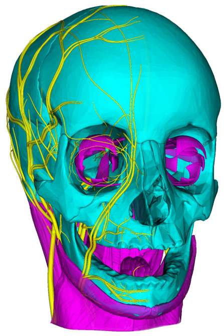
3.1 Physical Visualization Concepts: Color Lenses and Papercrafts

The first concept used in this project is the suitability of color blending, in combination with colored lenses, to enhance the educational effect of a physical visualization. This is possible, by taking advantage of the properties of light. The distinctive wavelengths of the light are blocking other particular wavelengths to a certain degree, resulting in a colorless, thus darker or black impression of the absorbed light [CFT]. Utilized on a visualization, the different layers of the figure could be explored through these colored lenses. This is also the effect that Carnovsky [Car] is using for illustrations, shown in Figure 2.2. This could be an alternative to the use of inaccessible transparent material for printing. A colored lens is necessary in exchange, but colored sheet protectors or colored notebook covers have proven to be sufficient. Better suited than these plastics are for instance old blue-red 3D glasses, camera or light filters and colored light itself.

The color combinations that interact the most efficient for our purposes are illustrated in Figure 3.1a, where opposing colors are absorbing each other. For example, cyan and red are interacting in this way, Figure 3.1c shows the bones, which are colored in cyan, turning black under the impression of a red filter, all the other structures are turning red and are hidden. The second concept used in this project is to create an approach that simplifies several of the steps involved in the creation of a physical visualization made out of paper, both in 2D and in 3D.

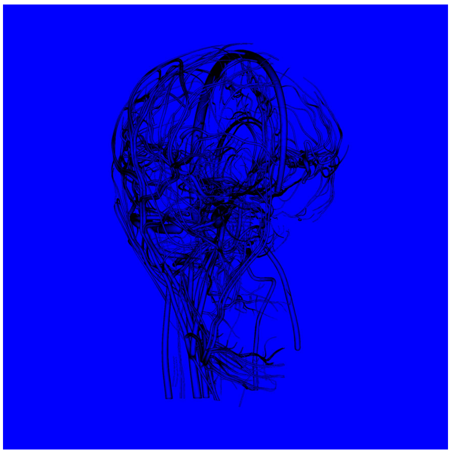


(a) Color compensation triangle.



(b) Anatomical structures loaded into the application.

(c) The structures from image (a) multiplied with a red filter.



(d) The structures from image (a) multiplied with a blue filter.

Figure 3.1: (a) The color compensation triangle that shows which colors are suited for filtering, where opposing colors absorb each other and (b) example rendering made with the Anatomical Entertainer. Image (c) and (d) show the effect of the color filtering when a red or a blue filter is applied to an image, that is processed with the Anatomical Entertainer.

Our approach should be able to visualize multiple 3D meshes and enable the user to adjust their respective visual parameters, mainly the color and the opacity values, but also the perspective values for the camera and the alpha-blending of the meshes. At this point, it is already possible to render an image, that is examinable with the colored lenses. This printable 2D visualization is a form of physical visualization, but it does not offer a lot of interactivity, for the users. More complexity would be offered to its observers if the physical model is three-dimensional. It makes sense to visualize anatomical data, using 3D physical visualization, as this approach preserves the spatial context and does not create an abstraction of the data through 2D projection. Therefore, the 3D viewport can be projected onto a texture, which is then mapped onto a simple 3D mesh, encapsulating the visualized anatomical structures and the resulting paper-mesh. In the last step, this mesh is unfolded to a 2D grid which satisfies the constraints of a papercraft, thus it is possible to print, cut it out and fold it into a physical visualization. Constructing the visualization oneself has the potential to enable a deeper understanding of the underlying data since the user has to give a lot of thought into the spatial relations of the anatomical structures to glue the single paper patches appropriately.

3.2 The Anatomical Entertainer Pipeline

The process which we followed to obtain the desired physical visualization involves several steps, which are structured according to the reference model described by Jansen and Dragicevic [JD13]. The pipeline of the Anatomical Entertainer branches out into two different visualization techniques at the presentation mapping step. The user has the option to print 2D visualizations or a template to create a 3D papercraft of the selected meshes. The pipeline is shown in Figure 3.2.

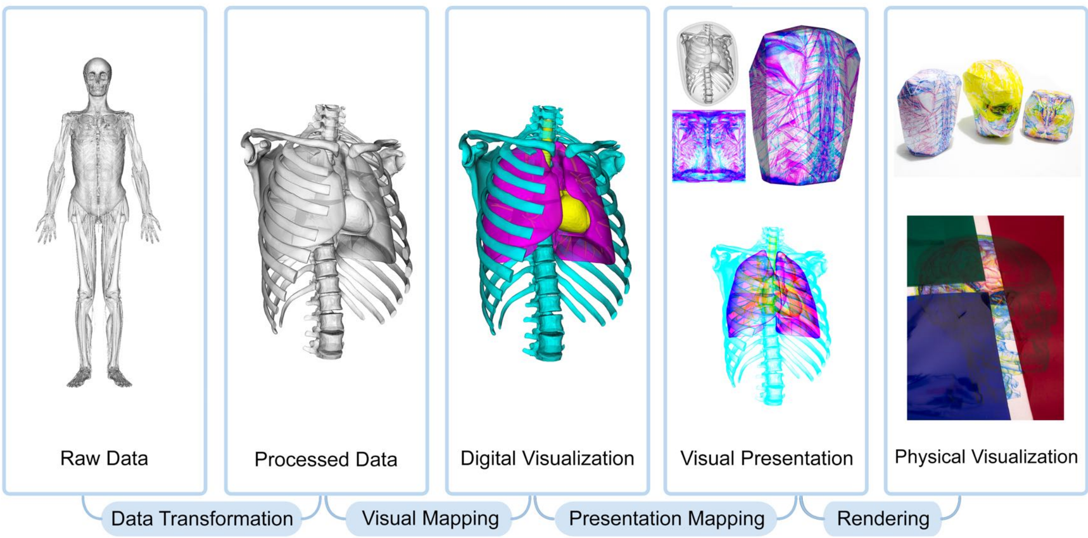


Figure 3.2: Visualization Pipeline of the Anatomical Entertainer.

In the first step of our visualization pipeline, it is necessary to obtain raw data. In our case, these are 3D meshes of the human anatomy, which we obtain from the Anatomography website [BP3]. These are closed triangulated meshes from the entire human body. For example, the skull contains 1.1 million triangles and the muscles of the torso about 1.3 million triangles. These are generic anatomical models, but if the users would have available segmentations from their medical imaging data (e.g., a CT scan of the upper abdomen or an MRI scan of the head), the segmentation could be employed in our application, as a surface mesh, as well.

Next, in the data transformation step, preprocessing is necessary to import the meshes into the application because we download the anatomical structures from [BP3] as a whole and need to split them up into smaller sub-components. The meshes are now ready to be imported into our application, depicted in Figure 3.3, which creates the 3D digital visualization of our data. All of the structures are displayed in the viewport, so the user can make sure that the right meshes are selected. Since the user can also select the transparency of the displayed meshes, the VTK "depth peeling" functionality is used. This feature enables the accurate rendering of translucent meshes without sorting the polygons, by peeling the geometry from front to back.

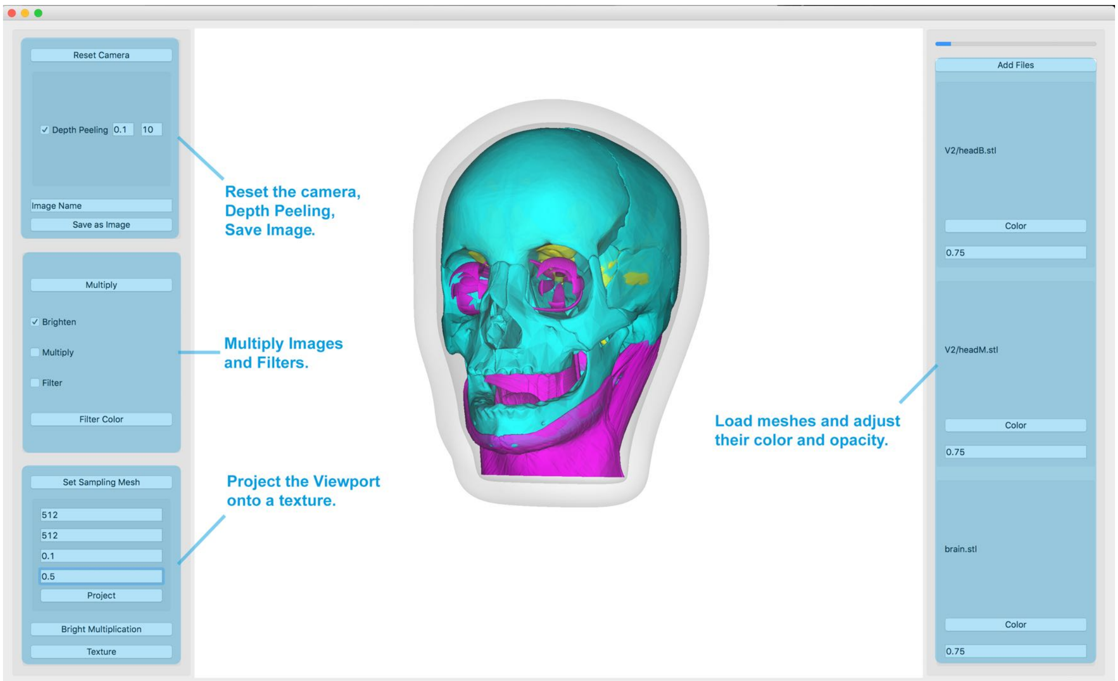
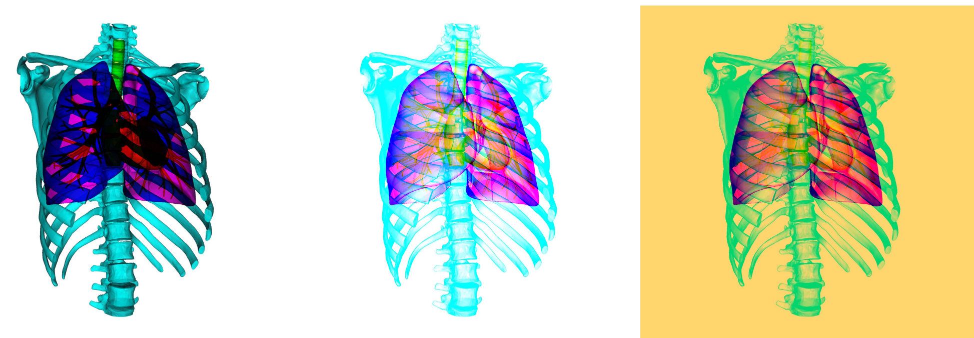


Figure 3.3: User interface of the application for the generation of the physical visualizations of the anatomical data.

In the step of visual mapping, the first thing the user needs to adjust are the several visual attributes of the meshes, to optimize the results that the future steps of the pipeline will produce. The most important thing is to adjust the color of the meshes. To get an optimal effect with the color blending of the resulting physical object only cyan, magenta and yellow should be used in combination, as discussed in Section 3.1. Other colors will not give a strong enough segmentation impression under a colored lens or colored light, but can regardless result in a presentable visualization without the use of lenses or light. If the visualization should only be processed as a 2D image and not unfolded, adjusting the camera is advantageous since the same camera is also used for the rendering process.

For the meshes, it does not make sense to use color values deviating from perfect cyan, magenta or yellow, since the blending and filtering would only work worse. Meshes that are colored in red, green and blue are not combinable with the way the meshes are blended and would result in black structures. It would be very easy to implement the blending of these colors, but the resulting visualizations would need a black background and are therefore not optimal for printing purposes.

The user can blend the selected and imported meshes in the presentation mapping step so that in the resulting rendering, nested and thus hidden structures, are drawn by multiplying the colors of all overlapping structures. Since the resulting renderings tend to get dark with multiple overlapping color values, especially since we are using very saturated colors, an additional processing step can be enabled by the user, resulting in a much more understandable, brighter image, shown in Figure 3.4b. The users can also select to create a 3D papercraft at the presentation mapping step of the pipeline. This way the algorithm will create a texture in the presentation mapping step.



(a) Multiplied structures before brightening.

(b) Multiplied structures after brightening.

(c) Color transformation using an arbitrary filter.

Figure 3.4: Example renderings made with the Anatomical Entertainer.

At the visual presentation step, it is possible for the user to test the digital visualizations, under the impression of different digital color filters. The filter and the rendered image are first normalized and then multiplied together. If the users can change the visual parameters of the rendering until they are pleased with the current visualization, given by the chosen meshes, colors and camera settings, then they can move on to the next step. For the 2D physical visualization, this step would be to render the final result and print it. The images, generated in the described way, already offer some degree of interactivity.

The use in combination with the color lenses is shown in Figure 3.5. Otherwise, the user chooses to create a papercraft.

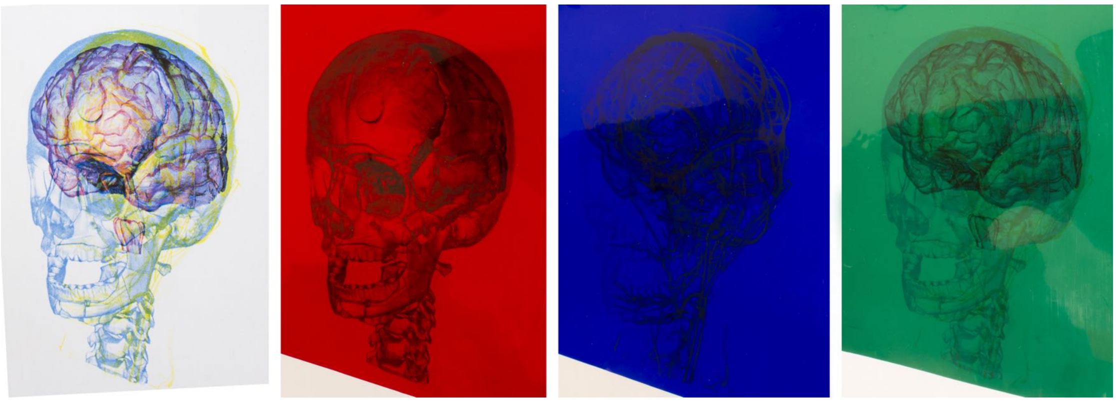


Figure 3.5: Printed visualization in combination with different color filters (red for the skull, blue for the vasculature and green for the brain).

If the users select the projection methodology to create a papercraft model additional steps are necessary to finish the visualization. Given that the user has already selected the anatomical meshes and set their color and opacity as desired, the presentation mapping starts with the selection of the position and the focus point mesh, shown in Figure 3.6. These two meshes act like a convex hull for the anatomical structures, where one is generated with a larger positive offset from the anatomical structures. The unfolding process and its results depend heavily on these meshes, referred to as sampling-meshes in this thesis. They are used for the camera settings in the projection process and have to be combinable among each other and also with the selected anatomical meshes.

There are three types of sampling-meshes used in this project. The first mesh that is created, is the paper-mesh which resembles the finished papercraft, the other sampling-meshes are easily built upon it. The second mesh is responsible for the positioning of the camera focal point, referred to as "focus point mesh". The last sampling-mesh is responsible for the positioning of the camera around the anatomical structures, to unfold the viewport onto a 2D image, referred to as "position mesh". With most of the structures tried out in this project, the two different meshes for camera focal point and position were not necessary. The position mesh is simply a replication of the focus point mesh, only altered to have a larger offset to the surface of the anatomical meshes. Yet, some special cases are profiting from two different meshes for the sampling, described in more detail later.

To project the anatomical structures onto the paper mesh, a camera is moving around those structures along the position mesh, using a regular spacing along each degree of latitude and captures a frame from each anatomical mesh. The frame size and other camera settings are based upon the user input. The single camera positions on the surface of the position mesh are calculated by casting a ray, originating from the viewport center,

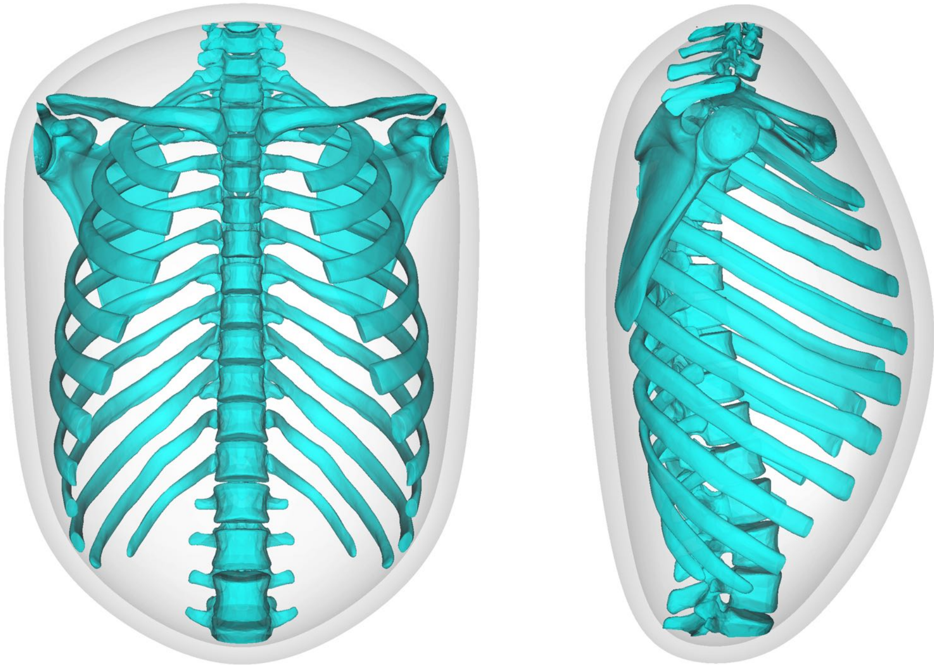


Figure 3.6: The bones of the torso colored in cyan, encapsulated by their focus point and position mesh colored in grey.

to points on the surface of a sphere that is stretched over all rendered meshes, as shown in 3.7. The surface points of the sphere are calculated with spherical coordinates, given a longitudinal and latitudinal step size from the user input. The up vector of the camera is calculated according to the transversion between spherical to cartesian coordinates with a 90 degree shifted polar value given by the following formulas,

$$x = Radius * sin(polar - \frac{\pi}{2}) * cos(azimut)$$
$$y = Radius * sin(polar - \frac{\pi}{2}) * sin(azimut)$$
$$z = Radius * cos(polar - \frac{\pi}{2})$$

and then orthogonalized via the VTK functionality. Each intersection of one of those rays, with the position mesh accounts as one position for the camera. The focus point of the camera is calculated with the help of the focus point mesh. Since this mesh acts as a tight convex hull for the anatomical meshes, the closest point on the surface of this focus point mesh to the respective position of the camera is a pretty stable value for the focus point.

To create a texture of the whole mesh unfolded, the frames need to be stitched together. Therefore, each of the different frames is placed onto a 2D regular grid, at the respective position according to the lateral and longitudinal position of the sampling camera. Therefore the camera uses a parallel projection so the background and structures altogether will not get distorted too much.

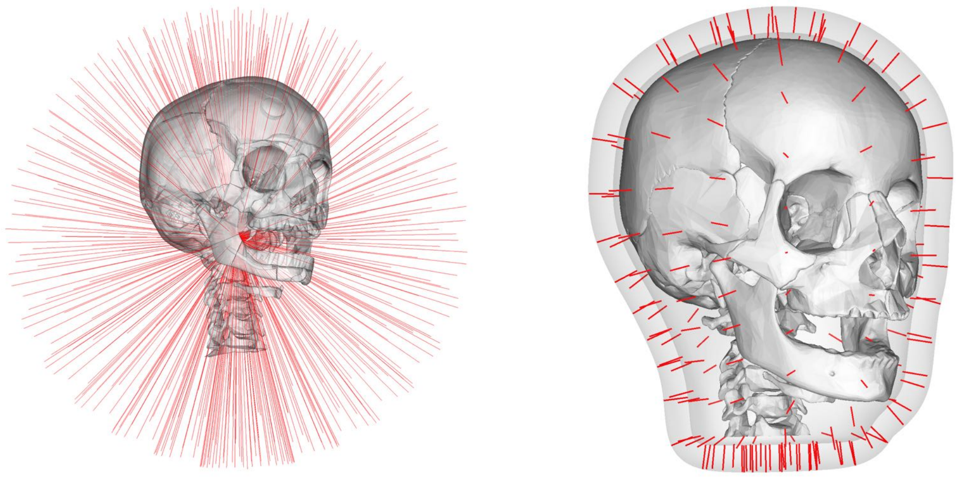


Figure 3.7: The image on the left shows the rays sampling the position mesh to get the position points of the camera. The one on the right shows red lines connecting the position points sampled on the larger sampling mesh and the focal points of the camera, on the smaller sampling mesh.

After the importing of the sampling meshes, the user has to adjust the settings of the projection, i.e., the longitudinal and latitudinal sampling rate and the different field of view values for the samples between pole and equator. These two values are used to reduce the oversampling of the pole areas since the lateral distance between the camera positions gets smaller the closer they get to the poles. The first value represents the camera field of view used close to the pole. VTK refers to it as the parallel scale in the context of the parallel projection. The second value is the parallel scale along the equator. The values between those two, are calculated using a cubic-ease-out-function:

$$ZoomMin + (ZoomRange * (\frac{longitudinalPoint}{\frac{maxLongitudinal}{2}} - 1)^3 + 1)$$

Therefore the camera zooms in when it gets closer to the poles from the equator, and zooms out when it is getting closer to the equator from a pole. This functionality could be improved, by an algorithm that automatically calculates an optimal zoom factor, based on the distance to the previous sample points. That would be helpful because now the user has to choose a value that can only be estimated based on previous results.

The sampling meshes are created manually in Blender [ble], the following workflow, describing the necessary steps, could have been implemented, but the efforts necessary to do so would have gone beyond the scope of this thesis. Additionally, it made sense to first test some different methods and topologies for the sampling meshes, to get optimal results.

The paper-mesh is created, in a very fast process, with the use of a few Blender methods. Having complete design freedom could be helpful to stylize the data sculptures, but in this project, it should be averted to abstract the anatomical structures even further. In detail, the workflow is to import the anatomical accurate meshes, that are associated with the body region that the paper-mesh should cover, into Blender. Then these meshes are joined together into one structure, therefore if one of those meshes is completely contained into another mesh it can be omitted. Afterwards, a simple mesh, i.e. a cube, encapsulating these anatomical meshes is created. This cube is then subdivided and wrapped around the original meshes using the „Shrinkwrap“ functionality. This moves the surfaces of the paper-mesh onto the respective nearest surface point of the joined structures. This way the paper-mesh encapsulates the anatomical meshes, using some positive offset, so it will not stick on the surface of the anatomical meshes. To be able to reproduce the mesh as a papercraft it must only contain planar faces, therefore the digital paper mesh gets partially triangulated in a final step. Sometimes it seemed to make sense to manually change the topology a little bit and remove sharp edges or spikes, not necessary for the information degree of the resulting papercraft, so it would be easier to unfold.

The second sampling-mesh responsible for the focal points of the camera is built upon this first one, by subdividing it more often and shrinkwrapping it each time back onto the original meshes, again using some positive offset. An additional last step of smoothing was necessary to get decent results for the camera focal point. The last sampling-mesh to calculate the position points of the camera is created by simply duplicating the focus-point mesh and offsetting it more. The used Blender modifiers are depicted in Figure 3.8 and the resulting sampling meshes are shown in Figure 3.6.

At this point, the user successfully created a texture containing all of the selected structures, an example is shown in Figure 3.9. It is possible to select a paper-mesh, that the texture should be applied to. This paper-mesh is then loaded into the application and automatically unwrapped to get appropriate UV-coordinates. These are coordinates assigned to each vertex of a 3D mesh, to indicate how a texture should be applied onto it. Afterwards, the textured paper-mesh gets displayed in the viewport in the visual presentation step and can be evaluated by the user, shown in Figure 3.9. This process also exports a ".obj" file of the unwrapped paper-mesh.

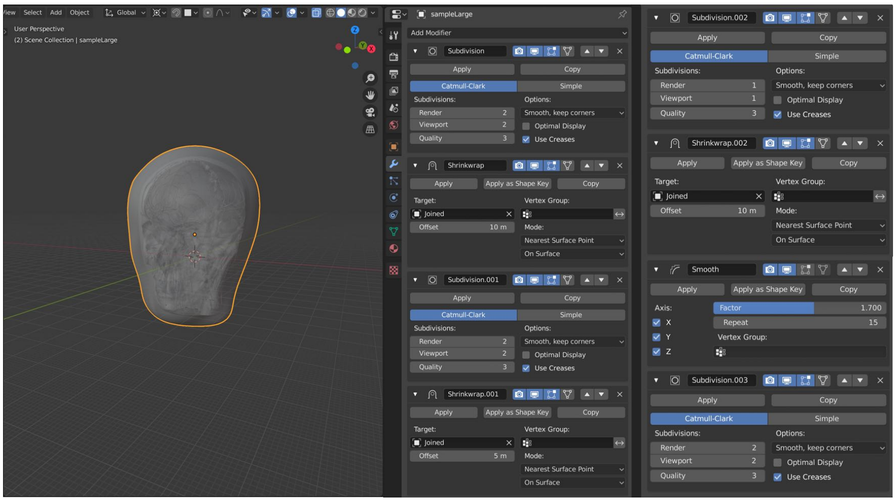


Figure 3.8: This image shows the necessary Blender [ble] modifiers to create the position mesh. Only the first two modifiers are needed to create the paper mesh. To create the focus point mesh the offset of the shrinkwrap modifiers needs to be reduced.

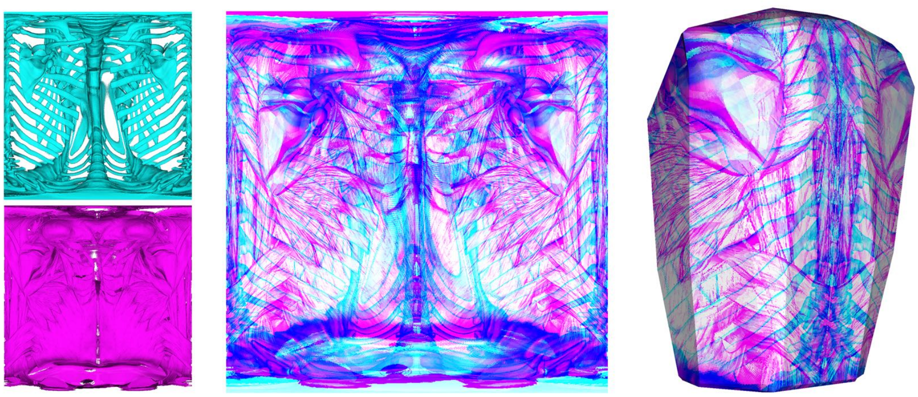


Figure 3.9: On the left are two textures, created using the bones and the muscles of the torso. The texture in the middle shows the multiplication of the two textures on the left. On the right is the texture mapped papercraft, displayed in the application.

In the rendering step of the paper meshes, the Blender add-on “Export Paper Model” is used. First, we need to import the unwrapped paper-mesh into Blender and use the “unwrap” function of the paper export add-on, to unfold it into a 2D grid. The add-on also creates lashes, labels and gridlines to make the folding of the papercraft easier and the color of these elements is adjustable. The unfolded paper-mesh is exported as a pdf document and ready to be printed.

The printed template is then cut out, along the grid and lashes and glued together, as shown in Figure 3.10. Multiple of these papercrafts can be combined to a bigger structure, where the hidden surfaces resemble crosscuts. All of the printed visualizations, either two- or three-dimensional can now be examined using the previously discussed colored filters, as can be seen in the rightmost part of Figure 3.2 and in Figure 3.5.

Figure 3.10: The resulting papercrafts. The torso on the left shows the bones colored in cyan and muscles in magenta. The head in the middle shows the skull in cyan, the brain in yellow and the lower muscles of the head in magenta. The hips on the right show the bones in cyan, the muscles in magenta and the veins in yellow.

CHAPTER 4

Implementation

All of the code is implemented in Python, a very popular general-purpose programming language and mainly build up upon the "Visualization Toolkit", short VTK framework [VTK]. VTK packs a lot of functionality to create visualizations and process two-dimensional, to multi-dimensional data. VTK is mainly written in C++, but wrapped into a lot of other language bindings, especially the Python binding is well refined.

The meshes used in this project are downloaded from the site "Anatomography" [BP3]. It offers very detailed scans of the human body under the creative commons license, converted to .obj-files. To download them, it is only necessary to select a version of the data and the desired parts of the anatomy, since they are split up into smaller meshes. There were some smaller problems with the Meshes, for instance, some structures of the anatomy were missing or modified, and it seemed like the point of origin differed a little for each mesh, thus they were a bit misplaced after downloading. Fortunately, that was not the case for their most recent dataset.

The interface of the application is written using the Qt Framework, more precisely the Python binding, PyQt5. The VTK framework offers an interface to work with Qt, making the inclusion very easy. The interface itself consists of three sections, depicted in Figure 3.3. On the left are the different tools to create visualizations, in the middle is the main window displaying the meshes or rendered previews and on the right side are the tools to import and adjust the meshes.

The tool section is subdivided into groups, according to the use case of the different Interface-elements. The first one consists of the depth peeling settings, which is the previously described VTK functionality to correctly render translucent geometry quickly. The following groups contain the main elements of the application. The first of them includes all of the functionality to create 2D visualizations. The "multiply"-button creates a single fixed image, using the current camera settings and loaded meshes.

This image is created with the help of numpy, a well-known python framework, that packs great features to handle multi-dimensional arrays. For every loaded mesh, the implemented function, first renders the respective mesh in its dedicated window, this window is then converted to an image using the current camera settings and converted to a numpy array using the VTK support for numpy. Afterwards, the numpy array gets normalized and multiplied with the following structure, processed the same way, as seen in Figure 3.4a.

Below is a “multiply”-checkbox, if it is enabled the multiplied functionality is running while the camera updates, but due to the time-consuming operations that are executed in the context of the multiplying, it is running with a very low frame rate.

Since the images tend to get very dark with the simple presented multiplying alone, especially when there is an additional filter enabled, as shown in Figure 3.4a, another processing step can be carried out, by enabling “Brighten”. If so, after each structure is rendered to a numpy array and before they are multiplied with the consecutive structure, all the color values of the rendered image are mapped to brighter values of the same color. Therefore, for instance, in an image containing a cyan-colored structure, all of the dark, near-black values would get changed to perfect cyan, thus (0,0,0) to (0,255,255) in RGB and higher color values are changed to brighter cyan values increasingly until perfect white is reached. The thresholds for this function could be adjustable by the user, but right now they are implemented as fixed optimal values.

The next setting that can be activated, is the “Filter” option. It simulates the effect of colored lenses or light. The user can select the color. If the filter is colored in one of the base colors of RGB and the meshes are colored in one of the three base colors of CMY, only one of those meshes will be rendered, colored in black with the background in the color of the chosen filter, as seen in the Figures 3.1c and 3.1d.

The next group contains the functionality to unfold the structures in the viewport onto a texture. This is also done with each loaded structure on its own to blend the single textures using the proposed multiplying and brightening methodology. Like previously described, the algorithm needs two additional meshes for the sampling of the viewport, the “Set Sampling Mesh”-button lets the user choose one after the other. Beneath it, are text fields to set the longitudinal and lateral sampling rate individually. These are the two values that set the accuracy for the projection process. The frame size for each sample that the camera captures is estimated by dividing the texture width, which is set to 1024 pixels by default, through the lateral sampling rate, same accounts for the height and the longitudinal sampling rate. The last two text fields set the maximal and minimal zoom to compensate for the oversampled pole areas.

For the last step of the unfolding, the user hast the “Bright Multiplication” button that multiplies the images of the different unfolded structures together, with the brighten option enabled. The result is a quadratic texture, containing as many structures as selected by the user. Now it is possible to select a paper-mesh with select “Paper Mesh” button and project the previously rendered texture onto it, so the user can see the digital version of the resulting papercraft. In this process, the UV-coordinates of the paper mesh are set by calculating the normalized position of the vertex in polar coordinates.

The resulting texture-mapped mesh is unfolded to a two-dimensional grid with the help of Blender [ble] and the “Export Paper Model” add-on, preserving the right size ratio of the faces and containing added laces to glue the adjacent edges of the papercraft together.

CHAPTER 5

Results

The developed application offers different possibilities to create interactive physical visualizations either as 2D images or as unfoldable 3D papercrafts, the results of which are going to be described separately.

5.1 2D Visualizations

For the 2D visualizations, a few steps described in Chapter 3 are required to generate physical models such as these depicted in Figures 3.5. As described before, we first have to import a selection of meshes in the data transformation step. Then the user chooses and adjusts the options available for the application viewport and meshes until he or she is pleased with the result in the visual mapping step.

The first implemented version of the blending between the meshes, turned out very dark as displayed in Figure 3.4a. The results of the presentation mapping step were much clearer with a brightening step, as shown in Figure 5.2. There are multiple different combinations of colors and transparency levels, for the meshes and filters, resulting in completely different impressions. Figure 5.3 shows the possible outcomes for the rendered visualizations, using only cyan, magenta and yellow to color the meshes and red, green and blue for the color filters. There are also other colors imaginable for the filters, shown in Figure 3.4c, but the filtering effect is reduced and these colors are harder to obtain as analog filters.

The filter material used to evaluate the prints was color foil for light kits from the label "Eurolite". The implemented digital preview of the color filters gives results that are a bit better compared to the analog counterpart, especially with the blue filter, but they still give a good first impression of the results achieved with an analog filter. Between the main three analog filter colors, the red filter gives by far the best results and absorbs only very little of the color spectrum besides the wavelength of red, so the resulting impression

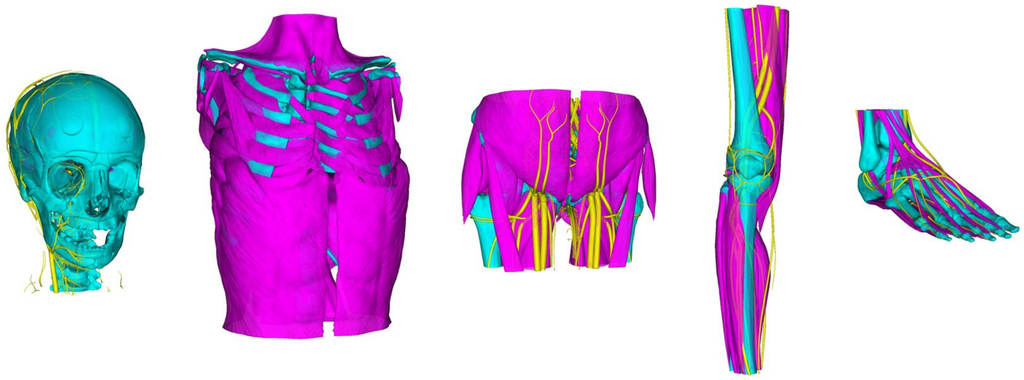


Figure 5.1: This image shows the bones, muscles and veins of, from left to right, the head, the torso (without veins), the hips, the leg and the feet, as displayed in the Anatomical Entertainer. The bones are always colored in cyan, the muscles are colored in magenta and the veins are colored in yellow. The head contains the brain colored in magenta instead of the muscles.

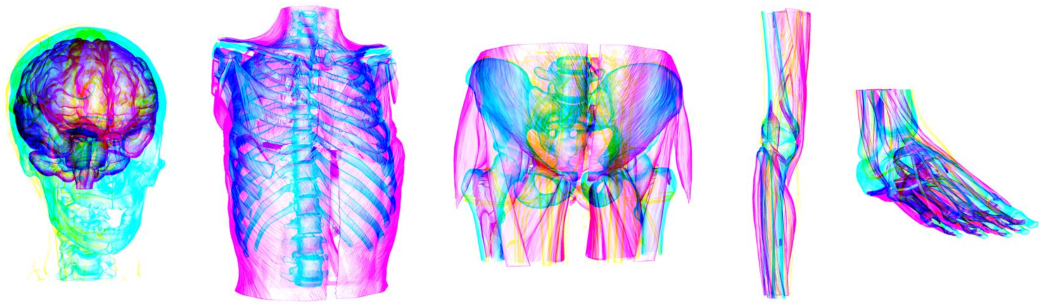


Figure 5.2: The structures shown in Figure 5.1 when multiplied and brightened.

is as good as the digital preview. The green filter also works pretty well and gives results close to the digital filter, but it also absorbs some of the blue values. The capabilities of the blue filter were only limited since the used filter absorbed a lot of the other colors, but also a lot of light in general, resulting in a very dark impression. The visualizations were printed using different low to middle-end printers, which lead to different colors in the prints, but overall the resulting impression using the lenses stayed the same.

To showcase the blending method with meshes that change their state a mesh created in personal context was used as a toy example, depicted in Figure 5.4. It is a rigged mesh of a stylized crocodile, showing a bit of its anatomy. The example suggests that the method would also work well with deforming anatomical structures, like for instance the beating heart or the blinking eye.

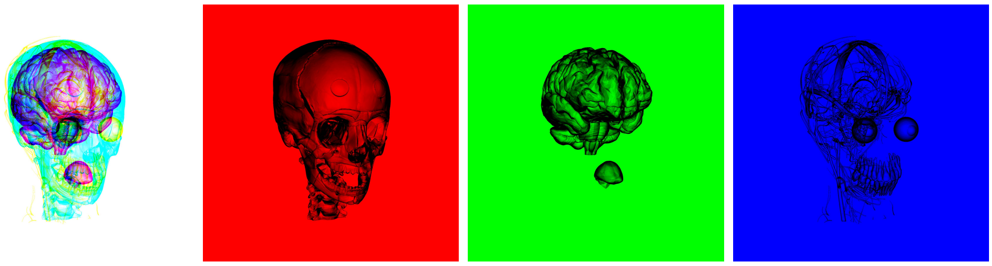


Figure 5.3: A digital red, green and blue filter applied to the image on the left. See Figure 3.5 for comparison to analog filters.

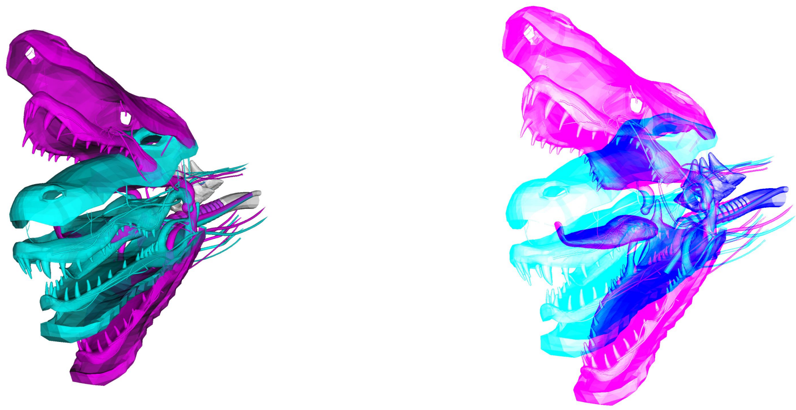


Figure 5.4: Example for state-changing meshes.

5.2 3D Papercrafts

The other visualization option that can be selected in the presentation mapping step is the creation of 3D papercraft models. The resulting figures are very easy to distribute and produce. Additionally, the papercraft introduces several levels of interactivity due to the assembling and the tangible figures. The effect introduced by the color blending in combination with the filters, adds layers to the figure and thus a view on the inner structures to the opaque papercraft.

The 3D papercraft option builds upon the 2D visualizations introduced in the previous section. Instead of printing the 2D visualizations on paper, we will first unfold them on the plane and print them for assembly to 3D. The unfolding of the viewport has some similarities with the process of texture baking, in modern modeling applications. The implemented algorithm shoots rays from a larger outer mesh onto a smaller inner mesh and projects texture and light information from one onto the other. Since the application is mostly built upon VTK, the sampling functionality was implemented using its core functionalities, without altering anything in its pipeline. The results, therefore, take very long to render, this process may be significantly accelerated by making adjustments to the implementation.

In the first state of the implementation, the program simply projected the rendered meshes onto a sphere, resulting in very distorted, but acceptable first results, as seen in Figure 5.5. Afterwards the code was adapted, to project the rendered meshes onto the previously created paper-mesh. This resulted in overall less distorted results but introduced a lot of artifacts. This happened especially in regions where the distance from the sampling meshes to the anatomical meshes varied since the zoom factor of the projecting camera was constant. For this reason, a varying zoom factor was added to the unfolding reducing the artifacts a bit, as seen in Figure 5.6. The user given zoom values for the projection are still suboptimal for the workflow and the results in general. The user has to enter a value, which is only based on previous results, thus it would be good to change this method, to automatically calculate the zoom factor for each sample.

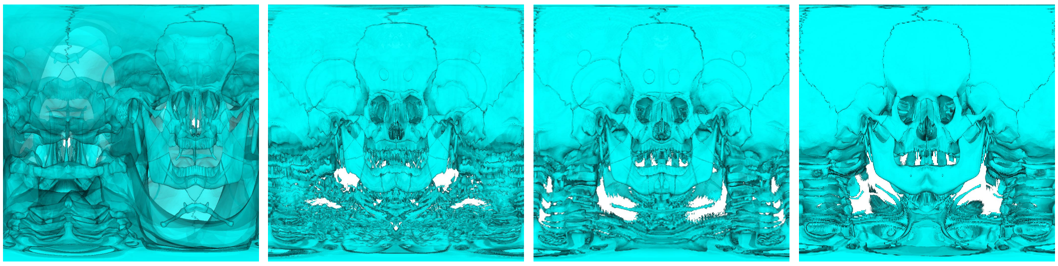


Figure 5.5: The influence of different sampling-meshes onto the projection of the skull. The image on the outer left is simply projected onto a sphere using no sampling-meshes. Note that for the rendering of the image on the outer right a slightly different anatomical mesh was used, then with the previous renderings.

Much more improvement has been achieved by changing the sampling meshes. First, the paper-mesh got substituted for an upscaled version of the focus point mesh, thus the smoothed convex hull of the anatomical structures, this resulted in a big decrease of the distortion and the artifacts. Afterwards, the creational process of the sample-meshes was changed, so it would only rely on implementable core methods of Blender. This again led to better results with the projection in general, because the distance between the surfaces of the focus point mesh and the surfaces of the anatomical structures is now set to a constant value, depicted in Figure 5.5.

Also early in the implementation, the camera sampling-meshes encapsulated all of the selected anatomical structures at once, but this approach introduced several unwanted artifacts. Since we project all the anatomical structures onto the used sampling-mesh, smaller inner structures seemed inflated to the surface of the used sampling-mesh. The

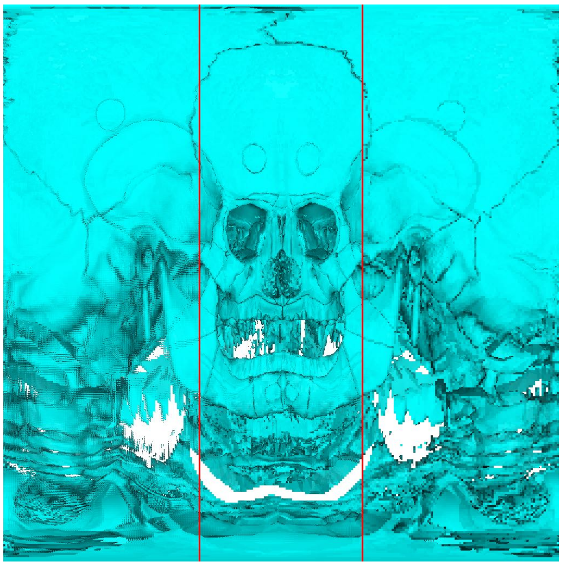


Figure 5.6: The left area of the image shows the artifacts resulting from a low zoom value. The right side shows a zoom value that is too high. The middle section of the image has a more fitting zoom value, thus fewer artifacts. Please note that the shown texture was generated, using deprecated sampling-meshes, leading to the artifacts along the neck region.

same also accounts for larger meshes, consisting of a lot of thin structures, like the human skeleton or the veins. The curvature of the skeleton mesh also introduces a lot of artifacts in combination with a larger sampling-mesh, as shown in Figure 5.8, that is created using the convex hull of the muscles.

Much better results are obtained by having a selection of dedicated sampling meshes for each anatomical structure that is loaded into the viewport, as seen in Figure 5.7. This approach also introduces the opportunity to have sampling meshes, that do not cover the whole sampling-space projected onto the texture, rather only a small area of it. This makes a lot of sense with nested anatomical structures. For instance, the eyes are heavily deformed on the resulting texture, if they are projected onto the much larger structure of the skull. This can be prevented by using only a selected area of the larger sampling mesh used in the same unfolding process, making it possible to showcase more anatomical structures on every single papercraft. Unfortunately, this leads to a strong dependence between the single sampling meshes, making only a few combinations reasonable for the same texture, therefore it would be good to implement an automatic approach for the creation of these sampling meshes during runtime, in the future. The results of the projection are also heavily dependent on the form of the sampling mesh, the offset to the anatomical meshes and the scale overall.

From this point on, each mesh had a dedicated sampling mesh, since they were easier to create and the results of the projection rely greatly on the combination of anatomical

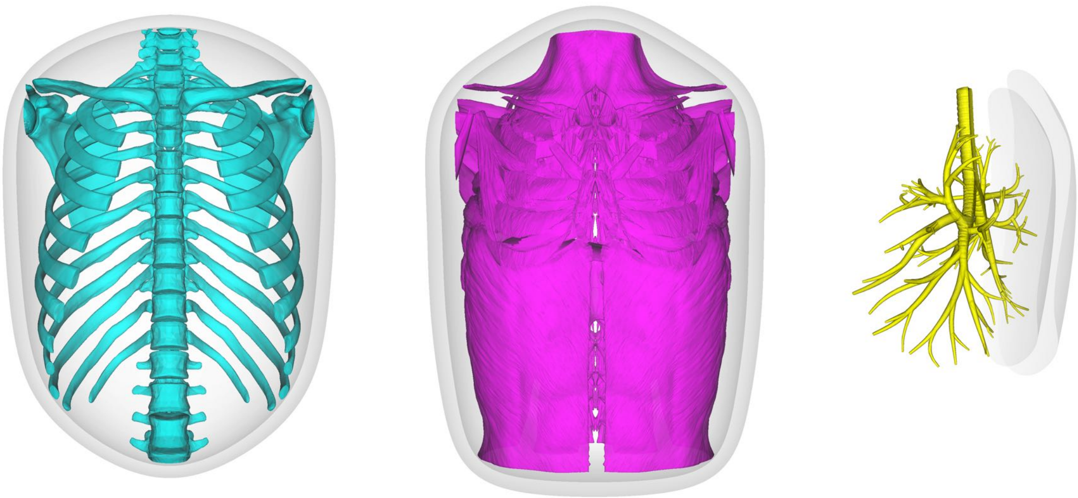


Figure 5.7: Sampling-meshes, on right for the bones of the torso, in the middle for the muscles of the torso, and on the left the partial sampling-meshes for the bronchia. Results for the projection of the bones and the muscles using the shown sampling meshes can be seen in Figure 3.9.

structures and sampling-meshes. Bigger structures like the bones or the muscles need a dedicated mesh, enclosing the whole structure. Other structures like the brain or the lungs, can be projected onto the sampling-mesh of a bigger structure. They will seem inflated but still readable. Smaller inner organs would get deformed too much if projected onto the surface of a much larger mesh, but they can simply be rendered by using smaller sampling meshes, created from a small region of the original encapsulating sampling-mesh. The problem with this method is, that in this process two different sampling meshes need to be created for the position and the focus point of the camera. Simply upscaling the used segment of the sampling-mesh would result in a deformation of the inner organ, approximately to the form of the sampling mesh. Unfortunately, the same applies to the bones in the human extremities, which happen to be rather long and thin and sit in the innermost position. The bones in this body region are therefore a good pick to be only projected from a limited angle and not onto the whole encapsulating sampling-mesh, in which they are contained.

The region of the human anatomy, that got the most attention was the head, which also led to the spherical sampling of the view space. The brain was projected using the convex hull of the skull, which reduced the distortion. Only the creation of the sampling-mesh for the eyes needs a bit of manual adjusting because the eyes were very prone to deformation. The manual adjustment consists of modifying the outermost vertices so that the camera will always capture the edges of the inner geometry. The veins happen to feature a lot of occlusion and thin structures, which the application cannot handle that well. The main problem here is that the camera can see a lot of the rear structures which cannot be rendered continuous due to the parallel projection. These structures get a little over-sampled, which results in some aliasing effects. The applied Gaussian blur and very fine sampling-meshes help a bit with this problem. We ended up using a sigma value of 0.6 in both directions for the blur, since smaller, close to pixel-sized, artifacts were removed but larger structures were only changed a little.

The structures contained in the torso also worked well with the algorithm. The first implementations and sampling-meshes resulted in a corrupted texture of the rib cage, but later versions of both handled the projection much better, as seen in Figure 5.8. The same as with the vein mesh in the head applies to the mesh of the veins contained in the torso. Because it is very complex and features a lot of depth and occlusion, many of the structures are not in an optimal render distance for the camera. The other inner structures of the torso, that were tested like the lungs, bronchia, heart and kidneys worked with the partly projection along on direction, but just like the eyes, these meshes are very prone to deformation resulting in corrupted results, if a suboptimal sampling-mesh is used.

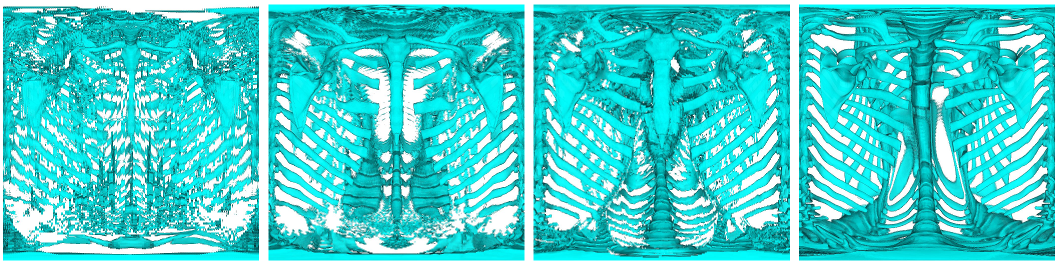
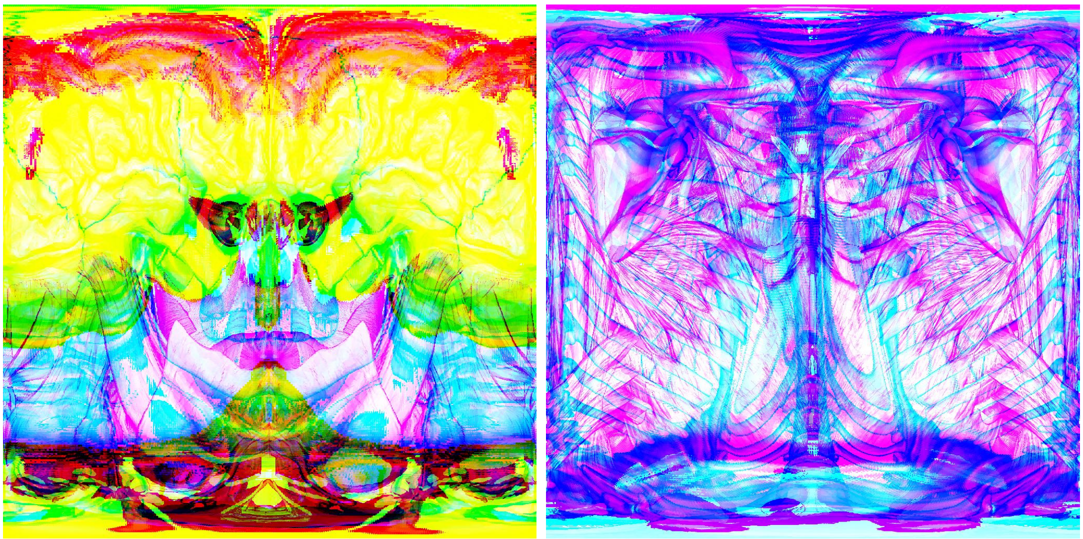


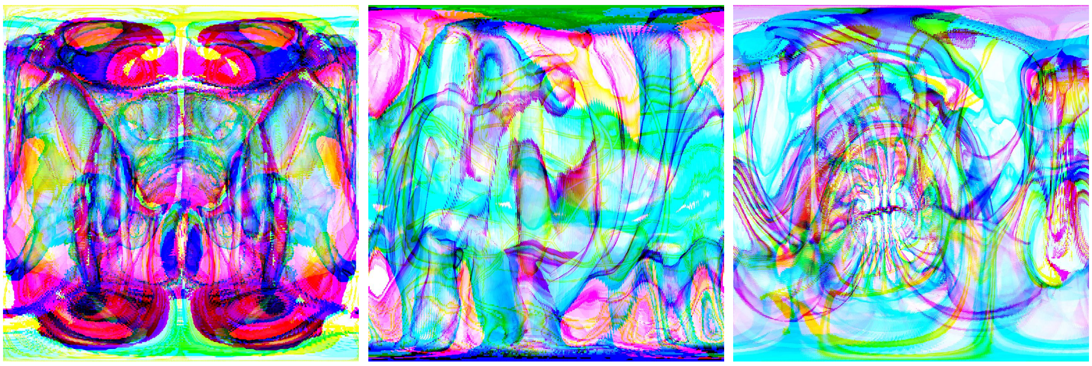
Figure 5.8: The improvement achieved using different sampling meshes shown from left to right. Both images on the left show a texture of the bones in the torso when projected using the sampling mesh of the muscles. The images on the right show an improved version where every structure has its dedicated sampling mesh.

The hips also gave decent results when unfolded. Only the meshes of the feet or the legs were rather difficult to project, without distorting them too much, due to their oblongness, as shown in Figures 5.9d and 5.9e. In the case of the legs, the muscle mesh worked fine, but the bones are too thin to be projected onto the paper mesh completely. Due to the form of the feet, the projection results here depend heavily on the point of origin of the mesh. If it is set to the center of mass, the rendering will give a decent result for the heel and ankle, but the toes will be very distorted and thus unrecognizable.

As described, the number of artifacts in the final results varied because of the different forms of the tested structures. In general, the artifacts can be reduced by choosing a higher sampling rate, resulting in longer rendering times. The projection algorithm will still have problems with overhangs and oblong structures. Better results may be obtained by creating a topology for the sampling mesh that consists mostly of regularly spaced vertices and then use the position of these vertices to calculate the position of the camera.



(a) The texture of the head. (b) The texture of the torso.



(c) The texture of the hips. (d) The texture of a leg. (e) The texture of a foot.

Figure 5.9: The structures shown in Figure 5.1 projected onto 2D textures using the Anatomical Entertainer. The texture of the head deviates from the previous Figure 5.1, here the brain is colored in yellow and a mesh of the lower muscles of the head colored in magenta was added.

The result around the pole area of the meshes varied the most. This results from the previously described problems with the pole areas, but also because of the topology of the papercraft, which sometimes is too coarse to calculate a decent interpolation for the UV coordinates.

The results using partial meshes are still more distorted than the results for the larger anatomical structures. The current pipeline does allow the integration of the partial sampling meshes to help with inner structures, but the effort to create them is very time-intensive. The main problem is that the focal points in the border area of the focal point mesh need to be placed appropriately so the camera can capture the outer edge of

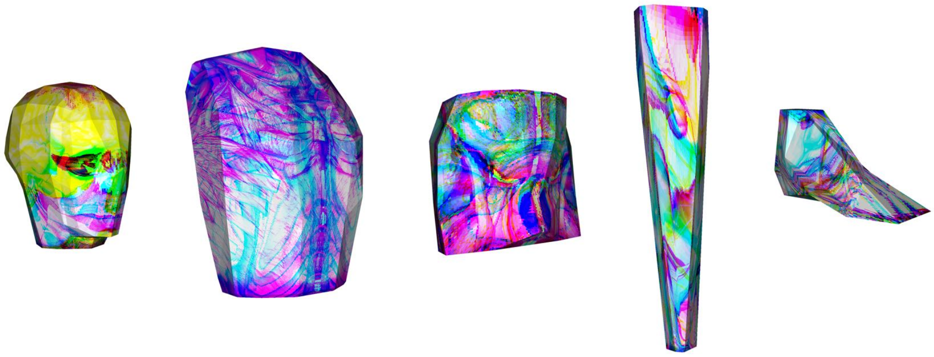


Figure 5.10: The textures from Figure 5.9 applied to the associated paper mesh.

the structure. The manual creation of these sampling meshes is not the optimal approach and an automatic creation would be better.

Three of the papercraft models tried out in this project were printed out and folded, the head, the torso and the hips. Since all of the components of these body regions differ greatly in shape and size, these meshes include a lot of different challenges for the application and worked well as test cases. Meshes with a completely different context could also be used. Generally, it is possible for the users to load new meshes into the application, but they also need to provide a sampling and for further use cases also a paper-mesh.

The craftsmanship necessary to build the papercraft was underestimated a bit. The more complex paper-meshes featured from 100 – 200 triangles and were not that easy to cut out and fold, if only unfold onto a single A4 sheet. Simplifying the meshes would be one option to tackle this problem or making bigger papercrafts out of bigger or multiple paper sheets. The Appendix 1, 2 features the templates for the papercrafts in Figure 3.10, feel free to print and fold them.

Figure 5.11: The image on the left shows several cut out prints, folded to the papercrafts shown in 3.10.

The results, in general, vary with the printing quality on hand, but as long as the right colors are selected for the meshes the visualization should change under the impression of the color filters, so that more or less only one of the rendered structures is visible. The best filtering results are achieved with the red filter. The interactive visualization on [CFT] suggests that this results from the edge position red has in the wavelength gamut of the light and thus a reduced overlap of the absorption filter with other wavelengths.

CHAPTER 6

Conclusion and Future Work

With the Anatomical Entertainer, a physical visualization application for education, the list of possible future add-ons seems endless, but the application would especially profit from efficiency improvements for a few specific components. The import of the anatomical structures as well as the color blending works well. The unfolding algorithm should be improved to enable an easier workflow and give better results. Especially the projection of inner anatomical structures could be further examined.

The conversion of the 3D mesh, to a 2D unfolded grid, as seen in Figure 5.11, so it is possible to print and fold into a papercraft, is done with the external tool Blender [ble], that already includes all of the necessary methodologies. A major improvement would be, to leave Blender out of the loop, resulting in a much faster workflow, because a lot of the steps necessary in the creation of a papercraft could be automated. In addition to the "Export Paper Model" add-on, only a few core functions of Blender were used in this thesis for pre- and postprocessing and the creation of the sampling-meshes. Unfortunately, it would have been too time-intense, to implement all of this functionality from scratch for this thesis. However, it is very conceivable to include this functionality into the application, but since it is open-source, Blender is very accessible and thus still matches the criteria of this thesis.

Some steps of the workflows, e.g., the calculation of the camera attributes are very time consuming and thus not scaling well for use cases with bigger textures or meshes, but they could be improved a lot by implementing additional functionality, like concurrency. Also, a more sophisticated approach for the image-stitching would result in faster-rendered textures, containing fewer artifacts. It would also solve the problem of the oversampling around the poles.

The proposed application eases the creation of paper-based visualizations. The additional color lenses enable a more interactive perception of the visualization, due to the color blending method. The application works well with anatomical data, but could also be used in combination with meshes that have a completely different context. It would be very interesting to try out different use cases for the created visualizations, for instance in an educational context and evaluate the benefits they may entail.

Figure 6.1: The papercrafts shown previously in Figure 3.10, combined with colored light coming from the top.

List of Figures

Bibliography

[API19] Anna Aimar, Augusto Palermo, and Bernardo Innocenti. The role of 3d printing in medical applications: A state of the art. *Journal of Healthcare Engineering*, 2019:1–10, 03 2019.

[ASS+19] Kathleen D. Ang, Faramarz F. Samavati, Samin Sabokrohiyeh, Julio Garcia, and Mohammed S. Elbaz. Physicalizing cardiac blood flow data via 3d printing. *Computers and Graphics*, 85:42 – 54, 2019.

[ble] Blender. https://www.blender.org/. Accessed: 30-03-2020.

[BP3] Bodyparts3d. https://lifesciencedb.jp/bp3d/. Accessed: 30-03-2020.

[Car] Carnovsky. https://www.carnovsky.com/. Accessed: 30-03-2020.

[CFT] Interactive color filter tutorial. https://www.olympus-lifescience.com/en/microscope-resource/primer/java/filters/gelatin/. Accessed: 30-03-2020.

[Dav17] K. Davies. *Illumanatomy: See inside the human body with your magic viewing lens*. Wide Eyed Editions, 2017.

[JD13] Y. Jansen and P. Dragicevic. An interaction model for visualizations beyond the desktop. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2396–2405, 2013.

[JDF13] Yvonne Jansen, Pierre Dragicevic, and Jean-Daniel Fekete. Evaluating the efficiency of physical visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 2593–2602, New York, NY, USA, 2013. Association for Computing Machinery.

[JDI+15] Yvonne Jansen, Pierre Dragicevic, Petra Isenberg, Jason Alexander, Abhijit Karnik, Johan Kildal, Sriram Subramanian, and Kasper Hornbæk. Opportunities and challenges for data physicalization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 3227–3236, New York, NY, USA, 2015. Association for Computing Machinery.

[Phy] Examples of physical visualization. http://dataphys.org/list/. Accessed: 20-04-2020.

[SAS+19] Faramarz F. Samavati, Kathleen D. Ang, Samin Sabokrohiyeh, Julio Garcia, and Mohammed S. Elbaz. Physicalizing cardiac blood flow data via 3d printing. *Computers and Graphics*, 85:42 – 54, 2019.

[SB17] Sergej Stoppel and Stefan Bruckner. Vol2velle: Printable interactive volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):861–870, January 2017.

[SSB15] Simon Stusak, Jeannette Schwarz, and Andreas Butz. Evaluating the memorability of physical visualizations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 3247–3250, New York, NY, USA, 2015. Association for Computing Machinery.

[SSD09] Martin Spindler, Sophie Stellmach, and Raimund Dachselt. Paperlens: Advanced magic lens interaction above the tabletop. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, page 69–76, New York, NY, USA, 2009. Association for Computing Machinery.

[SSJ+14] Saiganesh Swaminathan, Conglei Shi, Yvonne Jansen, Pierre Dragicevic, Lora A. Oehlberg, and Jean-Daniel Fekete. Supporting the design and fabrication of physical visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 3845–3854, New York, NY, USA, 2014. Association for Computing Machinery.

[Stu15] Simon Stusak. Exploring the potential of physical visualizations. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, page 437–440, New York, NY, USA, 2015. Association for Computing Machinery.

[TGK+17] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. Interactive lenses for visualization: An extended survey. *Comput. Graph. Forum*, 36(6):173–200, September 2017.

[Ven14] C Lee Ventola. Medical applications for 3d printing: Current and projected uses. *P and T : a peer-reviewed journal for formulary management*, 39, 2014.

[VTK] Visualization tool kit. https://vtk.org/. Accessed: 20-04-2020.

Appendix

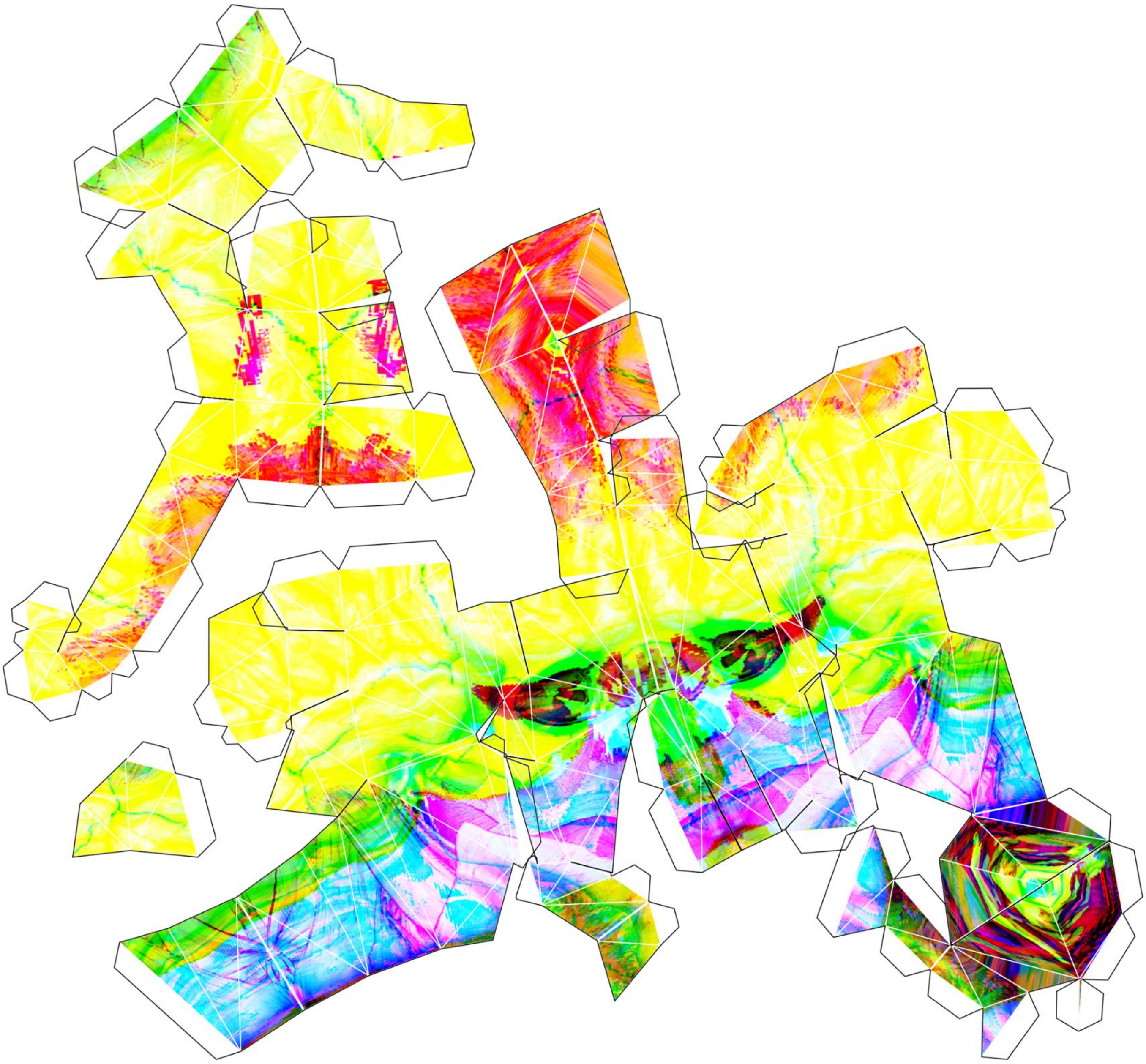


Figure 1: The template for the head physicalization.

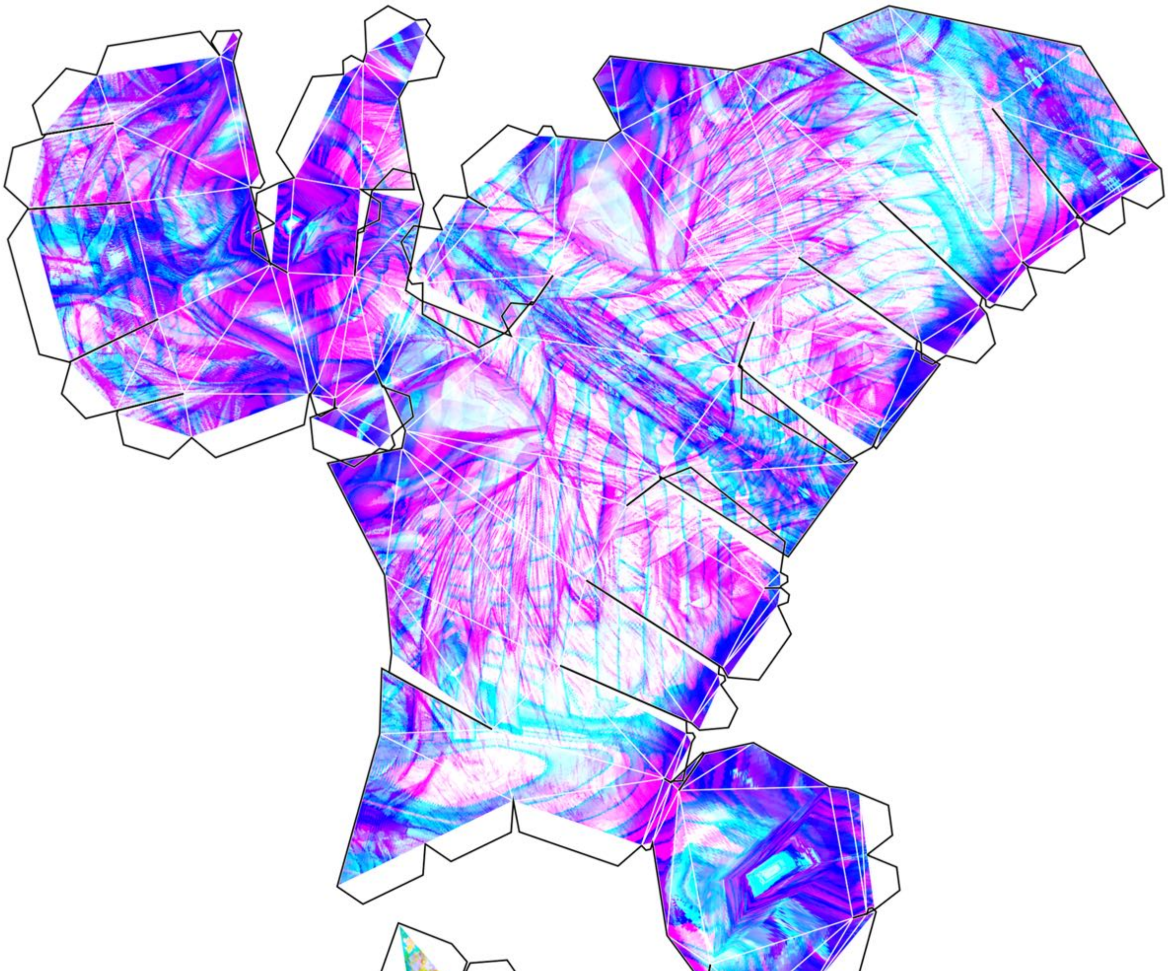


Figure 2: On the top is the template for the torso physicalization and on the bottom is the template for the physicalization of the hips.