UNIVERSITÄT
KOBLENZ · LANDAU

Fachbereich 4: Informatik

2D-View of the Mitral Valve Through Global Parameterization

Bachelorarbeit

zur Erlangung des Grades Bachelor of Science (B.Sc.)
im Studiengang Computervisualistik

vorgelegt von

Pepe Eulzer

Erstgutachter:	J.-Prof. Dr. Kai Lawonn (Institut für Computervisualistik, AG Medizinische Visualisierung)
Zweitgutachter:	Nils Lichtenberg (Institut für Computervisualistik, AG Medizinische Visualisierung)

Koblenz, im September 2018

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

	Ja	Nein
Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.	☐	☐

.....

(Ort, Datum) (Unterschrift)

Zusammenfassung

Die Mitralklappe ist eine der vier Herzklappen des Menschen. Sie befindet sich in der linken Herzkammer und agiert als ein unidirektionales Ventil, welches den Blutfluss vom linken Atrium zum linken Ventrikel steuert. Eine funktionierende Mitralklappe verhindert den Rückfluss von Blut in den Lungenkreislauf, wodurch sie einen unverzichtbaren Anteil zu einem gesunden Herzkreislauf beiträgt. Pathologien der Mitralklappe können eine Reihe von Symptomen hervorrufen, welche in ihrer Schwere von Brustschmerzen und Ermüdung bis zum Lungenödem (dem Eindringen von Flüssigkeit in die Lunge) reichen können. Im schlimmsten Fall kann dieses zum Atemversagen führen.

Dysfunktionale Mitralklappen können mithilfe komplexer chirurgischer Eingriffe wiederhergestellt werden, welche in hohem Maße von intensiver Planung und präoperativer Analyse profitieren. Visualisierungstechniken eröffnen die Möglichkeit, solche Vorbereitungsprozesse zu unterstützen und können zudem einer postoperativen Evaluation dienlich sein. Die vorliegende Arbeit erweitert die Forschung in diesem Bereich. Sie stützt sich auf patientenspezifische Segmentierungen der Mitralklappe, wie sie am Deutschen Krebsforschungszentrum entwickelt werden. Solche Segmentierungen resultieren in 3D-Modellen der Mitralklappe. Der Kern dieser Arbeit wird sich mit der Konstruktion einer 2D-Ansicht dieser Modelle befassen. Die 2D-Visualisierung wird durch Methoden der globalen Parametrisierung erzeugt, welche es erlauben, bijektive Abbildungen zwischen einem planaren Parameterraum und Oberflächen in höheren Dimensionen zu erstellen.

Eine ebene Repräsentation der Mitralklappe ermöglicht Ärzten einen unmittelbaren Blick auf deren gesamte Oberfläche, analog zu einer Karte. Dies erlaubt die Begutachtung der Fläche und Form ohne die Notwendigkeit unterschiedlicher Blickwinkel. Teile der Klappe, die in der 3D-Ansicht von Geometrie verdeckt sind, werden in der 2D-Darstellung sichtbar.

Ein weiterer Beitrag dieser Arbeit ist die Untersuchung verschiedener Visualisierungen der 3D- und 2D-Mitralklappenrepräsentationen. Merkmale der Klappe können durch Assoziation mit spezifizierten Farbschemata hervorgehoben werden. So können zum Beispiel Pathologie-Indikatoren direkt vermittelt werden.

Qualität und Wirkungsgrad der vorgestellten Methoden wurden in einer Studie am Universitätsklinikum Heidelberg evaluiert.

Abstract

The mitral valve is one of four human heart valves. It is located in the left heart and acts as a unidirectional passageway for blood between the left atrium and the left ventricle. A correctly functioning mitral valve prevents a backflow of blood into the pulmonary circulation (lungs) and thus constitutes a vital part of the cardiac cycle. Pathologies of the mitral valve can manifest in a variety of symptoms with severity ranging from chest pain and fatigue to pulmonary edema (fluid accumulation in the tissue and air space of lungs), which may ultimately cause respiratory failure.

Malfunctioning mitral valves can be restored through complex surgical interventions, which greatly benefit from intensive planning and preoperative analysis. Visualization techniques provide a possibility to enhance such preparation processes and can also facilitate post-operative evaluation. The work at hand extends current research in this field, building upon patient-specific mitral valve segmentations developed at the German Cancer Research Center, which result in triangulated 3D models of the valve surface. The core of this work will be the construction of a 2D-view of these models through global parameterization, a method that can be used to establish a bijective mapping between a planar parameter domain and a surface embedded in higher dimensions.

A flat representation of the mitral valve provides physicians with a view of the whole surface at once, similar to a map. This allows assessment of the valve's area and shape without the need for different viewing angles. Parts of the valve that are occluded by geometry in 3D become visible in 2D.

An additional contribution of this work will be the exploration of different visualizations of the 3D and 2D mitral valve representations. Features of the valve can be highlighted by associating them with specified colors, which can for instance directly convey pathology indicators.

Quality and effectiveness of the proposed methods were evaluated through a survey conducted at the Heidelberg University Hospital.

Contents

1 Introduction

This work's focus lies on the development of an interactive visualization of the mitral valve, one of the four human heart valves. It is build upon earlier research, which made it possible to create triangulated 3D models of patient-specific mitral valves, based on ultrasound imaging techniques [17]. The 3D-view of the mitral valve will be extended by a 2D-view, allowing assessment and comparison of the whole valve area at once. The transfer of the extracted valve surface into the 2D-domain will be done using global parameterization, a class of techniques applied in different mesh processing fields, which allow appending lower-dimensional coordinate systems to objects embedded in higher dimensions.

1.1 Motivation

The mitral valve plays a vital role in controlling the human heart's blood-flow. Its physiology can be impaired by different afflictions, the most common being mitral regurgitation, with an occurrence rate of 2 - 3% within the adult population [26]. Typical symptoms are respiratory distress, palpitation (a noticeably rapid, strong or irregular heartbeat) and nightly cough attacks. Depending on the severity, mitral regurgitation can have a perilous impact on a patient's physical condition.

Treatment of mitral valve defects often requires surgical intervention, with methods ranging from reconstruction to replacement of the valve. Cardiac surgery is a challenging domain, which can benefit from various visualization techniques. Pre-operative analysis of a patient's mitral valve can allow a more specific selection of the desired treatment approach. Post-operative comparison can help to evaluate the success of an intervention and facilitate improvement of clinical procedures.

This work will use already existing, patient-specific 3D models of mitral valves and propose a way of creating respective 2D-representations of these models, which could provide a clear overview of the valve surface and additional indicators during valve analysis. The benefits of a 2D-view can be derived from the world-map metaphor: in some aspects a map is easier to assess than a globe, as it is possible to see the whole surface at one glance.

1.2 Content of this Work

This thesis can be roughly partitioned into a presentation of relevant background information, a description of the proposed pipeline to create the 2D-view and a review of conducted evaluations. Chapter 2 will introduce necessary medical background, including heart function and mitral valve pathologies. Chapter 3 will describe the technical background, providing an introduction to surface parameterization and the underlying concepts

of differential geometry. Chapter 4 will then discuss more closely related work, featuring the use of surface parameterization in context of medical visualization. It will also reference work done in the light of mitral valve segmentation, which portrays a direct precursor of this thesis.

Chapters 5 and 6 will comprise the main part of this work. First, the concept of the developed 2D-view will be demonstrated, independent from a specific implementation. This will include an assessment of objectives, followed by a step-by-step description of the parameterization method and an overview of different visualization options. Chapter 6 will then provide further detail of the developed software prototype.

Chapter 7 will discuss an evaluation of this work, featuring a small survey conducted at the department of Cardiac Surgery at Heidelberg University Clinic. A final conclusion and summary will be provided in Chapter 8.

2 Medical Background

This chapter focuses on anatomical and physiological background information, which is based on [52] [59].

The human heart consists of four chambers, two of which form what is called the *right heart* and the other two respectively create the *left heart*. Each half includes an atrium, where blood enters the heart, and a ventricle, from where blood is pushed out of the heart. The two halves are separated. On the right side oxygen-poor blood is received in the atrium and is then passed to the right ventricle when the atrium contracts. Simultaneously, the left atrium receives oxygen-rich blood from the lungs, which is passed on to the left ventricle through contraction. This state is called *diastole*. When the atria relax again, both ventricles start a contraction-phase, forcing blood to leave the heart. The right ventricle pushes oxygen-poor blood into the lung cycle, while the left ventricle supplies the rest of the body with oxygen-rich blood. This state is called *systole*. Figure 1 depicts the two states.

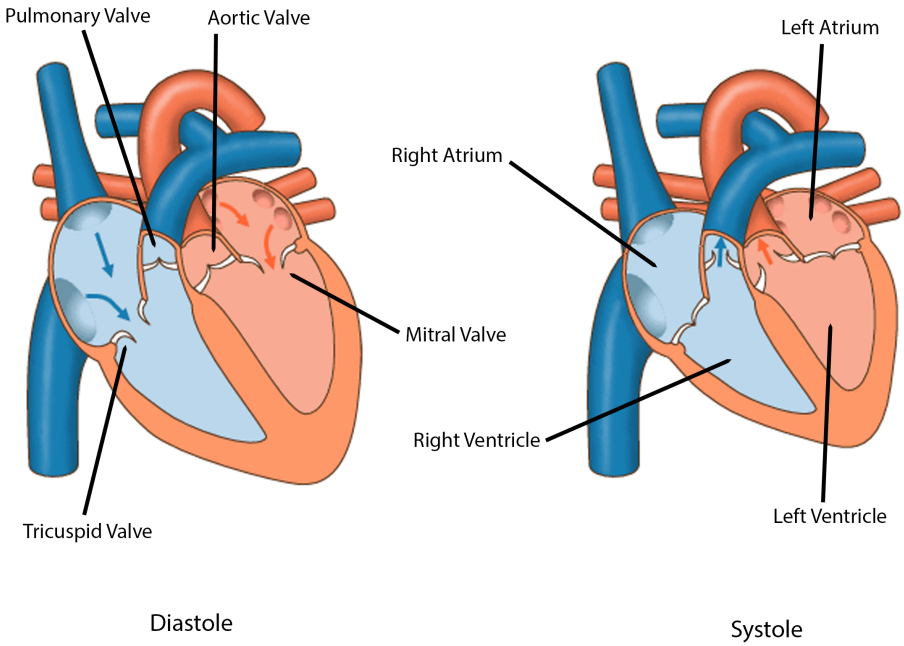

Figure 1: Blood flow diagram showing the four heart valves during diastole (left) and systole (right). Image based on [66].

To direct the blood flow the heart makes use of four valves, which are located between the atria and ventricles and at the start of the two outgoing arteries. The term *valve* originates from the fact that they work like mechan-

ical unidirectional valves. If functional, they allow blood to be pushed into a single direction only. To differentiate the four heart valves, they are usually grouped into two categories:

- **Atrioventricular valves.** These are the *mitral valve* in the left heart and *tricuspid valve* in the right heart. They control blood flow from the atria to the ventricles. During ventricle-contraction (systole) they prevent the blood from leaking back into the atria. During ventricle-relaxation (diastole) they open to let blood flow from the atria into the ventricles.

- **Semilunar valves.** This category contains the *aortic valve* (left heart) and *pulmonary valve* (right heart). They control blood flow from the ventricles into the body. During systole they open to let blood be pushed into lungs and body. During diastole they close to impede a backflow of blood into the ventricles.

2.1 Mitral Valve Function

Since this work focuses on the mitral valve, the specific physiological functions of this valve are further examined. The mitral valve controls the blood flow from the left atrium to the left ventricle. During diastole the open mitral valve lets oxygenated blood enter the left ventricle. The blood is then pushed into the body through the aortic valve (systole), while the closed mitral valve prevents a backflow into the left atrium.

The mitral valve consists of two leaflets: *cuspis anterior* and *cuspis posterior*. Therefore it is also referred to as *bicuspid valve*. The leaflets are highlighted on the right side in Figure 2. The origin of the name mitral valve stems from the form of the two leaflets: during their opened state they resemble a bishops mitre (a kind of hat with a central fold). The leaflets are held by the *mitral annulus*, a ring-like structure that connects the two parts of the valve and attaches them to the cardiac skeleton. During systole the annulus supports full closure of the mitral valve by contracting and reducing its diameter. On the bottom side, the leaflets are connected to the left ventricle walls by *chordae tendineae*. These are inelastic tendons held by papillary muscles at the other end (see Figure 2). They prevent the valve from prolapsing (opening in the wrong direction) and thus are essential to the proper closing of the valve during systole. The papillary muscles contract simultaneously with the left ventricle, in order to hold the mitral valve's leaflets in place.

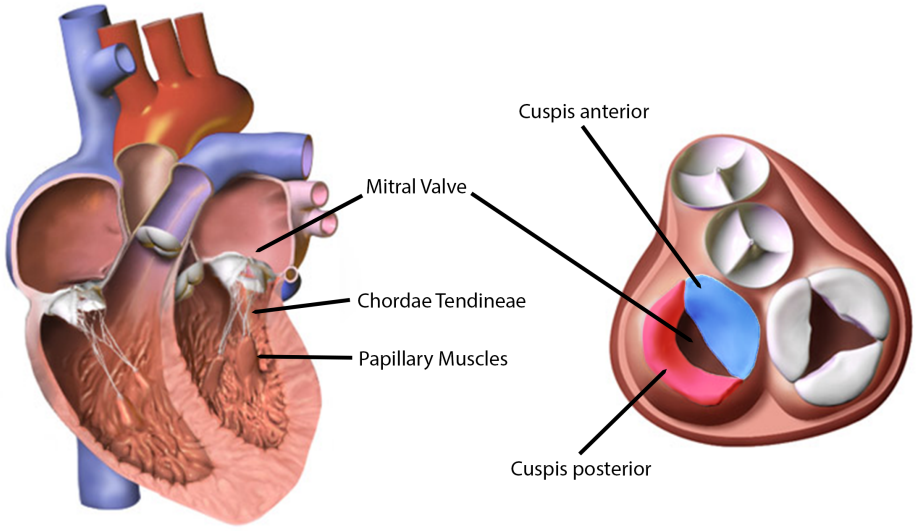

Figure 2: Left: Side-view of the heart valves during systole. The mitral valve is visibly held by chordae tendineae and papillary muscles. Right: Top-view of the valve layer. Anterior (blue) and posterior (red) leaflets of the mitral valve are marked. Image based on [65].

2.2 Mitral Valve Pathologies

This section gives a brief overview of possible mitral valve diseases and is based on [4] [14] [26] [59]. Physiological blood flow is prohibited if closing or opening of the mitral valve are in some way restricted. Depending on the severity of the affliction, implications can range from beeing non-noticeable to lethal. Pathologies of the mitral valve are grouped into the following categories:

Mitral stenosis. This describes a narrowing of the valve, where it does not sufficiently open anymore. Therefore, blood flow is restricted and the left atrium is exposed to increased pressure. This can lead to a lung edema. Mitral stenosis usually has bacterial causes, which can be treated by antibiotics. Therefore, stenosis has become an uncommon condition in most industrialized nations [59]. Possible therapeutic measures include opening a balloon-endoscope to widen or separate the leaflets.

Mitral regurgitation. This is the most frequent form of mitral valve dysfunction. When the mitral valve does not close completely a non-physiological backflow of blood into the left atrium can develop during systole. This state is called mitral regurgitation or insufficiency. The mitral valve is the heart valve most commonly affected by this affliction, as it is the one that has to withstand the highest pressure. During systole the left ventricle contracts,

pumping blood through the systemic circulation (whole body) with only the mitral valve stopping the liquid from beeing pushed back into the left atrium and pulmonary circulation (lungs).

Mitral regurgitation is commonly divided into acute and chronic. Acute regurgitation can for instance be caused by sudden rupture of papillary muscles due to a heart attack. The increase in pressure build up by blood being pushed back into the left atrium and the connected pulmonary circulation often results in acute dyspnea (respiratory distress, difficulty in breathing) with lung edema (liquid leaking into the lungs). In many acute cases immediate surgery is inevitable. However, if the dysfunction develops chronically, the myocardium (heart muscle tissue) of the left atrium adapts to the increased pressure through hypertrophy (enlargement). Patients with chronic regurgitation therefore sometimes only have slight symptoms that gradually develop. Nevertheless the pulmonary circulation is exposed to raised blood pressure levels, making surgery a conceivable option.

Figure 3: Endoscopic image of a mitral valve during an annuloplasty surgery. The artificial ring can be seen supporting the leaflets. Image from [67].

Mitral regurgitation is the second-most-often surgically treated heart valve defect [14]. About 300,000 people undergo cardiac surgery for mitral valve repair every year [8]. Procedures include replacement (mechanical or biological valve prosthesis) as well as reconstruction of the mitral valve, where the latter is preferred if possible. Reconstructive measures commonly involve annuloplasty, a technique where a coaptation assisting device is implanted to strengthen the annulus (see Figure 3). Additionally, artificial strings can be attached to the leaflets to substitute for ruptured papillary muscles. This complex surgical procedure requires thorough planning and preparation, a process which can be facilitated by visualization techniques. Among others, the thesis at hand was derived from this context.

Mitral prolapse. If the chordae tendineae are elongated or rupture, one or both of the leaflets can prolapse (flip) into the left atrium during systole (see Figure 4). This can lead to a backflow of blood into the left atrium (mitral regurgitation). About 5% of the population are at some point affected by mitral prolapse [4]. However, not in all of these cases do patients suffer from symptoms and the severity of a possible regurgitation needs to be assessed before composing further steps.

Combination of the above. Stenosis and regurgitation can occur simultaneously. Then the valve stays in a near-center position and neither opens fully nor closes completely during a cardiac circle.

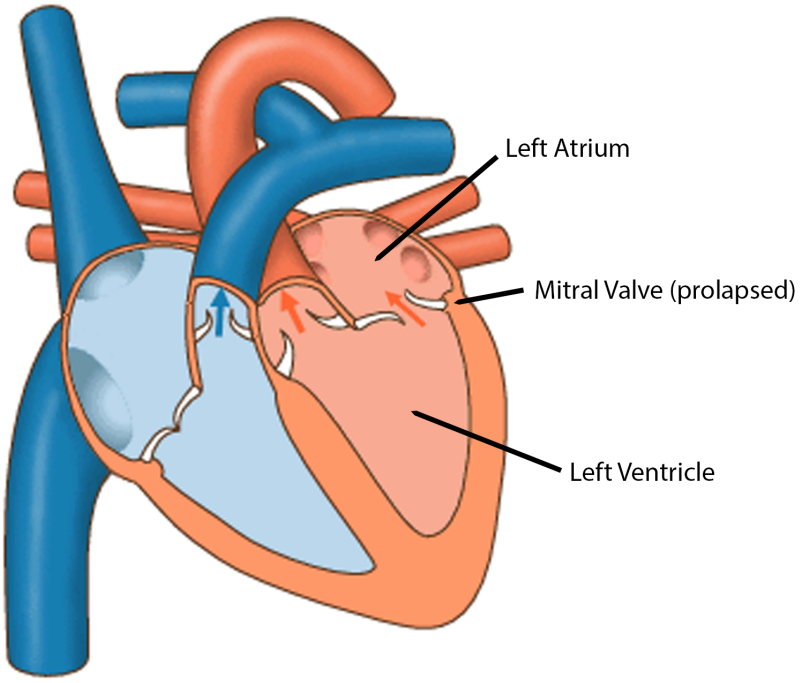

Figure 4: Blood flow diagram of the heart during systole with mitral valve prolapse. Blood is leaking back into the left atrium (mitral regurgitation). Image based on [66].

3 Surface Parameterization

Visualizations of the mitral valve can help in the process of analyzing pathological indicators and can facilitate the planning and evaluation of possible treatments. The core of this work will be the construction of a 2D-view that can convey meaningful information to a physician at a glance. The main tool that will be used to create such a perspective will be *parameterization*. Therefore this chapter will cover the technical background related to parameterization methods. It will rely on concepts from differential geometry, which will be quickly reviewed in section 3.2. To give a better understanding of what parameterization is, some typical use-cases will be presented in section 3.1.

Surface parameterization is an established term used to describe the creation of bijective mappings between surfaces, which can be embedded in distinct dimensions. If at least one of these surfaces is a polygon mesh, computing such a mapping is called *mesh parameterization* [29]. Usually, the expression refers to mappings between 3D and 2D euclidean spaces. In this scenario, the latter contains a subset labeled *parameter domain*. Coordinates on the surface of a 3D mesh can then be uniquely described with only two parameter coordinates. Each point in the parameter domain can be distinctly mapped onto the 3D surface and back (hence the bijectivity property). This is a useful tool that has proven valuable for many mesh processing applications, such as texture-mapping, remeshing or shape-analysis.

The introduction of surface parameterization into computer graphics was done in the light of mapping 2D textures onto surfaces [5] [42]. The first approaches of defining general mapping methods (like [19]) were based on earlier research on graph theory. Commonly, the work of Tutte [60] is referred to as a basis for the development of parameterization methods. Today, surface parameterization is a broad field of research. Different kinds of algorithms have been developed to deal with a diverse range of objects. Often, surfaces are cut in a way that the emerging patches are open genus-zero surfaces, which means they have a disk-like topology and can be mapped or flattened onto a plane (e.g., in [55]). Genus-zero surfaces, which topologically resemble ellipsoids, are sometimes also mapped onto a sphere. Similar approaches have been developed for genus-one (torus) or higher-genus surfaces, in order to avoid cutting. Mesh parameterization methods, which do not require dividing the surface into patches, are usually called *global* parameterization methods [28] [48]. These methods provide several benefits, for instance, they are capable of seamlessly mapping a texture onto an object. Open genus-zero objects, like the one targeted by this work, can be globally parameterized in the planar domain.

3.1 Goals and Applications

A ubiquitous amount of parameterization applications emerged during the past 20 years, spanning a vast number of mesh processing fields. Moreover, other domains like technical modeling or medical visualization make use of parameterization methods as well. The following listing gives an introductory overview of different application fields (based on [29]):

- **Texture mapping.** Textures are usually created in the 2D domain. In order to display the texture on a surface within the 3D domain, each of the surfaces point coordinates are assigned a *texture coordinate*. These 2D coordinates, which are also commonly called uv-coordinates, are placed on top of the texture. Their location defines the pixel values of points on the 3D surface. Points on the surface are triangulated by barycentric weights applied to the three uv-coordinates in 2D that correspond to the triangle the point lies in. The closest texture values can then be interpolated. This means that every point on the surface can be mapped to a uniquely determined point in the parameter space. Thus, a parameterization is defined. Research on texture mapping often focuses on minimizing texture stretch or deviation [50] but a wider range of problems has been adressed as well, e.g., reducing memory requirements [49].

- **Detail mapping.** In more general terms, texture mapping belongs to a class of techniques called detail mapping. Apart from texture values defining the color of a surface, different kinds of 2D maps can store information on the object's appearance. By using bump, normal or displacement maps, surface normals can be stored with a high resolution. They define the structure of an object and can be saved in a 2D array, which is later accessed during rendering to evaluate the lighting model. This can dramatically reduce the number of polygons required to display highly detailed objects.

- **Morphing.** In order to transform a surface into a different shape (e.g., during animation) a mapping between key-frame objects can be used to compute an interpolation at specific intervals. In these cases a parameterization is not computed to map to a lower-dimensional domain, but rather to create a map between two surfaces in the same domain.

- **Mesh editing.** Local parameterizations can be used to facilitate editing operations. In [38] Lèvy creates a planar parameterization of input models and uses it, i.a., to smooth the existing geometry while minimizing introduced distortions.

- **Remeshing.** Since polygon meshes are discrete approximations of the surfaces they represent, the same shape can be displayed by different triangulations, while preserving an analogous accuracy. Some applications require specific topological designs. For instance, numerical simulations benefit from triangles with a uniformly distributed aspect ratio. Sometimes, instead of triangles, quadrangles are preferred [57]. Remeshing of surfaces describes the act of replacing one triangulation by another. To achieve this, commonly a parameterization is constructed and then a desirable triangulation is mapped back onto the surface. Typical problems arise if the parameterization introduces too much distortion or requires cutting the surface, which usually manifests in visible discontinuities.

- **Compression.** Another application of remeshing is mesh compression. As lower data entropy corresponds with a higher compression rate, regular meshes are the preferred option for storing and transmitting geometric models. Regularity can be increased by remeshing with periodic sampling patterns.

- **Mesh completion.** Meshes, which are automatically generated from 3D scans, often suffer from holes. Techniques, such as those described in [38], use planar parameterization to obtain the shape of hole boundaries and construct a triangulation.

- **Sheets to object modeling.** Planar mesh parameterization is an important tool outside of the pure mesh processing domain. When modeling 3D objects from material sheets, e.g., during metal forming or forging [32], planar patterns are computed which will later form the desired shapes. In order to achieve this, computer models of the objects are first created and segmented into almost developable charts, which are then parameterized in the the 2D domain.

- **Medical visualization.** In pursuance of simpler and clearer visualization of complex medical geometric data sets, a mapping to a canonical domain can be constructed. This requires computing parameterizations of the polygon meshes. Applications and research in this area are separately examined in section 4.1.

3.2 Differential Geometry Background

Methods derived from differential geometry provide a baseline of tools used in various mesh processing algorithms, including parameterization. Therefore, this section reviews some fundamental concepts of differential geometry, required to understand the following chapters. The methods discussed here follow the descriptive approaches of Botsch et al. [9] and

Hormann et al. [29]. In order to stay within the scope of this section, proofs and in-depth discussion are omitted.

An intuitive way to understand parameterization is to start with parametric surfaces. In general, a continuous 3D surface $S \subset \mathbb{R}^3$ can be expressed in parametric form as

$$\mathbf{x}(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}, \quad (u,v) \in \Omega, \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is called *parameter domain* and x, y, z denote functions that map a coordinate (u,v) from the parameter space onto the surface. These functions need to be differentiable in u and v. An illustrative example of this surface representation is a globe (surface) and a world map (parameter domain). Each point on the map can be uniquely mapped to a point on the globe and back. The parametric equation can in this case be expressed using spherical coordinates $\theta \in [0, 2\pi]$ and $\phi \in [-\pi/2, \pi/2]$:

$$\mathbf{x}(\theta,\phi) = \begin{pmatrix} x(\theta,\phi) \\ y(\theta,\phi) \\ z(\theta,\phi) \end{pmatrix} = \begin{pmatrix} R\cos\theta\cos\phi \\ R\sin\theta\cos\phi \\ R\sin\phi \end{pmatrix}, \tag{2}$$

where R is the radius of the sphere. This equation provides a way of transforming a rectangle $[0, 2\pi] \times [-\pi/2, \pi/2]$ into a sphere (see Figure 5). Each vertical line on the map corresponds with a longitude, a straight pole-to-pole line, on the globe. As these lines are defined by constant values of θ, they are also called *iso-θ curves*. Analogously, lines with constant ϕ value are referred to as *iso-ϕ curves* and are mapped to the latitudes of the globe, which are lines parallel to the equator.

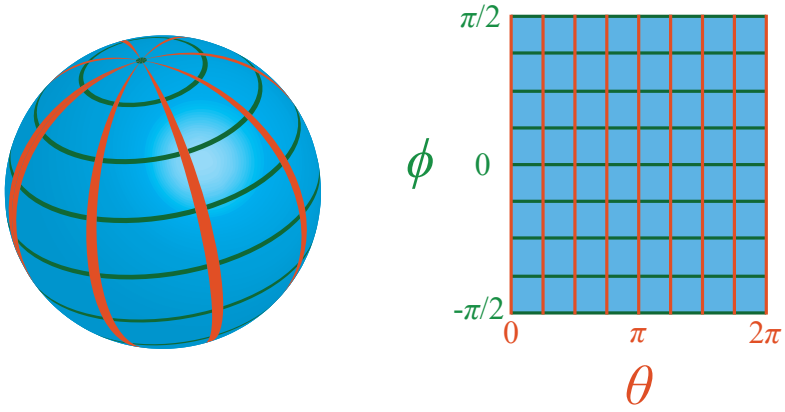

Figure 5: A possibility to map a rectangle to a sphere.

On the left side of Figure 5 it can be observed, that the relative distance between iso-θ curves on the surface converges to zero as they approach the poles. This means the map is *distorted* when applied to the surface. In order to quantify these distortions, the metric properties of a parametric surface $S \subset \mathbb{R}^3$ can be examined. The metric of S is determined by the first derivatives of $\mathbf{x}$, which are defined by point-wise partial derivatives:

$$\mathbf{x}_u(u_0, v_0) := \frac{\partial \mathbf{x}}{\partial u}(u_0, v_0), \quad \mathbf{x}_v(u_0, v_0) := \frac{\partial \mathbf{x}}{\partial v}(u_0, v_0). \tag{3}$$

On the surface, the partial derivatives correspond to the tangent vectors of the iso-parameter curves they lie on. At the point $\mathbf{x}(u_0, v_0) \in S$ these would be

$$\mathbf{C_u}(t) = \mathbf{x}(u_0 + t, v_0), \quad \mathbf{C_v}(t) = \mathbf{x}(u_0, v_0 + t). \tag{4}$$

For simplification of the following formulas the parameters (u_0, v_0) or (u, v) will be excluded. For instance, the partial derivative in u-direction will simply be referred to as $\mathbf{x}_u$. However, one should keep in mind that differential geometry properties are defined per point, because they typically vary across the surface.

Since the partial derivatives translate into two tangent vectors on the surface, the so called *surface normal* can be constructed from them:

$$\mathbf{n} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{||\mathbf{x}_u \times \mathbf{x}_v||}. \tag{5}$$

For this, the parameterization needs to be regular, i.e., the tangent vectors need to be linearly independent. Given a direction vector $\bar{\mathbf{w}} = (u_w, v_w)^T \in \Omega$, a directional derivative at (u_0, v_0) can be determined. The straight line spanned from this point by $\bar{\mathbf{w}}$ in Ω and parameterized by t corresponds to the following curve on the surface:

$$\mathbf{C_w}(t) = \mathbf{x}(u_0 + tu_w, v_0 + tv_w). \tag{6}$$

This means that the directional derivative $\mathbf{w}(u_0, v_0)$ of $\mathbf{x}(u_0, v_0)$ is also the tangent to $\mathbf{C_w}(0)$. Arbitrary directional derivatives can be mapped with help of the *Jacobian* $\mathbf{J}$, a matrix that lists the partial derivatives as column vectors:

$$\mathbf{J} = [\mathbf{x}_u, \mathbf{x}_v] = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{bmatrix}. \tag{7}$$

A directional derivative $\bar{\mathbf{w}}$ can then be mapped from the parameter domain to the corresponding surface tangent $\mathbf{w}$ via $\mathbf{w} = \mathbf{J}\bar{\mathbf{w}}$.

The Jacobian is also useful for measuring how specific surface characteristics change due to the mapping transformation. For this purpose a second, closely related tool is commonly used: the *first fundamental form* $\mathbf{I}$.

The first fundamental form of $\mathbf{x}$ is the matrix product of the Jacobian with its transpose:

$$\mathbf{I} = \mathbf{J}^T\mathbf{J} = \begin{bmatrix} E & F \\ F & G \end{bmatrix} := \begin{bmatrix} \mathbf{x}_u^T\mathbf{x}_u & \mathbf{x}_u^T\mathbf{x}_v \\ \mathbf{x}_u^T\mathbf{x}_v & \mathbf{x}_v^T\mathbf{x}_v \end{bmatrix} \tag{8}$$

The importance of this relationship gets clear if we look at how angles are distorted when the mapping is applied. If we denote $\bar{\mathbf{w}}_1$ and $\bar{\mathbf{w}}_2$ as two unit direction vectors in parameter space, the scalar product $\langle \bar{\mathbf{w}}_1, \bar{\mathbf{w}}_2 \rangle$ gives us the cosine of the enclosed angle. Using $\mathbf{w} = \mathbf{J}\bar{\mathbf{w}}$ we can also compute the scalar product between the corresponding tangent vectors on the surface:

$$\langle \mathbf{w}_1, \mathbf{w}_2 \rangle = \bar{\mathbf{w}}_1^T\bar{\mathbf{w}}_2 = (\mathbf{J}\bar{\mathbf{w}}_1)^T(\mathbf{J}\bar{\mathbf{w}}_2) = \bar{\mathbf{w}}_1^T(\mathbf{J}^T\mathbf{J})\bar{\mathbf{w}}_2 = \bar{\mathbf{w}}_1^T\mathbf{I}\bar{\mathbf{w}}_2 \tag{9}$$

Similarly, the first fundamental form offers the ability to measure the length of a tangent vector $\mathbf{w}$ as

$$||\mathbf{w}|| = \sqrt{\bar{\mathbf{w}}^T\mathbf{I}\bar{\mathbf{w}}}. \tag{10}$$

This property can be extended to allow measuring of lengths of curves and areas on the surface.

The first fundamental form is also sometimes called the *metric tensor*. In general, mathematical tensors can be understood as mappings that transform circles into ellipses. If we further examine the first fundamental form, we can observe that this property holds true for $\mathbf{I}$ as well. Given a point (u_0, v_0) in parameter space we can construct unit vectors in all possible directions, which will therefore define a circle around the point. Mapping all of these vectors to the surface (with help of the Jacobian) will create an ellipse around $\mathbf{x}(u_0, v_0)$ that lies in the tangential plane. This ellipse is called the *anisotropy ellipse* and holds information on the distortion of the mapping. A singular value decomposition of $\mathbf{J}$ reveals the linear transformations that occur during the mapping:

$$\mathbf{J} = U\Sigma V^T = U \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{pmatrix} V^T. \tag{11}$$

It can be observed that

- Points on the circle are rotated around (u_0, v_0) by V^T. The singular vectors are aligned with the u and v-axes.
- The circle is scaled by Σ with factor σ_1 in u and factor σ_2 in v direction.
- U maps the unit vectors into the tangent plane at $\mathbf{x}(u_0, v_0)$.

This means that all information about the amount of the distortion can be found in the singular values σ_1 and σ_2. It can be shown that if they are identical, the parameterization results in no angle-distortion and if

$\sigma_1 \cdot \sigma_2 = 1$ the transformation is area-preserving. The computation can be simplified by making use of the first fundamental form, as singular values of any matrix A are the square roots of the eigenvalues of $A^T A$. The eigenvalues of $\mathbf{I}$ can be found using

$$\lambda_{1,2} = \frac{1}{2}(E + G) \pm \sqrt{4F^2 + (E - G)^2}. \tag{12}$$

where E, F, G denote the coefficients of $\mathbf{I}$ from Equation (8). The eigenvalues $\lambda_{1,2}$ correspond with eigenvectors $\bar{\mathbf{e}}_{1,2} \in \Omega$ of $\mathbf{I}$ in parameter space. On the surface, their mapped tangents $\mathbf{e}_1 = \mathbf{J}\bar{\mathbf{e}}_1$ and $\mathbf{e}_2 = \mathbf{J}\bar{\mathbf{e}}_2$ are the axes of the anisotropy ellipse with lengths $\sigma_1 = \sqrt{\lambda_1}$ and $\sigma_2 = \sqrt{\lambda_2}$. Knowing this, the nature of a specific parameterization can be quantified by measuring introduced distortions, which can be broken down into the properties of all point-wise defined anisotropy ellipses.

Formulas (1) to (10) are adapted from [9] and formulas (11) and (12) are based on [29].

3.3 Parameterization of Triangulated Surfaces

In computer graphics, 3D objects are commonly represented by *triangle meshes*. A triangle mesh $S_\mathcal{T}$ can be defined as the union of a set of triangles or faces $\mathcal{T} = \{T_1, \ldots, T_m\}$. The triangles in $\mathcal{T}$ may only intersect at common edges and vertices. Each vertex is a point $\mathbf{p} = (x, y, z)$ in $\mathbb{R}^3$. The set of vertices $\mathcal{V} = \{\mathbf{p}_1, \ldots, \mathbf{p}_{n+b}\}$ can be split into a set of *interior vertices* $\mathcal{V}_I = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ and a set of *boundary vertices* $\mathcal{V}_B = \{\mathbf{p}_{n+1}, \ldots, \mathbf{p}_{n+b}\}$. A vertex is labeled as a boundary vertex if it is the intersection of two edges that lie on the boundary, which means they are only part of one triangle in $\mathcal{T}$. Furthermore, two distinct vertices $\mathbf{p}_i, \mathbf{p}_j \in \mathcal{V}$ are classified as *neighbors*, if they lie on opposing ends of a shared edge. For any $\mathbf{p}_i \in \mathcal{V}$ the set N_i consists of the indices of all neighbors of $\mathbf{p}_i$ (definition based on [29]).

Computing a parameterization of such a configuration proves to be a non-trivial problem. The first thing to note is that in contrast to parametric surfaces described in section 3.2, where a mapping was declared from the parameter domain Ω to the surface S, in this scenario the mapping needs to be constructed the other way around. This is because the surface $S_\mathcal{T}$ is already well defined and the parameter domain is not. As a triangulated surface consists of piecewise linear patches, a mapping is usually build using piecewise linear functions, each one corresponding to a specific triangle on the surface. These linear functions are defined by associating each surface vertex $\mathbf{p}_i = (x_i, y_i, z_i)$ with a distinct coordinate $\mathbf{q}_i = (u_i, v_i) \in \Omega$, where the parameter domain $\Omega \subset \mathbb{R}^2$ is the union of all parameter triangles. The coordinates $\mathbf{q}_i$ are generally referred to as uv-coordinates. It is therefore possible to determine a parameterization of a triangulated surface by providing a complete set of uv-coordinates (one for each object ver-

tex). Additionally, a valid parameterization is required to follow some constraints, demanding the uv-coordinates not to be setup in a way that the surface image self-intersects. This means the intersection of any two triangles is always a common edge, a common vertex or empty. Then, at a point $(u, v) \in \Omega$ the parameterization $\mathbf{x}$ is given as

$$\mathbf{x}(u, v) = \alpha \mathbf{p}_i + \beta \mathbf{p}_j + \gamma \mathbf{p}_k, \tag{13}$$

where the point (u, v) needs to be inside of the triangle $[q_i, q_j, q_k] \subset \Omega$ corresponding to the triangle $[p_i, p_j, p_k] \subset S_{\mathcal{T}}$. The triplet (α, β, γ) consists of the barycentric coordinates of the point (u, v) inside the triangle in parameter space. Figure 6 provides an example of how such a mapping can appear.

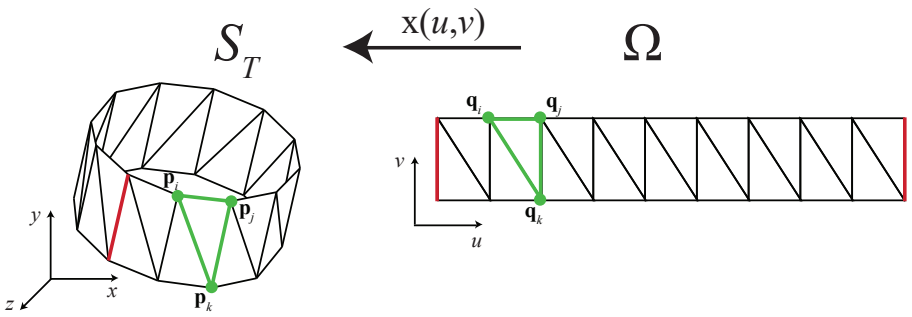

Figure 6: Visualization of a parameterization. A triangle on the surface (left) is mapped to a triangle in parameter space (right). The coordinates define the piecewise linear function (13) that bijectively maps points from parameter space Ω onto the surface $S_{\mathcal{T}}$.

Practically all parameterization algorithms follow this approach and try to construct a set of valid uv-coordinates from a given triangulated surface. Still, the question remains how such a valid configuration can be computed. In the following, different methods will be outlined that provide a solution to this problem. From differential geometry we know that:

1. For any surface different parameterizations exist.
2. For most surfaces a parameterization will inevitably introduce distortions of some kind.

It is not determined which parameterization is *the best* parameterization. How the quality of a parameterization is measured always depends on the context of the use-case or desired outcome. Most of the methods presented now will introduce their own constraints that define what an "optimal" parameterization is. The goal of this work will then be to construct a method that provides the most benefits in the targeted utilization.

3.3.1 Barycentric Mapping

Barycentric mapping is a category of widely used parameterization methods, which can be viewed as physical spring-relaxation models. Imagine a triangulated disk-like surface, where the edges are springs connecting the vertices. If all of the boundary vertices $\mathcal{V}_B$ are fixed in the plane, following a convex shape, the interior springs will relax in the plane as well, coming to rest in the energetically most efficient position. This allows taking the resting vertex positions and assigning them to valid uv-coordinates.

Let's assume the boundary vertices have been set to a valid convex 2D configuration (for instance around a square in an equidistal distribution). We consider all edges to be ideal springs, which have a rest length of 0 and potential energy equal to $\frac{1}{2}Ds^2$, where s is the current length and D is the spring constant. Following the approach of Hormann et al. [29] and the mesh configuration described in section 3.3, the overall spring energy to be minimized can be described as

$$E = \frac{1}{2}\sum_{i=1}^{n+b}\sum_{j\in N_i}\frac{1}{2}D_{ij}||\mathbf{q}_i - \mathbf{q}_j||^2, \tag{14}$$

where $D_{ij} = D_{ji}$ is the spring constant between $\mathbf{p}_i$ and $\mathbf{p}_j$. The points $\mathbf{q}_i = (u_i, v_i)$ are the corresponding parameter points to $\mathbf{p}_i$, where the interior point locations in Ω with $i = 0, \ldots, n$ are currently unknown. Hence, to minimize E we examine its partial derivative with respect to $\mathbf{q}_i$:

$$\frac{\partial E}{\partial \mathbf{q}_i} = \sum_{j\in N_i} D_{ij}(\mathbf{q}_i - \mathbf{q}_j), \tag{15}$$

which is 0 when

$$\forall i \in \{1, \ldots, n\} : \quad \sum_{j\in N_i} D_{ij}\mathbf{q}_i = \sum_{j\in N_i} D_{ij}\mathbf{q}_j \tag{16}$$

and thus the minimum of E is obtained (for positive values of D_{ij}). From this one can derive an equivalent assertion by interpreting interior parameter points as affine combinations of their neighbors. This can be formally expressed as

$$\forall i \in \{1, \ldots, n\} : \quad \mathbf{q}_i = \sum_{j\in N_i} \lambda_{ij}\mathbf{q}_j, \tag{17}$$

where λ_{ij} are the so called *barycentric weights*, which are simply the normalized spring constants

$$\lambda_{ij} = \frac{D_{ij}}{\sum_{k\in N_i} D_{ik}}. \tag{18}$$

This particular point of view is important, as it draws a parallel to Tutte's barycentric mapping theorem [60], which states that a valid parameterization can be obtained if:

1. A triangulated surface homeomorphic to a disk is given.

2. The boundary uv-coordinates are mapped to a convex polygon.

3. The coordinates of the internal uv-coordinates are a convex combination of their neighbors.

The third statement corresponds to Equation (17). This theorem is the basis of practically all barycentric parameterization methods and has been proven through various approaches, e.g., by Tutte [60] and Gortler et al. [22]. The first time a parameterization was constructed based on this theorem goes back to Floater [19].

Equation (17) can be solved by first expanding the sum on the right hand side in order to separate interior from exterior vertices:

$$\mathbf{q}_i - \sum_{j \in N_i, j \leq n} \lambda_{ij} \mathbf{q}_j = \sum_{j \in N_i, j > n} \lambda_{ij} \mathbf{q}_j. \tag{19}$$

This shows that the unknown interior parameter points $\mathbf{q}_i = (u_i, v_i)$ can be found by solving the linear systems

$$A \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} \bar{u}_1 \\ \vdots \\ \bar{u}_n \end{pmatrix} \quad \text{and} \quad A \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} \bar{v}_1 \\ \vdots \\ \bar{v}_n \end{pmatrix}, \tag{20}$$

where

$$\bar{u}_i = \sum_{j \in N_i, j > n} \lambda_{ij} u_j \quad \text{and} \quad \bar{v}_i = \sum_{j \in N_i, j > n} \lambda_{ij} v_j. \tag{21}$$

The $n \times n$ matrix $A = (a_{ij})_{i,j=1,\ldots,n}$ contains the barycentric weights accordingly:

$$a_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -\lambda_{ij} & \text{if } j \in N_i, \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

The linear system (20) can be solved in various ways. For large meshes sparse iterative or direct methods are the preferred option. Equations (14) to (22) are adapted from [29].

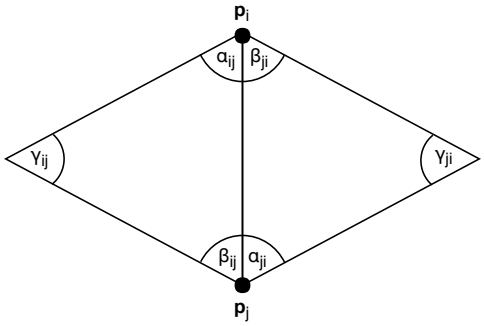

Figure 7: Angles used for the construction of barycentric coordinates.

The spring constants D_{ij}, or in general the normalized coefficients λ_{ij}, can be chosen in different ways. If a vertex is only connected to three neighbors, the barycentric weights are uniquely defined. For more than three neighbors however, a way of constructing these weights needs to be implemented. The simple solution of constant normalized values results in a valid configuration [60], but it also leads to a high amount of introduced distortion, as the mesh geometry is not taken into account. Therefore the popular choice is to compute homogeneous coordinates w_{ij}, which depend on the angles and distances of the mesh as denoted in Figure 7. Some of the proposed methods to construct w_{ij} are:

- *Wachspress coordinates* [62], which have been used for parameterization by Desbrun et al. [45]:

$$w_{ij} = \frac{\cot \alpha_{ji} + \cot \beta_{ij}}{{r_{ij}}^2}. \tag{23}$$

- *Discrete harmonic coordinates*, which were used for parameterization purposes, for instance, by Eck et al. [16]:

$$w_{ij} = \cot \gamma_{ij} + \cot \gamma_{ji} \tag{24}$$

- *Mean value coordinates*, which are based on the discretization of the mean value theorem. These coordinates were introduced by Floater [20]:

$$w_{ij} = \frac{\tan \frac{a_{ij}}{2} + \tan \frac{\beta_{ji}}{2}}{r_{ij}} \tag{25}$$

In all cases the calculated weights are normalized through

$$\lambda_{ij} = \frac{w_{ij}}{\sum_{k \in N_i} w_{ik}}. \tag{26}$$

Even though all of these coordinates have been used to construct parameterizations, it should be noted that only the mean value coordinates yield uv-coordinates which are provably contained within the convex hull of the neighboring points and thus never lead to self intersections of triangles in the parameter domain. This is due to the fact that Wachspress and discrete harmonic coordinates can assume negative values in certain configurations [29].

Another aspect to consider is choosing shape and distribution of the boundary points. While a convex boundary shape will guarantee the bijectivity of the parameterization, if positive weights are used, it may also introduce significant distortion. The fact that the boundary needs to be fixed in advance is one of the biggest drawbacks of barycentric mapping. A solution to circumvent this restriction is to introduce additional boundary polygons, which create virtual layers of geometry around the actual boundary (e.g., as implemented in [37] and [34]). After the barycentric mapping has been computed the additional boundary polygons are removed, leaving the mesh in a less distorted shape. For distribution of the boundary points along the convex target shape, univariate methods like *chord length* [2] or *centripetal parameterization* [36] can be utilized.

Barycentric methods are generally fast and reliable but the requirement of fixing the boundary severely limits their possibilities. Therefore, the next sections will review methods that establish the boundary during their computation and often lead to more natural solutions with less introduced stretch.

3.3.2 Conformal Mapping

Conformal maps are a family of parameterization methods that target angle preservation. These methods try to construct a mapping such that the anisotropy ellipse (defined in section 3.2) is a *circle* for all surface points. This results in the relation

$$\mathbf{x}_v = \mathbf{n} \times \mathbf{x}_u \tag{27}$$

where $\mathbf{n}$ denotes the normal vector. This means the partial derivatives $\mathbf{x}_u$ and $\mathbf{x}_v$, which are also referred to as the *gradients* of the mapping, are orthogonal and have the same length. Another aspect to note is that the inverse of a conformal parameterization is conformal as well [9]. This is important as we target a mapping that needs to be constructed from the known surface S to the unknown parameter domain Ω, which mathematically is an inverse parameterization.

An intuitive understanding of the notion of conformality can be supported by discretizing the gradient point of view, since conformality can be seen as the relation between the parameterization gradients. For a function that maps the triangulated surface to its parameter domain, a gradient can

simply be described by an orthonormal basis that is constructed per surface patch. This basis (X, Y) of a triangle $[\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k]$ is build with help of the triangle normal, as for example described in [40]:

$$X = \frac{\mathbf{p}_j - \mathbf{p}_i}{||\mathbf{p}_j - \mathbf{p}_i||}, \quad \mathbf{n} = \frac{X \times (\mathbf{p}_k - \mathbf{p}_i)}{||X \times (\mathbf{p}_k - \mathbf{p}_i)||}, \quad Y = \mathbf{n} \times X. \tag{28}$$

The point $\mathbf{p}_i$ can be used as an origin of this system. As shown in [9], the gradients of a parameterization that maps from (X, Y)-space to (u, v)-space are defined by the transformation M_T, which can be written out as

$$M_T = \frac{1}{2A_T} \begin{bmatrix} Y_j - Y_k & Y_k - Y_i & Y_i - Y_j \\ X_k - X_j & X_i - X_k & X_j - X_i \end{bmatrix}, \tag{29}$$

where A_T denotes the area of T. The gradients are then given by

$$\nabla u = M_T \begin{pmatrix} u_i \\ u_j \\ u_k \end{pmatrix}, \quad \nabla v = M_T \begin{pmatrix} v_i \\ v_j \\ v_k \end{pmatrix}. \tag{30}$$

Now, conformality, as characterized by Equation (27), can simply be expressed through a (counterclockwise) rotation of 90 degrees in the (X, Y) plane, or around $\mathbf{n}$:

$$\nabla v = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \nabla u. \tag{31}$$

An approach based on this criteria is the *least squares conformal maps* method (LSCM), which was established by Lévy et al. [41]. The proposed functional simply expresses the conformality condition (31). Since piecewise linear functions usually do not admit a truly conformal parameterization, an energy term corresponding to the "non-conformality" is introduced, which is then minimized. From the established gradient-perspective, this energy functional can be written as

$$E_C = \sum_{T \in \mathcal{T}} A_T || \nabla v - \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \nabla u ||^2. \tag{32}$$

It should be noted that E_C requires the fixing of two (u, v) coordinates for it to be a well defined optimization problem. This is due to its invariance regarding translation and rotation in Ω. A drawback of this method is that these fixed parameter points heavily determine the outcome of the parameterization.

Along analytical conformal methods like LSCM, geometrical methods to approximate conformality have been developed as well. A method dominant in this field is *angle-based flattenig* (ABF), introduced by Sheffer et al. [54] and later optimized several times [56] [68]. ABF and its derivations

minimize an energy term that is defined in angle-space, which means they measure the relative deviation of the unknown parameter angles to the known surface angles. Similar approaches approximating conformal symmetry have been developed under the name *circle packings* [6] and *circle patterns* [7]. Usually, these geometrical methods introduce less stretch into the parameterizations of objects with regions of high Gaussian curvature.

Conformal mapping methods target defining circular anisotropy ellipses, or equivalently the preservation of angles. A problem they do not address is *area preservation*. Enforcing the conformality of a mapping can lead to severe distortion of areas, as the anisotropy ellipse might need to be scaled in order to achieve a circular appearance. The next section will therefore further examine the area-angle relation and introduce methods based on additional analysis of the properties of the anisotropy ellipse.

3.3.3 Distortion Analysis

Differential Geometry methods, as partly reviewed in section 3.2, provide an important baseline of understanding what parameterization methods are able to achieve. It can be shown that all information on the distortion amount of a specific parameterization is contained within the axes of the anisotropy ellipse, or equivalently the singular values σ_1, σ_2 of the Jacobian $\mathbf{J}$. It is therefore logical that a parameterization method, which endeavors to minimize introduced stretch, relies on the optimization of σ_1 and σ_2. Since we are dealing with triangulated surfaces, it is necessary to first discretize the Jacobian $\mathbf{J}$ and the first fundamental form $\mathbf{I} = \mathbf{J}^T\mathbf{J}$, as it can be used to efficiently compute the singular values.

The discrete Jacobian $\mathbf{J}_T$ is defined per linear patch, i.e., triangle T. It can be quickly described using the gradients of the mapping from local triangle coordinates (X, Y) to parameter coordinates (u, v) introduced in section 3.3.2, as it is the matrix that consists of the column vectors of partial derivatives or gradients:

$$\mathbf{J}_T = \begin{bmatrix} \partial u/\partial X & \partial v/\partial X \\ \partial u/\partial Y & \partial v/\partial Y \end{bmatrix} = [\nabla u, \nabla v], \tag{33}$$

where the gradient vectors ∇u and ∇v are defined triangle-wise through M_T (see section 3.3.2). The first fundamental form $\mathbf{I}_T$ can then be deduced per triangle T:

$$\mathbf{I}_T = \mathbf{J}_T^T\mathbf{J}_T = \begin{bmatrix} E & F \\ F & G \end{bmatrix} = \begin{bmatrix} \nabla u^T\nabla u & \nabla u^T\nabla v \\ \nabla u^T\nabla v & \nabla v^T\nabla v \end{bmatrix}. \tag{34}$$

Then the singular values, or equivalently the lengths of the axes of the anisotropy ellipse, can be computed through

$$\sigma_{1,2} = \sqrt{\frac{1}{2}(E+G) \pm \sqrt{4F^2 + (E-G)^2}}. \tag{35}$$

The formulas (33) to (35) are adapted from [9].

The approach from here generally consists of describing a relation between σ_1 and σ_2, which needs to be a minimization problem that can be solved within reasonable time. A good starting point is to remember that the configuration $\sigma_1 = \sigma_2$ describes conformality: the anisotropy ellipse is a circle and the parameterization is angle-preserving. We further know that when $\sigma_1\sigma_2 = 1$ the anisotropy ellipse is area-preserving. Such a parameterization is called *equiareal* and is also desirable in some cases. If we combine these two properties to $\sigma_1 = \sigma_2 = 1$ we would get an *isometric* parameterization, where all lengths are preserved. However, such a mapping can only exist for so called *developable* surfaces, such as cylinders, planes or cones, which was already shown by Gauß in 1827 [21]. Additionally, trying to converge to the isometry-property through minimization results in a highly non-linear problem with many local minima [9]. Therefore, other relations between σ_1 and σ_2 have been examined, some of which will now be quickly reviewed. These *distortion analyzing methods* try to minimize an energy term, which can be expressed through the singular values of the Jacobian, but is sometimes denoted in different forms. To keep these energy terms comparable, in the following, they are all expressed through the singular values, using a notation adapted from [29]:

- **Dirichlet Energy.** Also described as *harmonic mapping*, the Dirichlet energy was one of the first parameterization methods developed in context of computer graphics [47]:

$$E_D(\sigma_1, \sigma_2) = \frac{1}{2}(\sigma_1^2 + \sigma_2^2). \tag{36}$$

  A disadvantage of this energy is that the obvious minimum can be achieved through $\sigma_1 = \sigma_2 = 0$, which results in a parameterization that maps all surface patches to a single point. Therefore, harmonic maps require a fixed boundary. Then the minimization of the energy term (36) is equivalent to computing a barycentric mapping with barycentric coordinates as in Equation (24), which are therefore labeled *discrete harmonic coordinates*.

- **Conformal Energy.** A conformal mapping, as discussed in the previous section, can also be derived from minimizing the conformal energy

$$E_C(\sigma_1, \sigma_2) = \frac{1}{2}(\sigma_1 - \sigma_2)^2. \tag{37}$$

  This approach has been discovered by Desbrun et al. [45] and Lévy et al. [41]. As already described in section 3.3.2, when applying this method, two uv-coordinates need to be fixed in advance. This may result in undesirable outcomes, as the choice of which coordinates to fix can have a considerable impact on the parameterization.

- **MIPS Energy.** The first method that could fully create a natural boundary was introduced by Hormann and Greiner [28] as the so called *most isometric parameterization of surfaces* (MIPS). It is based on the ratio of the two singular values:

$$E_M(\sigma_1, \sigma_2) = \frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1} = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2}. \tag{38}$$

  The minimum is achieved when $\sigma_1 = \sigma_2$, which means MIPS is also a conformal mapping. Since conformal maps tend to favor angle versus area preservation, an adaptation of the MIPS energy was proposed by Degener et al. [15], where a second term is introduced that measures the area deviation:

$$E_\theta(\sigma_1, \sigma_2) = E_M(\sigma_1, \sigma_2) \cdot \left(\sigma_1 \sigma_2 + \frac{1}{\sigma_1 \sigma_2}\right)^\theta, \tag{39}$$

  where $\theta > 0$ is a user-specified value that defines the impact of area distortion on the energy. This energy is minimal for locally isometric mappings, however both E_M and E_θ result in non-linear problems that can be difficult to solve or approximate. In the approach of [15] an iterative method is used that moves the uv-coordinates each iteration to a more desirable position.

Distortion analysis provides a valuable baseline for comparing different parameterization methods. However, it should be kept in mind that in addition to distortion, factors depending on the context of use should be considered when selecting a method. For example, some applications work well with fixed-boundary methods, which are usually simpler to implement and can be solved fast with state of the art linear solvers. Furthermore, most appliances require a parameterization to be bijective, which means triangle flips must not occur (local bijectivity) and/or the boundary may not self-intersect (global bijectivity). Not all parameterization methods can guarantee these properties.

4 Related Work

This chapter will focus on related work in general, including applications of parameterization in medical domains (section 4.1) with which the mitral valve models used in this work were created.

A great part of medical imaging research is dedicated to developing methods for manually or automatically extracting triangulated meshes from medical images or scans. Source image modalities include, but are not limited to, computer tomography (CT), magnetic resonance imaging (MRI) and 3D ultrasound (US). The process of creating a 3D mesh out of such a data-set is referred to as *surface segmentation*. Recent developments in the domain of heart valve segmentation are reviewed in this chapter in section 4.2. This work strongly relies on these approaches, as it processes patient-specific triangulated mitral valve models, which have already been extracted from US data.

The goal of this work is to transform the triangulated 3D mitral valves to a 2D-view through parameterization, in order to gain a perspective that allows inspecting the whole valve at once. This method has been used for other medical visualization purposes, including brain and colon visualization. Some of the methods that were proposed in these domains will now be reviewed.

4.1 Surface Parameterization in Context of Medical Visualization

Computing parameterizations of medical segmentations can provide several benefits:

- A visualization of a planar parameter domain provides a view on the whole surface geometry at once. This can be useful if certain features of the object are occluded or difficult to assess in a 3D representation. Such a view can also prove valuable when comparing points in time or surfaces from different scans (e.g., pre- and post-surgery). This can be facilitated by applying color-maps to the surface, which correspond to certain features of the object.

- Parameterizations can also be understood as coordinate systems intrinsic to a surface. If these surface coordinates are applied in a way that they correspond with important anatomical landmarks, they can be used for systematic localization, even across data-sets from different patients or of distinct modalities.

- *Image registration*, the process of aligning two separate data-sets of the same real world object, is a common problem in medical image analysis. A parameterization can be used to support this operation by transferring a 3D problem into the easier to manipulate 2D domain.

A popular field of parameterizations in medical context is brain surface analysis. Generally, the cortical surface (usually extracted from MRI data) is mapped to a sphere, as brain surfaces are topological spheres. The process can be thought of as an inflation of the brain, making the whole surface visually accessible [18]. Similar approaches have been used to generate a brain surface atlas [58], or analogously a surface based coordinate system for localization purposes and intersubject matching [12]. Wang et al. [63] introduced a "curvilinear" coordinate system (instead of systems based on a sphere or plane) that can be utilized to partition the brain surface into patches and map these conformally to parallelograms. Zhu et al. [70] proposed a structured parameterization that can be used for statistical shape analysis. They observed that solely preserving angles or areas does not really guarantee good anatomical correspondence on parameterized medical objects like a brain surface. Therefore they use shape-based landmarks which are always mapped to the same uv-coordinates. Other vertices are then mapped bijectively to the spherical parameter domain based on a heat conduction model. This leads to a correspondence of specific anatomical landmarks across different parameterized data-sets.

The realization that only relying on minimizing stretch in a parameterization might not lead to the desired results (specifically in medical visualization) is an important point to stress. The idea of fixing certain landmarks in the parameter domain to achieve better correspondence and comparability should be kept in mind, as it will be an important concept used later on in this work.

Another broadly explored domain of medical surface parameterization is colon analysis. The rather obtrusive practice of *colonoscopy*, the endoscopic examination of the large bowel, is a crucial tool in the detection and prevention of colon cancer. Nowadays, a *virtual colonoscopy* can provide comparable or complementary diagnostic abilities. For this, a model of the colon is reconstructed from CT or MRI scans, which can then be visually explored by a physician. Navigating this 3D model can be cumbersome, which quickly lead to the development of approaches where the colon model is flattened in the plane, providing a single 2D view of the whole colon. This can be imagined like a virtual dissection, where the tube-like colon structure is cut on one side and then laid out flat. Such methods can assist with polyp detection [27] [3] [24]. Polyps are small growths that can appear in the colon and can turn malignant. Through parameterization the 3D polyp detection problem is transformed to a 2D pattern recognition problem. The parameter domain can also be used as a map when navigating the 3D object. Zeng et al. [69] used a similar method to help with the registration of supine (facing up) and prone (facing down) colon CT scans. Thus the 3D registration problem becomes a 2D feature alignment problem. A recent proposal by Nadeem et al. [46] further expands these approaches through eigenfunction analysis. This allows refined global and

local registration and an automatic way to find a cut alongside the colon model.

Apart from brain surface and colon visualization, parameterizations have been used in other medical domains as well. Vera et al. [61] proposed a way to apply a volumetric coordinate system to liver models, which incorporates the interior of the organ. The approach is landmark based and thus allows integration of data across image modalities and patients. They use a harmonic mapping to compute the parameterization around the fixed landmarks. Marino et al. [44] [43] investigated the visualization of tubular structures, namely blood vessels and air ways. They applied 2D mappings to these objects with the intention to preserve the overall geometric structure. These flat representations can, for instance, provide a guidance map during endoluminal navigation.

Last but not least, surfaces generated in a more abstract scope can also benefit from visualizations of certain parameter domains. Krone et al. [35] recently showed that the visualization of molecules can be enhanced by applying a parameterization. Here, the same concepts as above apply as well: In order to achieve a simple view of biochemical properties or geometrical features of complex molecules, a mapping to a spherical or planar domain can be constructed that allows easier evaluation. Therefore the analysis and comparison of different data-sets or points in time is simplified.

4.2 Heart Valve Segmentation and Simulation

Patient-specific 3D visualizations of heart valves can provide means to characterize their pathophysiology and assist in preoperative planning. Disorders, as discussed in section 2.2, can be better detected and strategies can be reflected or even simulated. Several approaches to derive a triangulated surface geometry from medical scans of heart valves have been proposed. These include mechanisms based on cardiac MRI [30] [13] and cardiac CT [23] [25]. These methods allow, for instance, approximation of myocardium material properties and simulation of forces generated along fiber orientation. They can help to determine the strain and stress distribution in the ventricles and simulate blood flow. For example, Hammer et al. [25] proposed a mass-spring model of the mitral valve to simulate valve closure. Their goal is to provide a system in which potential mitral valve repair strategies can be virtually tested and compared. A drawback of methods like this is that the modeling and simulation of ventricular mechanics requires a huge part of the segmentation to be conducted manually, as various material types and locations need to be differentiated.

Another approach to achieve a high resolution heart valve segmentation is using 3D US techniques [1] [10] [51] [17]. Also, combinations of several source modalities have been proposed [31]. Heart valve assessments through US are usually based on *3D transesophageal echocardiography*

scans (TEE), a procedure where a 3D US probe is inserted into the esophagus. This allows a positioning of the probe very close to the back of the heart and a view unobstructed by the rib cage. See Figure 8 as an example. A common challenge to increase the clinical viability of TEE based heart valve segmentation is achieving a high degree of (fast) automation while keeping the generated models close to what a precise but slow manual segmentation is able to obtain. Furthermore, when building a geometry from a TEE heart valve scan, it is almost obvious to suggest the additional construction of a simulation, since TEE scans provide real time feedback of the valve's movements and shape. Burlina et al. [10] proposed the computation of a closed mitral valve from an open configuration, as the closed state is more difficult to assess. Aggarwal et al. [1] introduced a simulated model based on splines. The method showed by Engelhardt et al. [17] can capture different states of the valve during a whole cardiac cycle and differentiate between the annulus, posterior and anterior leaflet.

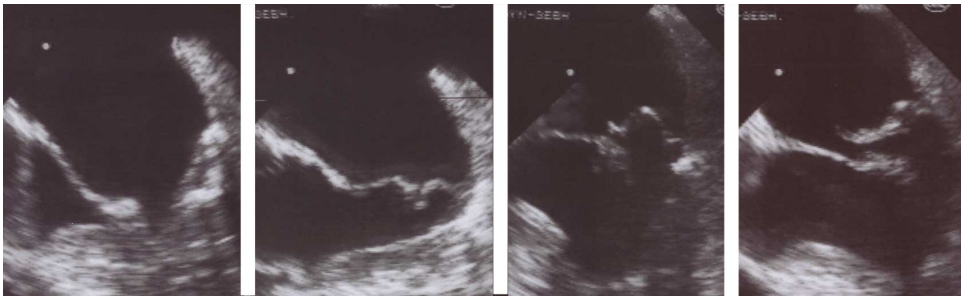

Figure 8: TEE images of a prolapsed mitral valve at different time steps. Image from [64].

4.3 Mitral Valve Models Used in This Work

The work at hand is based on already segmented mitral valve models generated by the German Cancer Research Center[1] (DKFZ) in collaboration with the Heidelberg University Hospital. They are obtained using the open-source Medical Image Interaction Toolkit[2] (MITK). The MITK is developed at the DKFZ by the Department of Medical and Biological Informatics. It acts as an interface between the Insight Segmentation and Registration Toolkit[3] (ITK) and the Visualization Toolkit[4] (VTK). A possible future goal could be the integration of the here developed software prototype into the MITK, which is the reason why an important implementation requirement is compliance with the VTK data formats.

[1] https://www.dkfz.de/
[2] http://mitk.org/wiki/MITK
[3] https://itk.org/
[4] https://www.vtk.org/

On the technical side, the segmentation of the mitral valve models used in this work relies on the algorithm introduced in [17], which in turn builds on an approach by Schneider et al. [51] to detect and automatically segment the mitral valve annulus. For the annulus computation only a central valve point needs to be defined. The segmentation of the leaflets is independent from the state of the mitral valve and results in the creation of two 3D models, one for each of the leaflets. The process can be fully automatic but also leaves open the option of manual assistance or correction. A detailed description of the acquisition pipeline can be found in [39]. The approach is 3D TEE based and an original intend was to utilize it in an automatic estimation of the coaption zone, which is the area where the two leaflets touch during systole. The mean errors of the valve geometry were found to be 1.16 ± 0.38mm for the anterior leaflet and 1.24 ± 0.37mm for the posterior leaflet (for reference, the typical mitral valve has a diameter between 27 and 35mm). As models segmented with this method are used in the work at hand, these metrics apply here as well.

5 Concept

At this point, basic knowledge of the characteristics of parameterizations and their applications in mesh processing and visualization should have been attained. Along with the medical background, this concludes the premise upon which the approach developed and implemented in this work is build on. To establish a 2D-view of the mitral valve, a parameterization method is used, which will be illustrated in this chapter. The method is proposed in accordance with the mentioned 4D data sets (3D models corresponding to different time steps) from the DKFZ, which will be described in detail in section 5.1. Furthermore, the developed parameterization method targets to fulfill certain criteria, which will be discussed in section 5.2.

The 2D-view can only reach its full potential when information on a variety of factors (like pathological indicators or anatomy) is visualized on the surface. This is achieved through color-mapping of computed scalar fields and will be examined in section 5.4. Unlike a 3D model, the 2D-view can give an instantaneous overview of the whole valve surface, allowing such mappings to be intuitively compared with one another.

5.1 Initial Data-Sets

The mitral valve models used in this work are 4D segmentations from TEE scans. Each data-set consists of around ten 3D models acquired at different time-steps of the cardiac cycle. This means that the examined valves can be viewed in their opened and closed conditions, as well as states in-between. During this cycle, features of the valve, like shape and area, are constantly changing, requiring a parameterization algorithm that can adapt to these changes while displaying comparable 2D views. Figure 9 depicts a closed and opened configuration of one of the initial valve data-sets.

A benefit that comes with the algorithm used to segment the TEE data-sets, is the automatic differentiation between the anterior and posterior leaflet (and also the annulus). Each 3D model is split into three meshes corresponding to the surface of the two leaflets and the annulus. Combined, the two leaflet meshes of a single time-step consist of around 4.700 triangles and 2.500 vertices. Due to the capturing technique and movements of the patient's heart, location and orientation of the generated models varies slightly over time and across data-sets. This means a local coordinate system of the valve in terms of specific landmarks, does not exist yet. However, some landmarks are tagged within the meshes' data. They are specified by a per-vertex ID-list, which includes markers for annulus points and leaflet tips.

From a topological perspective, the leaflet meshes are genus-zero open boundary objects. This means a global parameterization in the planar domain is possible and the objects do not need to be cut. It should be kept

in mind though, that in some cases introduced cuts can greatly improve the stability and distortion amount of a parameterization. Another aspect to point out is the topological structure of the meshes. The segmentation results in almost unidirectional lines going from the annulus to the coaptation zone (the lower part of the leaflets), as can be observed in Figure 9. These lines bear resemblance to the longitudes of a globe. This geometrical feature will later be utilized to build a fast initial 2D configuration.

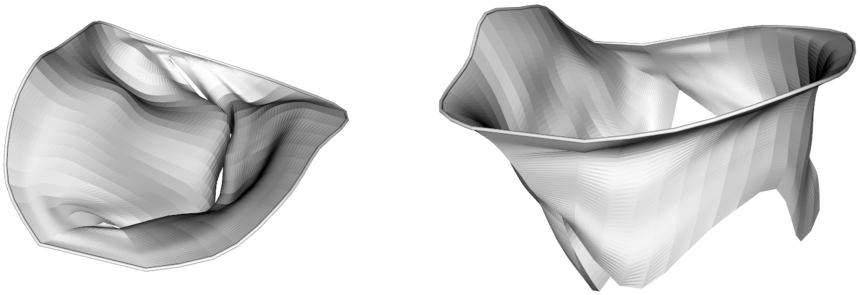

Figure 9: 3D mitral valve models in a closed (left) and opened (right) state.

5.2 Objectives

At the core of this work lie two fundamental questions: Which benefits can a 2D-view of the mitral valve comprise and how can a parameterization be constructed, such that these benefits can be provided to an optimal extend? In order to delve into these issues, the objectives of a possible 2D-view need to be narrowed down. The desired utility of such a view is the facilitation of pathophysiological analysis of the mitral valve. This includes preoperative planning and postoperative comparison, i.e., assisting in the process of evaluating the success of a medical intervention. A 2D-view can aid or simplify these processes by providing a perspective showing all of the valve's surface at once. Features of the valve can be visualized through color-mapped scalar fields. This leads to a formulation of the goals the parameterization algorithm should target:

- Color-maps of valve features should be simple to deploy. They should be applicable to both, a 3D- and 2D-view of the surface. Each color-map should have direct correspondence between the two views. This can be achieved through an internal representation of 3D and 2D geometry, which highlights the fact that in this scenario a parameterization is the favorable approach for constructing a 2D-view. The generation of uv-coordinates will provide a suitable frame for color-maps, since each 2D point can be mapped onto the 3D surface and back.

- The bijective mapping between the 3D and 2D representation should be intuitive to comprehend. The algorithm is targeted at physicians, not mesh processing experts, which means there needs to be an additional emphasis on the correspondence of points in 3D and 2D.

- As the application aims to provide a visceral framework for the comparison of different valves or time-steps, it is highly necessary to secure the similarity of generated mappings. Orientation, shape and size of the 2D-views should be akin to one another. It is desirable that certain landmarks of the valve always correspond to coinciding coordinate ranges in the 2D-view.

- Last but not least, the algorithm should be able to parameterize the geometry of all time-steps. Regardless of the state of the valve (open, closed or in-between), all of the points above should still apply.

Building on these objectives, some technical targets for the application prototype can already be determined. Of course, these focus on the important aspects first, in order to stay within the scope of this bachelor's thesis:

- The prototype should provide a view on the 3D as well as the 2D geometry of a single valve. Additionally, the user should be able to open a second valve model and its parameterization for direct comparison.

- It should be possible to switch between different color-maps and time-steps of the valve.

- Since uv-coordinates will be constructed, it is only sensible to provide an option to save these coordinates or "bake" them into the model. This will allow using the parameterization outside of the application.

- To facilitate utilization in a real-world scenario, calculations should be quick enough that interactivity is possible. This is only a secondary objective, as run-time optimization might prove excessively time-consuming and a proof of concept should be prioritized and evaluated first.

- It is feasible that the developed prototype will at some point be included in an MITK plugin. Therefore, heavy dependency on external libraries is discouraged, with exception of frameworks on which the MITK is already based on (like the ITK or VTK). This also means that the implementation of a user interface will not be a focus of this work and the presented classes will be designed in a backend-oriented approach.

5.3 Parameterization Method

The final goal of the parameterization method will be the accomplishment of the described objectives, through the construction of a mapping algorithm designed to work with the data sets described in section 5.1. The current chapter will examine the parameterization method in general and mathematical terms, keeping separate from a specific implementation. Chapter 6 will further inquire the particulars of the developed software prototype.

From a technical perspective, the quality of a parameterization is most often measured by metrics describing the distortion amount introduced by the mapping. As demonstrated in chapter 3, this property can be divided into area and angle distortion. If a mapping can uphold both these characteristics, it is length preserving and called isometric. While total isometry would indicate a "perfect" mapping, for most surfaces a bijective isometric mapping between 2D and 3D does not exist. When mapping the valve, distortion is necessarily introduced and the parameterization algorithm needs to be constructed in a way that the most important features are prioritized during distortion minimization.

The second aspect to consider, when constructing the mapping algorithm, is comparability and readability of the 2D-view. While a low distortion amount can increase the similarity of features when comparing a 2D and 3D representation, it does not guarantee a consistent depiction of the 2D-view. This is especially due to the changing shape of the mitral valve during the cardiac cycle, which leads to differences in the 3D models that could manifest in varying shapes of the parameter domain. To keep a high comparability, specific parts of the 2D-view should always correspond with the same valve landmarks. These include the saddle horns and commissure lines, marked in Figure 10. Location of broader structures, like the two leaflets and the annulus, should also be analogous across different 2D-views.

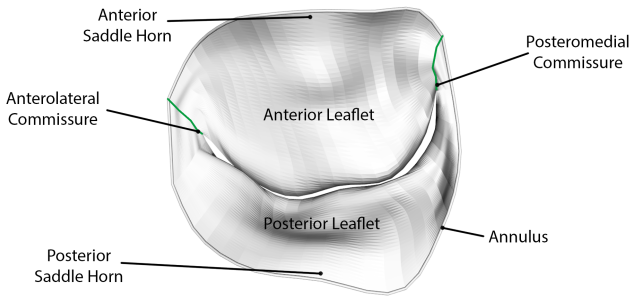

Figure 10: Landmarks of the mitral valve. The 3D model is viewed from the top, where the left atrium would be.

In the following sections the parameterization approach developed in this work will be explored in detail. The leading idea is to virtually cut the valve along the anterolateral commissure line, then "fold" it open and flatten the leaflets in a way that they lie side by side. This is somewhat equivalent to the typical parameterization of a cylinder and based on the conclusion that the mitral valve exhibits a cylindrical shape, especially in its open configuration (see right side of Figure 9). The 2D-view will then show the anterior leaflet next to the posterior leaflet with the annulus on top and the coaptation zone at the bottom. A view like this can be found in heart valve literature such as [11]. Figure 11 shows the desired layout. It is however not possible to include features like the papillary muscles and cordae tendineae into the computed 2D-view, as they are too small to be detected by a TEE-scan and could only be approximated. The presented approach will rely on patient-specific measurements only, which are discretized in the 4D data sets.

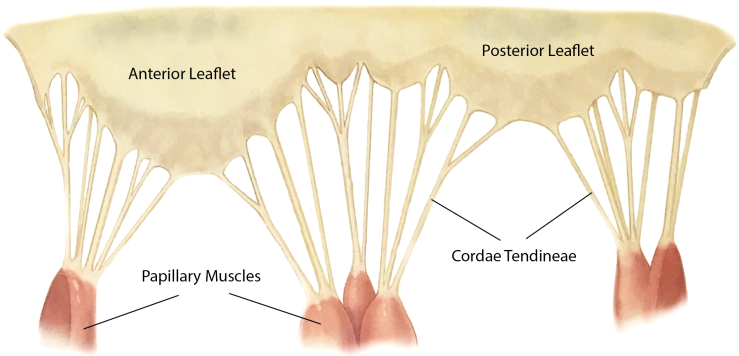

Figure 11: Depiction of the mitral valve cut along the anterolateral commissure and laid out flat. Image taken from [11].

The algorithm to compute such a layout is divided into several steps. Their order is determined by the importance of features during clinical analysis. First, the annulus is parameterized (section 5.3.1), keeping its length and relative shape. Then the valve surface is "unrolled", starting from the annulus and going to the coaptation zone, while maintaining the intrinsic distance in this direction (section 5.3.2). Lastly, the uv-coordinate positions are relaxed, using an iterative approach, minimizing introduced distortions measured by an energy term (section 5.3.3).

Before these phases are demonstrated in detail, the mathematical notation introduced earlier will be extended, to serve as an explanatory basis. Each 3D valve mesh consists of vertices $\mathcal{V} \subset \mathbb{R}^3$. A 3D vertex is always denoted as $\mathbf{p}_i = (x_i, y_i, z_i)$ and its counterpart in 2D as $\mathbf{q}_i = (u_i, v_i)$. The

points $\mathbf{q}_i$ make up the parameter domain $\Omega \subset \mathbb{R}^2$. The algorithm will address distinct subsets of vertices $\mathcal{V}_l = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \subset \mathcal{V}$, each of which contains points on a single line (see the red points in Figure 12). If the mesh is thought to be cut from a sphere, these lines can be imagined like topological longitudes. The vertices in $\mathcal{V}_l$ are ordered from top to bottom, i.e., from the annulus to the coaptation zone. In the processed data sets all points are located on exactly one of these lines. This means the union of all sets $\mathcal{V}_l$ is equal to $\mathcal{V}$ and the intersection of two longitude sets is always empty.

Another important subset will consist of vertices lying on the annulus $\mathcal{V}_a = \{\mathbf{p}_1, \ldots, \mathbf{p}_m\} \subset \mathcal{V}$ (see the green points in Figure 12). They are ordered clockwise, starting from the anterolateral commissure. All annulus-points are the first vertex in one of the longitude sets $\mathcal{V}_l$.

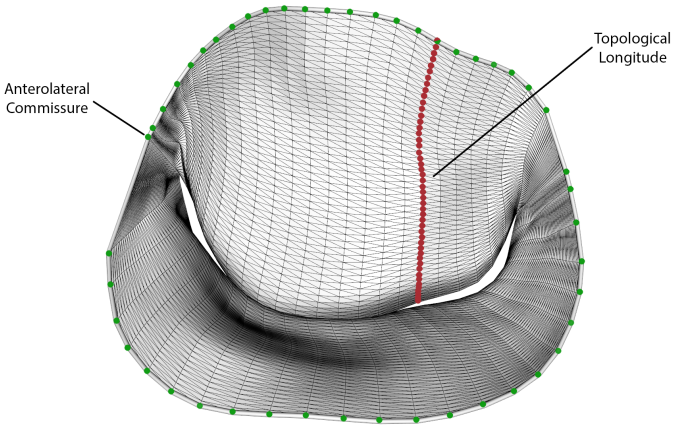

Figure 12: Visualization of the sets of vertices used. The annulus points $\mathcal{V}_a$ are marked in green, the first one beeing the point on the anterolateral commissure. The vertices of one topological longitude $\mathcal{V}_l$ are marked in red.

5.3.1 Annulus Parameterization

The annulus is independently parameterized first. Only after that will the rest of the surface be mapped to a 2D domain, while the annulus remains fixed to its initial configuration. This allows a high amount of control over the general shape of the 2D view, which is determined by the annulus layout, resulting in intuitively comparable outcomes across time steps and different data sets.

As the annulus is a topological ring it can be bijectively mapped to a 2D curve through the use of polar coordinates, similar to how a circle around the globe would appear on a world map. First, a point needs to be chosen at which the annulus is cut. As the leaflets are supposed to be separated

at the anterolateral commissure, the annulus point on this commissure will be used. This landmark based approach is uniformly employed to increase similarity and comparability of the 2D view. The anterolateral commissure point would then correspond to the polar coordinates of $\phi = 0$ and $\theta = 0$. Since we are dealing with discrete points in $\mathbb{R}^3$ that require a counterpart in $\mathbb{R}^2$, this is executed by assigning a uv-coordinate. If the set of vertices $\mathcal{V}_a = \{\mathbf{p}_1, \ldots, \mathbf{p}_m\}$ contains all annulus-vertices in a clockwise order, starting from the anterolateral commissure point $\mathbf{p}_1$, then its uv-coordinate is described with $\mathbf{q}_1 = (0, 0)$. Now the rest of the points in $\mathcal{V}_a$ needs to be assigned uv-coordinates. The simplest approach would be to understand the annulus as a circle or ellipse approximation. It could be imagined like a topological latitude or an equator of the model. This would map its points to a straight line in the parameter space, what can be expressed by

$$\begin{aligned} v_i &= 0, \\ u_i &= \begin{cases} 0 & \text{if } i = 1, \\ u_{i-1} + ||\mathbf{p}_i - \mathbf{p}_{i-1}|| & \text{otherwise.} \end{cases} \end{aligned} \tag{40}$$

Such a mapping makes the annulus an iso-v curve, as all coordinates share the same v-value. Since the distances between points are equal in 2D and 3D, a mapping like this could be used to visualize annulus length. However, it does not contain any other clues and might introduce unnecessary distortions, as the form of the annulus is usually described as a "saddle-shape", rather than ellipse- or circle-like. As can be seen in Figure 10, certain points of the annulus are located higher than others in healthy mitral valves. This especially holds true for the anterior saddle horn. Only a non-physiological dilation of the valve can move it to a position where the mitral valve annulus appears more like an actual ellipse. Height-deviations in the annulus shape can be pathological indicators, which means it could prove advantageous to visualize them in a 2D view.

Differences in height can be expressed through v_i, which makes them relative to the u-axis in the parameter space. A possible method to assess these values in 3D, is to construct a plane $\mathbf{P}_a$, which corresponds to the u-axis in 2D. The relative heights of annulus points can be measured by computing distances to $\mathbf{P}_a$. Such a plane needs to be approximated per valve model, a process that is somewhat equivalent to creating a local coordinate system. The first approach that comes to mind to compute $\mathbf{P}_a$ is to use a least-squares method, where a plane is constructed in a way that it has minimal distance to all annulus points. This is a standard procedure, that can for example be reviewed in [53]. However, trying this method showed several problems:

- If only a small subset of points change their location (e.g., over time), the parameterization will display a movement in all points, because the reference plane will rotate slightly.

- If the annulus model is not sampled with equal distances between all points, $\mathbf{P}_a$ will lean towards regions of higher sampling rate. This will offset the expected orientation.

- The resulting distances do not correlate well with the height differences between landmarks of the valve. For example, the saddle horn, which is usually obviously higher than the rest of the valve, does not show this height difference in the 2D view, because the reference plane is rotated towards it (see also Figure 13).

Because of these results, a second, more landmark-based approach was implemented. The annulus-plane $\mathbf{P}_a$ is defined by three points: the two annulus commissure points and the barycenter of all annulus points, which belong to the posterior leaflet. The first two form a natural axis through the valve and will automatically be mapped to $v = 0$, which means the height of all other points is visualized in relation to these landmarks. As this will be consistent across all models, comparability will be increased. There exists no similarly natural way to place the third point. The barycenter of the posterior annulus is chosen, because this part of the annulus already approximates a planar distribution of points in most valve models. Using the barycenter, instead of a specific point, will counter the impact of small height deviations. The resulting 2D view puts emphasis on the location of the anterior saddle horn and allows visceral evaluation of its height compared to the rest of the valve.

Regardless of how the annulus-plane $\mathbf{P}_a$ is approximated, the mapping is build the following way: The normal $\mathbf{n}_a$ of $\mathbf{P}_a$ is build to point upwards, towards where the left atrium would be. The direction of $\mathbf{n}_a$ needs to be equal in all cases, because it will be implicitly used to determine whether annulus points are above or below $\mathbf{P}_a$. Let's assume the function $proj(\mathbf{x}, \mathbf{P})$ returns the projection of a vector $\mathbf{x}$ onto the plane $\mathbf{P}$. The uv-coordinates can then be constructed in a way that v_i is equal to the signed distance of $\mathbf{p}_i$ to $\mathbf{P}_a$. Then u_i is derived from the distance to the previous annulus point in 3D, such that it stays the same in 2D:

$$
\begin{aligned}
v_i &= \mathbf{n}_a \cdot (\mathbf{p}_i - proj(\mathbf{p}_i, \mathbf{P}_a)), \\
u_i &= \begin{cases} 0 & \text{if } i = 1, \\ u_{i-1} + \sqrt{||\mathbf{p}_i - \mathbf{p}_{i-1}||^2 - (v_i - v_{i-1})^2} & \text{otherwise.} \end{cases}
\end{aligned} \tag{41}
$$

In the developed software prototype the user can choose the annulus parameterization method. Figure 13 displays a comparison of the approaches.

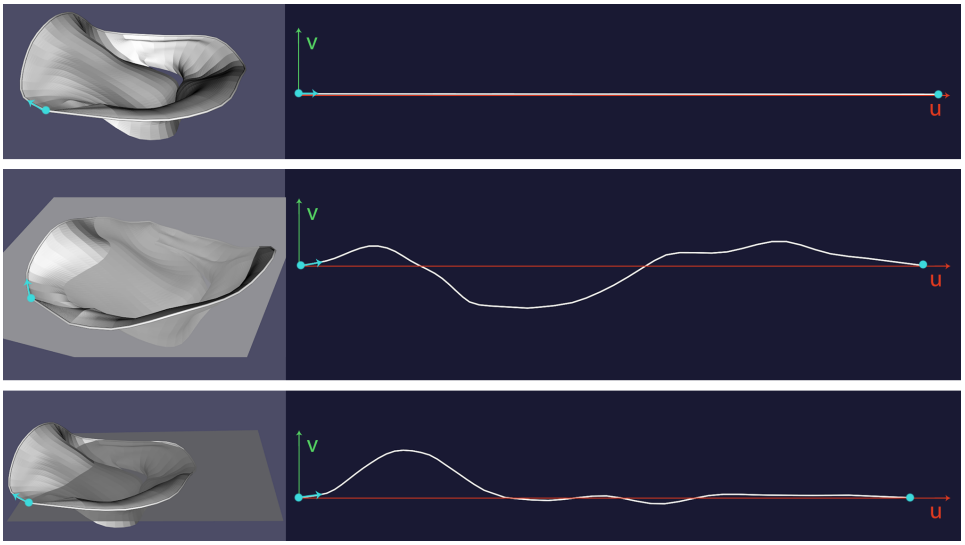

Figure 13: Different methods of annulus parameterization. The cutting point mapped to $q_1 = (0, 0)$ is marked. Top: The annulus is defined as an iso-v curve. Middle: The plane in which the annulus lies is approximated with the least-squares method. Then the relative distances of the annulus points to this plane are parameterized by v in 2D. Bottom: The plane is constructed through a landmark based approach.

5.3.2 Initial Leaflet Layout

At this point, the annulus is already parameterized. All its vertices have corresponding uv-coordinates in the parameter domain. Additionally, its arc length is preserved. Now, the rest of the leaflet's geometry needs to be parameterized. As mentioned above, to keep up coherence across models, the uv-coordinates are always laid out below the annulus. This step requires a configuration of uv-coordinates, such that the mesh does not self intersect and no triangle-flips occur. Afterwards, iterative methods can be applied to minimize stretch by moving the coordinates to better positions.

Since the initial uv-setup is relatively unconstrained, there are many possible ways to construct it. The approach used in this work utilizes a topological feature the valve models presented in Section 5.1 exhibit: the valve discretization results in a triangulation with lines of vertices going from the annulus to the lower verge of the leaflets. In a parameterization context, these can be compared to the longitudes of a globe (imagine the valve model to be cut from a deformed sphere). In the world-map metaphor, the longitudes are iso-θ-curves. In the case of uv-coordinates they can be thought of as iso-u curve approximations. This means all points corresponding to one longitude are mapped to a shared u-coordinate and distances between these points are preserved in v-direction. If the set of vertices on a shared longitude is described with $\mathcal{V}_l = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$, where

$\mathbf{q}_1 = (u_1, v_1)$ is the already assigned uv-coordinate of the annulus, then each uv-coordinate $\mathbf{q}_i = (u_i, v_i)$ corresponding to the vertex $\mathbf{p}_i \in \mathcal{V}_l$ can be set through

$$\begin{aligned} u_i &= u_1 \qquad \forall i \in [2, n] \\ v_i &= v_{i-1} - ||\mathbf{p}_i - \mathbf{p}_{i-1}|| \qquad \forall i \in [2, n]. \end{aligned} \tag{42}$$

For this to work the vertices in $\mathcal{V}_l$ need to be ordered from the upper to the lower edge of the model and the longitudes may not intersect each other. Also, all points need to be located on exactly one longitude. While this holds true for the models used in this work, it can be a potential drawback if the method is at some point extended to other valve data sets. However, this step of the pipeline can be replaced by another algorithm, keeping in mind that only an initial valid uv-configuration needs to be found.

This approach was chosen, because it is a fairly fast method that naturally develops a free boundary. The uv-configuration is build in real-time without a noticeable delay and can already provide expressive information, as the arc length of the longitudes is visualized in the 2D-view. These lengths are good approximations of the intrinsic "vertical" distances one could measure on the valve's surface, i.e., when walking straight paths on the mesh from the annulus to the lower rim. Figure 14 shows an example output of this method with a marked topological longitude.

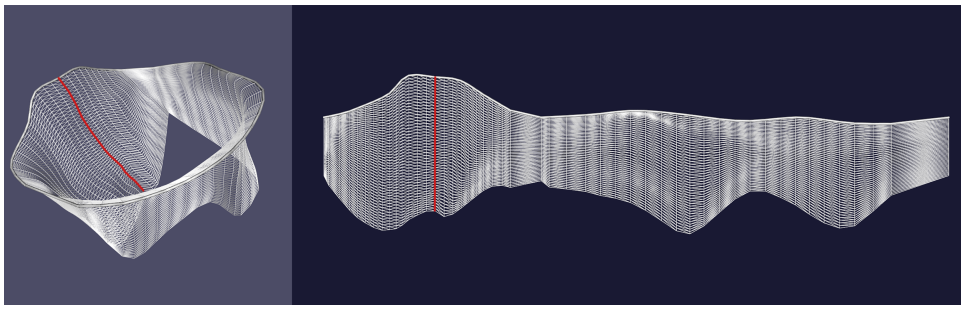

Figure 14: Initial layout of the mapping from 3D (left) to 2D (right) below the already parameterized annulus. Following the world-map metaphor, iso-u curves are approximated using the model's topology. An example iso-u curve is shown in red. For better visualization only the wireframe of the mesh is shown.

5.3.3 Relaxation

The final step of the parameterization algorithm will be the application of an iterative relaxation method. The current 2D configuration preserves consistency of anatomical landmark positions across data sets. Additionally, feature lengths of the annulus and the approximated longitudes are maintained. However, certain aspects of this layout introduce an unnatural stretch into the parameterization, mainly due to two aspects: the vertex-

curves, which are interpreted as longitudes, deviate slightly from straight lines on the surface. Also, the valve diameter gets smaller when approaching the coaptation zone, especially in closed states. As the u-length of the parameter domain has only been determined by the annulus, the lower parts of the leaflets are unnecessarily widened. The goal will now be to minimize these distortions, while keeping the anatomical correspondence across time steps and data sets high.

In general, to find a configuration of uv-coordinates with minimal distortion, an energy term is introduced, which is dependent on some relation of the Jacobian's singular values (see section 3.3.3). Then, this energy is minimized, either with the use of direct solvers or by iterative refinement of vertex positions. The latter approach is used here, because we already have a valid configuration of uv-coordinates and only desire optimization. As the distortion is mainly concentrated around the lower parts and the commissure lines of the leaflets, the annulus will remain fixed. This will ensure that the annulus' diameter and height differences remain visually expressed factors of the 2D-view.

As mesh parameterization is a broadly examined subject, literature provides an abundance of different distortion energy terms. There are two primary focuses: angle preservation or conformality and area preservation. Both of them can be seen as important for the desired 2D-view of the mitral valve. As shown in chapter 3, they can be combined to target *isometry*, i.e., length preservation. Based on the assumption that optimizing edge length stretch of the parameter domain will help converge to an isometric 2D representation, an energy depending on these distortions will now be introduced. The 3D mesh consists of vertices $\mathbf{p}_i$, corresponding to uv-coordinates $\mathbf{q}_i$. If the set N_i contains the indices of all neighbors of $\mathbf{p}_i$, a per-vertex length energy can be described as

$$E_l = \frac{1}{|N_i|} \sum_{j \in N_i} \frac{||\mathbf{q}_i - \mathbf{q}_j||}{||\mathbf{p}_i - \mathbf{p}_j||} + \frac{||\mathbf{p}_i - \mathbf{p}_j||}{||\mathbf{q}_i - \mathbf{q}_j||} \tag{43}$$

Note that this energy reaches its minimum $E_l = 2$, if and only if all observed edge lenghts in the 3D mesh are equal to their counterpart in the parameter domain and therefore both fractions evaluate to 1. The factor $1/|N_i|$ accounts for different numbers of neighbors, because this energy is evaluated per vertex.

Global minimization of E_l can be achieved by applying a local iterative spring-relaxation method. Similar to how barycentric mapping is explained in section 3.3.1, the edges between vertices can be interpreted as physical springs. However, a barycentric mapping cannot be applied directly, because the lower mesh boundary is supposed to develop freely. Therefore, an Euler method will be used, where the vertices are displaced each iteration in the direction of a summed spring force calculated based

on the model's edge lengths. This approach is equivalent to mass-spring systems commonly used to model the behavior of cloth or fabric. A similar method has been proposed to simulate mitral valve closure [25], as its tensile tissue behaves similar to a network of springs.

A linear spring exerts a force equal to $F = -kx$, where k is the spring constant and x is the distance the spring is stretched or compressed in relation to its rest length. The minus sign shows that the force pushes into the opposite direction of the stretch or compression. For each uv-coordinate $\mathbf{q}_i$ the force F_i is evaluated over its one-ring, by assigning each edge (i.e., spring) a rest-length equal to its actual length in the 3D model. The summed force is applied through:

$$F_i = \sum_{j \in N_i} k\Big(||\mathbf{p}_j - \mathbf{p}_i|| - ||\mathbf{q}_j - \mathbf{q}_i||\Big) \frac{\mathbf{q}_j - \mathbf{q}_i}{||\mathbf{q}_j - \mathbf{q}_i||}, \tag{44}$$

where k is the constant stiffness parameter and will be set to $k = 1$ for our purposes. The minus sign is omitted, because the direction the force is applied to can just as well be inverted.

A drawback of the iterative Euler method is its error when using large time intervals. For the application at hand this means the calculated forces cannot be applied fully, but need to be scaled down. Otherwise, vertices on the border will be pulled towards the inside and can cross mesh edges and thus lead to self-intersections. This can be prohibited by restraining points to stay within their neighbor-kernel, but this method requires extensive additional computation. A simpler way of minimizing self-intersections is to use small time intervals (scaling the forces down) and increase the number of total iterations. It should be noted though, that a high number of iterations will be time consuming. In a future adaptation this process could be facilitated through parallelization. In the current approach the user can define the number of iterations.

The results after five iterations can be seen in Figure 15. For comparison, the length energy E_l (43) has been calculated for each vertex and is displayed through a scalar-mapping. The specifics of this mapping are explained in section 5.4.2.

It can be observed that there is a second cut introduced between the two leaflets at the posteromedial commissure (see the lower part of Figure 15). It increases the effect of the relaxation on the immediate area and visually separates the leaflets. The user can specify whether this cut should be performed.

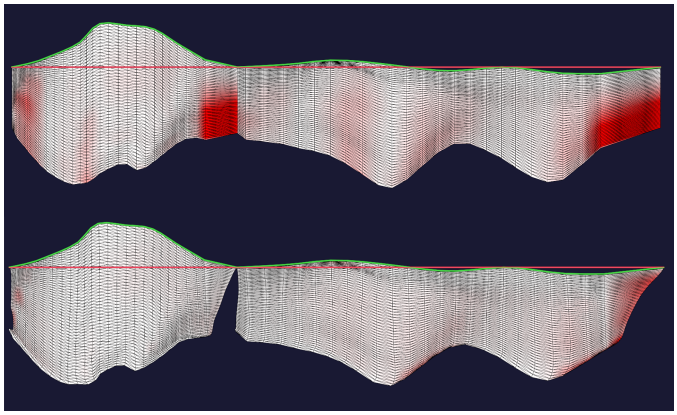

Figure 15: Per-vertex distance energies before (top) and after (bottom) relaxation. Red refers to higher distortions of edge lengths. Annulus shown in green and $u = 0$ shown in red. The mesh wireframe is visualized in black.

5.4 Scalar Fields Visualization

The final goal of this work will be the implementation of scalar field visualizations, which are computed on the valve surface. Scalar field values can be depicted through the use of color-maps. Roughly, this means assigning each vertex or triangle a scalar value, which can be derived in various ways. Then, certain scalar values are associated with specified colors and the colors of scalars with values in-between are linearly interpolated. The color information can be displayed on the model surface and the parameter space. It thus gives both, the 3D- and 2D-space, another dimension, which can be intuitively interpreted by a viewer.

The color-maps examined here can be grouped into the following categories:

- Color-maps to help strengthen the 3D/2D-correspondence. Applying the same color to a vertex in both views makes it easier to understand which parts of the valve coincide with one another. See section 5.4.1.
- Color-maps to visualize the distortions introduced by the parameterization. One of these mappings displays the distance distortion, which was already shown in Figure 15. See section 5.4.2.
- Color-maps that display medical information, which can facilitate mitral valve analysis. See section 5.4.3.

The user can quickly switch through these different visualizations. All remaining Figures in this chapter will show these color-maps in the way the developed software prototype displays them: The 3D model is on the

left and can be interacted with (the camera can be moved). The parameter space is on the right. The background is kept dark to provide a better contrast. Also, the annulus is shown in both views to further improve their relation and $u = 0$ is always displayed in red to provide a uniform reference line.

5.4.1 Localization Mappings

The first kind of mapping is used to visually underline the relation between the 3D and 2D view. If points are assigned the same color in both depictions, they are easier to locate. Two such mappings have been defined (see Figure 16). The first one shows all vertices belonging to the anterior leaflet in blue and all vertices belonging to the posterior leaflet in red. Apart from clarifying the 2D/3D-correspondence, this mapping allows easy differentiation between the two leaflets. The second mapping of this category displays a color gradient in direction of the u-coordinate. This means each vertex is assigned a color corresponding to its u-coordinate in parameter space. The leaflets are again separated by mapping the vertices of the anterior leaflet from red to green and the vertices of the posterior leaflet from green to blue. This color-map provides a clear depiction of the cut on the anterolateral commissure and makes it easy to locate in the 3D view.

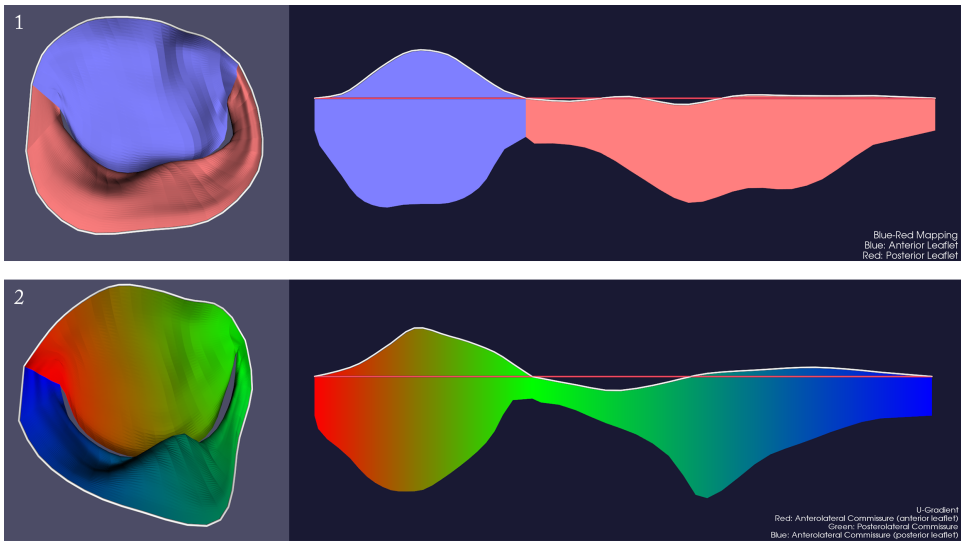

Figure 16: Localization Mappings.
1: The anterior leaflet is depicted in blue and the posterior leaflet in red.
2: Visualization of the gradient in u-direction. The anterior leaflet is shown from red to green and the posterior leaflet from green to blue.

In addition to the localization mappings, two other features have been implemented to help with 2D/3D-correspondence. The user can overlay the models with a texture that divides the parameter space into eight regions of equal u-intervals (see the upper part of Figure 17). The application of this texture map results in labeled vertical lines, providing a clear correspondence of regions between the two views. It can be observed that the posterolateral commissure is located at about $3/8$ of the flat leaflet's horizontal length. The texture can be activated on top of all color-maps.

A more direct feature to immediately localize specific points was implemented as well. A right click on either of the views provides a local marker at the center of the selected triangle in both the 2D and 3D model (see the lower part of Figure 17). Of course, the marker can be set independently of any currently active color-maps or textures. Selecting the background will remove the marker.

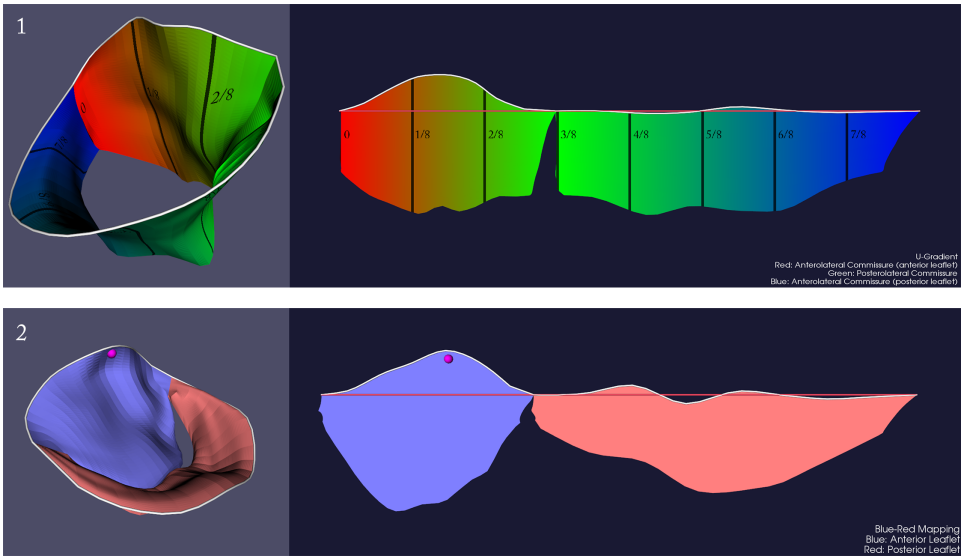

Figure 17: Further localization features.
1: A texture map that divides the valve into eight intervals of equal u-lengths.
2: A marker that can be placed anywhere on the 2D or 3D valve surface and always shows the same point in both views.

After consultation with cardiac surgery experts from the Heidelberg University Hospital, another labeled overlay, akin to the one shown in Figure 17 (1), was introduced. The leading idea was to approach a more common clinical partitioning of the heart valve, in order to simplify the analysis procedure performed by a physician, who is probably already accommodated with such a segmentation.

The classification scheme being used was introduced by Carpentier [11]. It divides each leaflet into three segments, accordingly labeled A1 to A3 for

the anterior and P1 to P3 for the posterior part. The division can be made using a closed valve by finding the one- and two-third points of the coaptation line. Connecting both points with the anterior saddle horn and projecting the two resulting lines onto the leaflets creates the separation between the segments of each leaflet. As the coaptation line is not always well-defined, a computation based on this approach is not simple, especially when the valve is open. Therefore, the implementation uses a texture-based approach, relying on the placement of already defined uv-coordinates.

In the 2D-view the leaflet models are treated separately. Each leaflet is normalized in direction of the u-gradient and translated such that the annulus point with the largest v-coordinate is set to $v = 0$. Then, for the anterior leaflet, the separation lines' origin is chosen at the normalized coordinate of $(u, v) = (0.5, 0.0)$. The lines are drawn towards the one- and two-third length of the u-gradient. For the posterior leaflet, the lines are "extended" along its one-third and two-third u-value, resulting in a vertical division in the 2D-view.

This method of applying the partitioning is easier to comprehend when looking at the result, which can be seen in Figure 18. Naturally, the division lines of the two leaflets will not always meet at the same coaptation point. However, this uv-based approach provides a uniform way of characterizing the valve's segments, including an intuitive labeling and should provide a straightforward way to understand the correspondence between the 3D- and 2D-view.

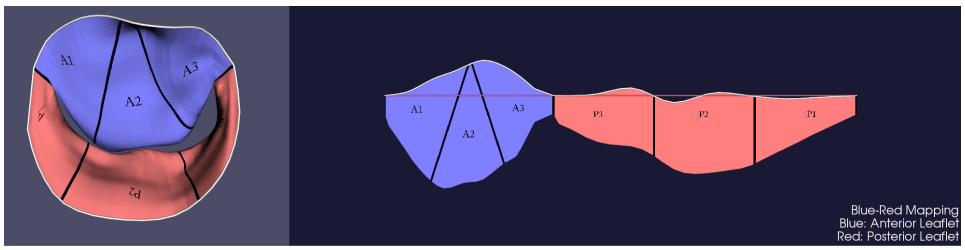

Figure 18: A localization feature that also incorporates a typical clinical classification scheme of the mitral valve. It is computed through appliance of per-leaflet texture maps based on the created uv-coordinates.

5.4.2 Distortion Mappings

As described in chapter 3, parameterizations inevitably introduce distortions. These deformities can be measured by different energies, which can also be visualized. The software prototype can display several distortion mappings, which can help to determine the quality of parameterizations. Areas of the 2D view, which are greatly stretched or deformed, might be misleading during valve analysis. Therefore, a distortion color-map may provide a physician with valuable information on the correctness of the

current view. Three types of distortion have been analyzed: Area, angle and edge-length. Their respective mappings are computed as follows:

Area Distortion. See Figure 20 (1). Let's assume the 3D mesh to be composed of triangles $\mathcal{T} = \{T_1, \dots T_n\}$. If the surface of a triangle is determined through $A(T)$ and each triangle T_i can be bijectively mapped to a corresponding triangle $\bar{T}_i$ in parameter space, then the area distortion can be computed per triangle by examining their proportions:

$$E_A = \frac{A(T_i)}{A(\bar{T}_i)} + \frac{A(\bar{T}_i)}{A(T_i)} \tag{45}$$

This energy was originally proposed by [15] and was already introduced in terms of singular values in section 3.3.3 in the second part of Equation (39). The minimum of E_A is at 2 and the energy is equal if the triangle's area in parameter space is enlarged or shrunk by the same factor (e.g., a doubled area would lead to the same energy as a halved area). The visualization takes into account whether the area is too large or too small and uses a red or blue color-map respectively. An energy of $E_A \leq 2.0$ is mapped to white and an energy of $E_A \geq 3.0$ is mapped to blue (shrinkage) or red (enlargement). Results in-between are linearly interpolated. This energy range is globally chosen, because individual definitions based on the values of specific valves would make the visualization results less comparable. The lower threshold 2.0 is an obvious choice, as it is equal to the minimum possible energy. The upper threshold 3.0 is equivalent to an area stretch with factor 2.6, since $2.6 + 1/2.6 \approx 3$. It is based on observations of the data sets and chosen in a way that most sets allow a differentiation of the color spectrum. This leads to clearly identifiable regions of higher distortion. This mapping scheme is applied to the following color-maps as well.

Angle Distortion. See Figure 20 (2). This second distortion measure is technically independent from area distortion, though it can be observed that stretched regions usually exhibit a higher amount of both, area and angle energies. Still, for the sake of completeness, angle distortion can be visualized separately. For this purpose the MIPS energy is used, as proposed in [28]. It was already reviewed in section 3.3.3 and expressed using singular values in Equation (38). It can be written in terms of distances and angles using the notation of Figure 19:

$$E_M = \frac{\cot\alpha|a|^2 + \cot\beta|b|^2 + \cot\gamma|c|^2}{2A(T_i)}. \tag{46}$$

The minimum of E_M is at 2 as well, which will only be reached if the triangles T_i and $\bar{T}_i$ have equal angle-to-edge proportions. The energy is computed per triangle and then mapped via a red color-map. $E_M \leq 2.0$ corresponds with white and $E_M \geq 5.0$ corresponds with red. Values in-between are linearly interpolated.

Distance Distortion. See Figure 20 (3). The isometry of a mapping can be measured by examining the edge lengths of the meshes. For this purpose the energy shown in Equation (43) is computed per vertex. A color-map is used, where white corresponds to the minimum energy of $E_l \leq 2.0$ and red to a value of $E_l \geq 2.2$.

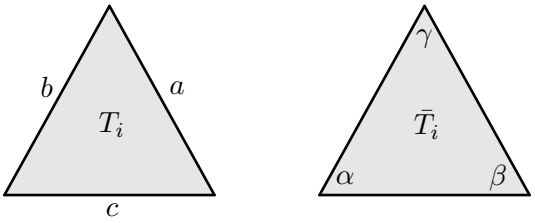

Figure 19: A triangle T_i on the surface and its counterpart $\bar{T}_i$ in parameter space. Notation used in Equation (46).

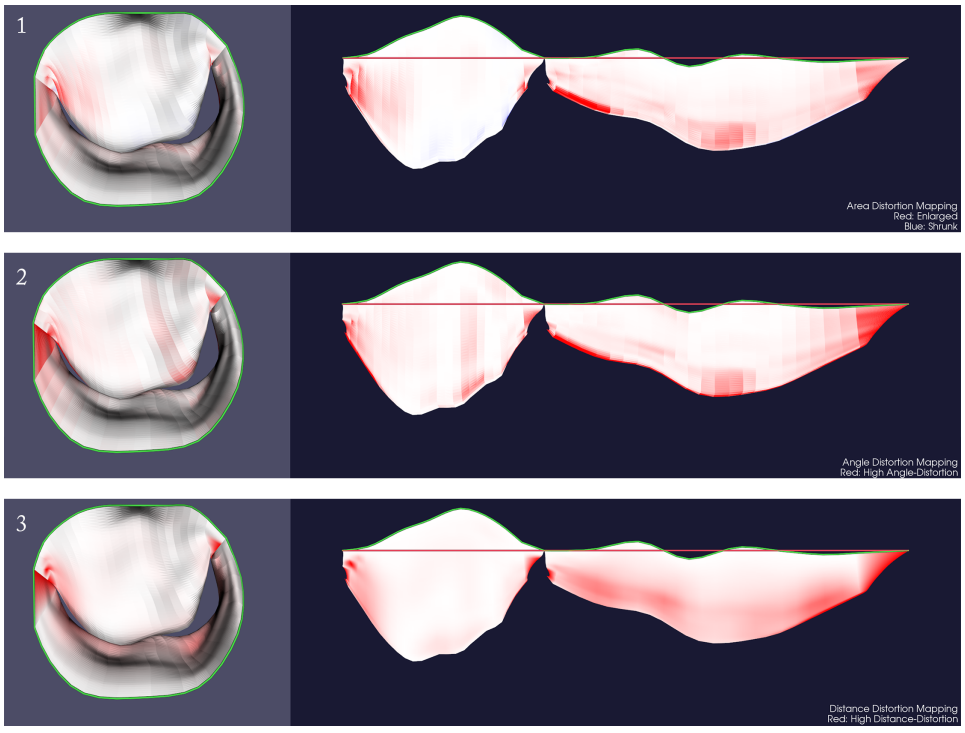

Figure 20: Comparison of different distortion mappings.
1: Area distortion energy E_A.
2: Angle distortion energy (MIPS) E_M.
3: Distance (edge-length) distortion energy E_l.

In addition to these color-maps, the implementation can also display a *checker-pattern* through the application of a surface texture, as shown in Figure 21. This is a typical approach to visually asses the quality of a parameterization, as it gives clues to all kinds of distortions. Area distortion is shown, as the individual rectangles will be larger or smaller than they should be when applied to the original surface. Angle distortion can be observed as well, because the patches will deviate from their rectangular form.

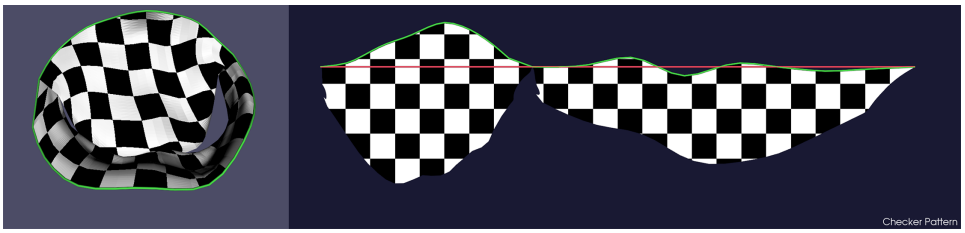

Figure 21: Checker-pattern mapping.

5.4.3 Valve Pathophysiology Mappings

While an uncomplicated localization of points and a low distortion amount of the parameter space are important aspects of the 2D view, they might only indirectly facilitate the process of medical mitral valve analysis. For direct assistance in the detection of pathological indicators, two additional types of color-maps are now introduced.

In the original work, where the segmentation algorithm for the used valve models was proposed [17], a visualization focus was the depiction of the coaptation zone. The coaptation zone is the area where the two leaflets touch during systole. As described in chapter 2.2, a common pathology of the mitral valve is insufficiency or regurgitation. This dysfunction describes a non-physiological backflow of blood during systole, because the valve does not close completely. Analysis of the coaptation zone can thus show which part of the valve is responsible for the leakage. Also, a narrow coaptation zone might indicate increased risk of a possible regurgitation. A visual representation of the coaptation zone might help in assessment and procedure planning.

As the valve's geometry is not volumetric and the surfaces are extracted from the center of the tissue, the 3D leaflet models do not actually touch. However, a distance threshold can be defined that approximates when the two leaflets are closed. For each 3D point the closest distance to the opposing leaflet is calculated by selecting the nearest vertex of the other leaflet. If the distance is lower than a default threshold set to 2 millimeters, the point will be marked to be inside of the coaptation zone. The color-map will apply a green color to all these vertices and a yellow color to all vertices

within the doubled threshold. All other points will be displayed in white. The user is given the option to manually increase or decrease the threshold via a slider. Figure 22 shows the same valve and its parameterization twice with different manually chosen coaptation thresholds.

While this kind of color-map can also be evaluated in the 3D domain, it is vastly less troublesome to quickly assess the 2D view. In the 3D model the valve can be rotated to expose different parts of the coaptation zone, but some areas will always be occluded by geometry. Take for example Figure 23, which displays a healthy mitral valve at the top and a pathological, prolapsed mitral valve at the bottom, using the same coaptation threshold. In the current perspective of the 3D view it is not possible to see whether the pathological valve is completely closed or not. The 2D view gives immediate indication of areas with a high risk of leakage, where the lower rim of the leaflet is not covered in green (and partly not even yellow).

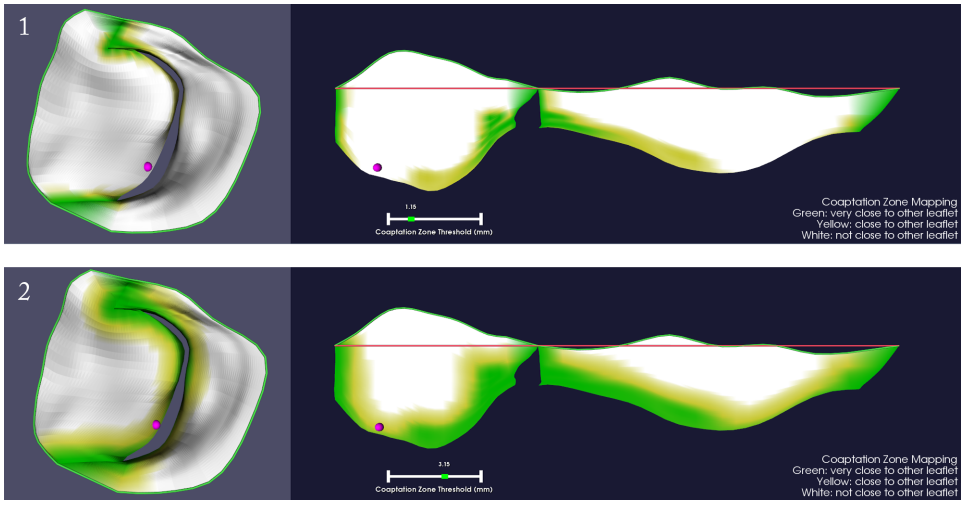

Figure 22: An mitral valve shortly before systole with visualized coaptation zone.
1: Low coaptation thresold.
2: High coaptation threshold. Note the area around the purple marker, which might either be already closed or still slightly open.

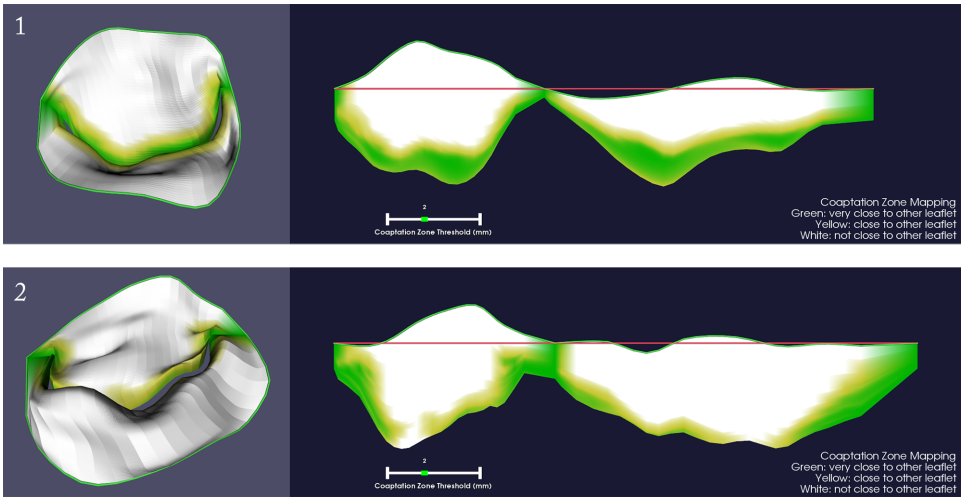

Figure 23: Visualized coaptation zone of mitral valves during systole.
1: Healthy valve - the 2D view displays a large approximate coaptation zone without leaks.
2: Pathological valve - the 2D view shows discernible parts of the valve that are still open.

A second mapping with a similar objective was implemented as well. Since the annulus plane is already approximated to perform the annulus parameterization, it can also be used to compute a valve property called *tenting*. Tenting usually coincides with a valve prolapse, which might in turn lead to or indicate regurgitation. A non-pathological closed valve usually exhibits a kind of "V"-shape, where the annulus plane lies above the leaflets. A prolapsed mitral valve will have parts of the leaflets above the annulus plane, which can manifest in a tent-like shape. This property can be used to quickly identify pathological valves when color-mapped onto the leaflets.

For each point the signed distance to the approximated annulus plane is computed. Vertices with a distance greater than 3mm below the plane are mapped to white, vertices directly on the plane are mapped to yellow and if they are more than 3mm above the plane they are mapped to red. As usual, values in-between are linearly interpolated.

The results can be seen in Figure 24. It can be observed that the prolapsed mitral valve (3) has a large part of its leaflet area located above the annulus plane. The healthy valves (1) and (2) only contain some area around the anterior saddle horn, which is above the annulus plane and thus marked in red. It should be made clear that this section of the valve is supposed to be above the plane, in contrast to the leaflets' lower parts, which are usually located below the annulus plane. Such a 2D view could for instance be used to visually compare the prolapsed area before and after a surgical intervention.

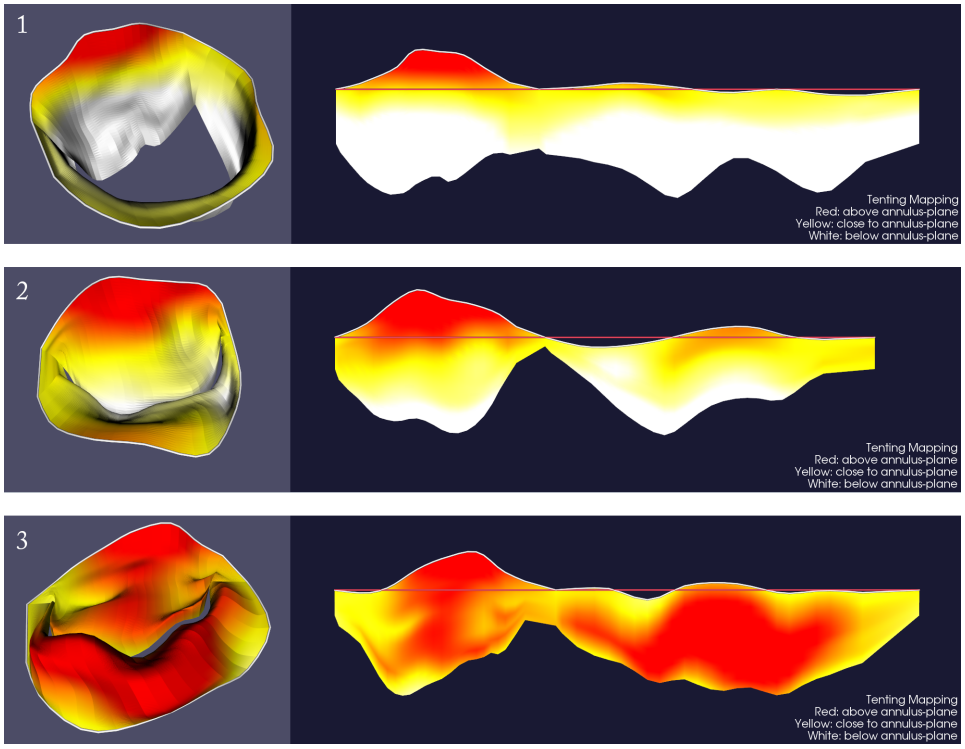

Figure 24: Tenting visualization of different mitral valves.
1: Healthy mitral valve during diastole.
2: Healthy mitral valve during systole.
3: Pathological mitral valve during systole.

6 Implementation

Until now, a general description of the developed algorithms has been provided. Implementation details have only been touched upon; for instance, color-map thresholds have been declared and the look of the resulting visualization has been depicted.

This chapter will further describe the developed software prototype. The code structure will be outlined. Used libraries will be explored and guidance for a possible future adaptation or improvement of the prototype will be provided. The focus will lie on the technical side, yielding a rough documentation of the actual implementation. Firstly, the VTK will be further elucidated, as it is *the* baseline tool used in this work. After that the implementation's classes and their most important methods will be portrayed one-by-one.

6.1 VTK Functionality

On its website, the Visualization Toolkit (VTK) is described as an "open-source, freely available software system for 3D computer graphics, image processing, and visualization" [33]. It can be used as a library tool to visualize various kinds of data, including meshes, scalars, vectors, tensors, textures, and volumes. It also provides an array of advanced modeling techniques, such as mesh smoothing or cutting. Its implementation uses an object-oriented approach and is natively written in C++, although wrappers for other languages, like Python and Java, exist.

As it is possible that the developed software prototype might get included in a MITK plugin in the future, compliance with VTK data formats, used by the MITK, is an objective. The prototype developed in this work is build upon the VTK, in order to provide the best integration with the segmentation and visualization pipeline applied to create, store and display the used mitral valve models. The VTK is deployed as a C++ library, which serves as a backend for various tasks. The most important uses are:

- The VTK executes the final visualization, for instance it manages cameras, lights and shaders.
- The VTK is used for loading and saving data types. This includes the valve models and textures.
- It provides an interface for several mesh processing tasks, for example the arrangement and transformation of vertices. Furthermore, VTK's internal data structures are used to store additional information on the rendered objects, such as per-vertex lists containing computed scalar information. Another benefit of the mesh structure is its automatic differentiation of vertices, edges and triangles of a mesh,

all of which can be iterated independently. This is for example used to find the one-ring neighborhood of vertices.

- The VTK also brings along some math functionality, including functions for trigonometry purposes and vector operations. Math classes were for example used to find covariance matrix eigenvalues, during the best-fit-plane algorithm.

The VTK classes are commonly instanced in an encapsulating pointer-object called a `vtkSmartPointer`. Internally, objects are thus only passed via pointers, which accommodates for structures with large internal arrays, such as triangulated meshes. In a typical visualization scenario `vtkSmartPointers` to all objects seen in Figure 25 are initialized, then the objects are connected to form the displayed pipeline. Each `vtkPolyData`-object contains information on an individual single mesh, which needs to be loaded or created. Then its `vtkActor` can be added to a scene (i.e, a `vtkRenderWindow`). A benefit of this layout is that parts of the pipeline can be exchanged or modified along the way, while every object up the chain is automatically updated.

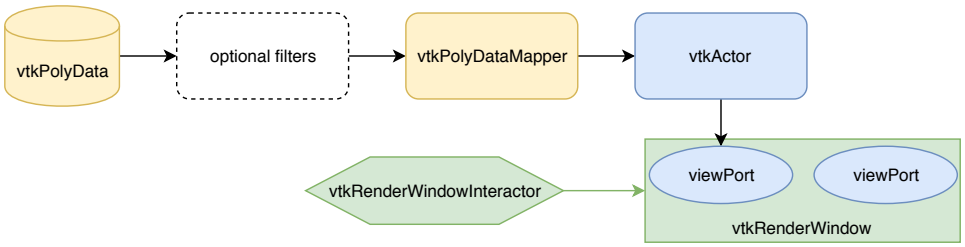

Figure 25: The typical VTK visualization pipeline. Filters are optional and can be concatenated at will. Only actors can actually be rendered in a scene.

In the prototype the 3D leaflets for a single time-step and their counterparts in 2D are represented each by a `vtkPolyData`. Color information, derived from computed scalar fields, can be added directly to the vertices or cells (triangles) of the mesh. If enabled, filters are applied to perform additional tasks, such as extraction and individual display of the annulus. The used `vtkRenderWindow` is comprised of two view-ports, one for 3D and one for 2D respectively. A second window can be instantiated in a new thread, in order to make direct comparison of two leaflets possible. A subclass of `vtkRenderWindowInteractor` is created, which is used to deploy supplementary mouse and keyboard interaction.

6.2 Class Structure

The software prototype is build in a way that easy utilization of individual components is possible. Classes are deployed for specific tasks, minimizing

dataflow across interfaces. Thus, a future adaptation could make use of parts of the implementation, while others are replaced or omitted.

In the current development state, the prototype can be executed via a command-line interface, which takes up to two parameters. Each parameter is interpreted as a path (local or global) describing a directory where the valve data-sets are stored. Each given path will lead to the execution of an individual thread of the program. This way two valve data-sets can be compared. It is also possible to display different time-steps of a single data-set, by passing the same path twice. If no path is supplied, a default directory will be used.

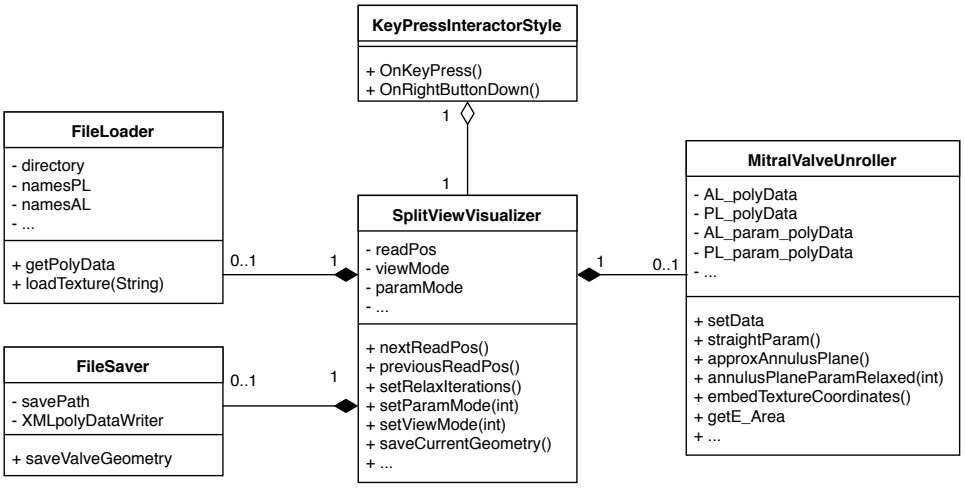

Figure 26: Class diagram of the prototype's components and their key fields and methods.

The program is split into two major components: a *Visualizer* and an *Unroller* (see Figure 26). The *main* instantiates the program threads and manages the windows, as well as the command-line parameters. Everything else, which is related to the display of the mitral valves, is handled by the *Visualizer*. The parameterization and all other mesh processing tasks are performed by the *Unroller*.

The *Visualizer* initializes a split-view of the 3D model on the right and the 2D representation on the left. It includes the following functionality through public methods:

- Incrementing or decrementing the current time-step. This will load the respective 3D model from the specified directory (if a model exists). Its parameterization will be automatically computed by the associated *Unroller*.

- Additional screen-space elements can be toggled on or off. This includes geometry to underline the annulus, the u-axis in the 2D-view

and a visualization of the computed annulus-plane in the 3D-view. Internally, this will simply trigger the respective `vtkActor` to be displayed or removed, therefore a re-computation of these objects is not necessary.

- The mode for the parameterization and the number of relaxation-iterations can be set. The computation will be handled by the *Unroller*.

- The view-mode can be changed. The possible modes include the color- and texture-maps described in section 5.4. The color information is also computed within the *Visualizer*. Additionally, localization features such as the u-intervals and the point-picker (see section 5.4.1) are visualized from here.

To provide a separation of the mesh processing operations (most notably the parameterization) from the visualization, the *Unroller* is tasked with these computations. The *Visualizer* and the *Unroller* share pointers to the four valve models (one for each leaflet in 3D and 2D). Therefore, these large objects do not need to be individually passed between classes. The *Unroller* provides the following functionality through public methods:

- Before any operations can be executed a `setData`-method needs to be called. This is necessary, because each time-step provides its own valve geometry, which needs to be assessed first. The *Unroller* builds a matrix-like structure composed of the meshes' vertex-IDs, sorted along topological longitudes. This allows quick navigation within the geometry, which is for example used during the traversal of vertices on the annulus.

- Different parameterizations can be computed, for instance, a "straight" parameterization can be performed, where the annulus is mapped to $u = 0$ or the final parameterization method can be used.

- A pointer to the annulus plane used during the parameterization can be returned for visualization purposes.

- Different distortion-energies (area, angle, length) can be computed per vertex or triangle.

- The resulting uv-coordinates of a parameterization can be embedded within the `vtkPolyData` of the leaflets.

As the class-diagram in Figure 26 reveals, additional functionality has been transferred to some minor classes. A *Loader* manages the reading of valve geometry stored in standard VTK-files. It creates `vtkPolyData`-objects and passes the respective pointers to the *Visualizer*. The same is

done with textures. Mesh-files within the target folder are distinguished via regular expression in order to comply with the naming convention of the used data-sets. Files with "AL" in their name are considered anterior leaflets and files containing "PL" in their name are handled as posterior leaflets. Other files are considered to be separate annulus-geometry and are not required for the parameterization, but can be displayed if needed. The alphabetical order determines which leaflet-files (anterior/posterior) correspond to the same time-step. Usually, this results in a correct pairing, since the files are labeled with ascending numbers.

Analogously, a *Saver*-class is able to write `vtkPolyData`-objects to a file, including all associated arrays, which can for instance contain computed uv-coordinates.

Direct user-interaction is facilitated through a subclass of VTK's `vtkTrackballCamera` called `KeyPressInteractorStyle` (this is the common naming convention). In addition to the simple camera control of the parent class, it provides an interface for key-mappings, enabling control over all public *Visualizer* methods with real-time feedback. It also catches the right mouse-click used for point-picking. Apart from a slider used to change the coaptation-zone threshold, the implementation of a graphical user-interface was intentionally omitted. The presented classes are designed to be integrated in an already existing framework (like the MITK), which will bring its own UI-elements that can be attached to the *Visualizer* and *Unroller*.

7 Evaluation

The presented software prototype was developed with the intention to provide most optimal 2D representations of mitral valve models, facilitating the analysis process that takes place when assessing possible valve disease treatments or evaluating the success after an intervention. Although the prototype is still somewhat remote from practical usage (e.g., due to its lacking user-interface), an evaluation of the current development state should provide valuable information on the feasibility of the targeted utilization.

The evaluation is split into two parts: A technical review of the distortion energies measured and a survey, oriented to be conducted with physicians who might benefit from the proposed 2D-view. As a large-scale study of such a kind is a challenging task on its own, the scope of this thesis only allowed a limited test run of the designed evaluation. Nonetheless, acquiring the expert-impressions of individual heart-surgeons should prove beneficial during this state of development.

7.1 Measurements

Among all declared objectives, the technical goal was to keep distortions in the 2D domain, inevitably introduced by the parameterization, to a minimum. The main tool used to achieve this is a relaxation method which iteratively moves vertices to a more optimal position through edge-length comparison. The method is physically inspired, interpreting the mesh as a mass-spring system, similar to how cloth or semi-elastic tissue is modeled. If applied correctly, this should result in a natural relaxation of the 2D model, parallel to how a real valve would behave if it was flattened along its circumference.

In order to evaluate the effectiveness of the relaxation method, the three distortion energies introduced in Section 5.4.2 (area, angle and length distortion) were sampled across 50 mitral valve models taken randomly from 10 different data-sets. For every model, the arithmetic mean of each energy was computed across all polygons, once before and once after five relaxation iterations. The results can be seen in the boxplots depicted in Figure 27.

It can be observed that the edge-length (green) is clearly optimized, approaching the minimum of 2 after the relaxation. This means all edges in the 3D model exhibit almost equal length compared to their equivalents in the 2D model. A positive side effect of this optimization is that the area distortion (blue) also dramatically decreases, providing a potential user with the ability to quickly assess the leaflet area.

The only energy that is visibly not affected by the relaxation is angle distortion. This is to be expected, as the deployed optimization method

does not take the meshes' angles into account. For a user this means the mesh layout might slightly deviate from reality, even though the area is correctly portrayed. It is highly unlikely that an increased number of relaxation iterations would provide better results in this regard, seeing that the length distortion (which is the energy optimized by this method) is already almost at its minimum. Future adaptations could take different kind of energy-optimizers into account, which might result in divergent outcomes.

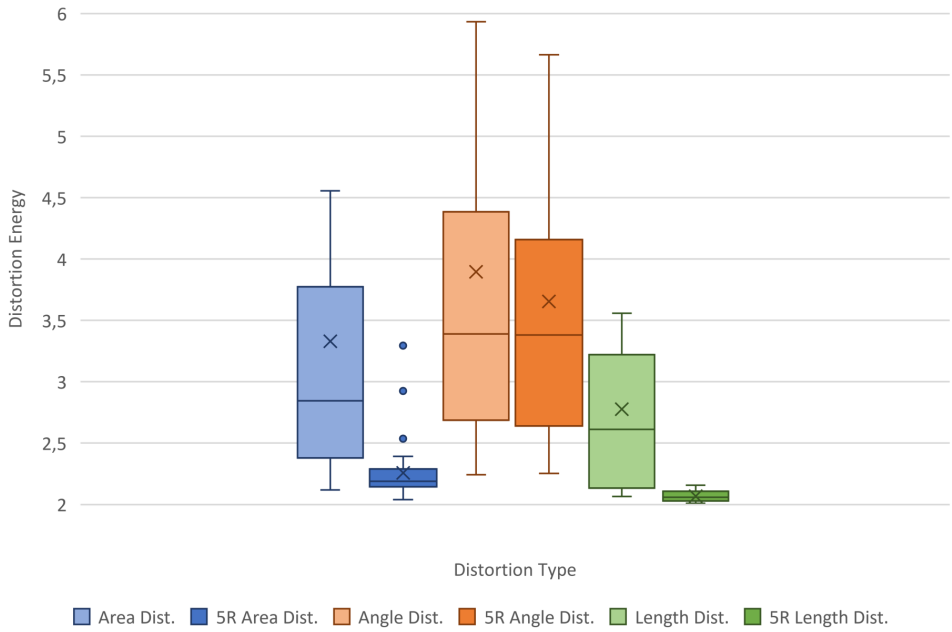

Figure 27: Boxplot diagram of different distortion energies before and after five relaxation iterations (5R) of the parameter domain. Sampling was performed on 50 mitral valve models taken randomly from 10 patient datasets.

A second issue to consider, when comparing these energies, is the effect different states of a valve might have. It has already been mentioned that the parameterization of open valves should yield less distorted results, as they resemble a cylindrical shape. For a closed valve the energetically best parameter space layout would contain an annulus shape similar to a half-circle. This is because in closed valves the coaptation zone's circumference is measurably smaller than the annulus' circumference. However, this would contradict another goal of this work, which is the easy comparability of all constructed 2D-views. Therefore the parameterization of the annulus is always done first and in the same manner, regardless of the state of the valve.

The results of this procedure can be viewed in Figure 28, where distortion energies are compared in opened and closed valves. Again, 10 mitral valve data-sets were used. For each set the closed (systole) and opened (diastole) states were sampled. Their arithmetic mean is shown, comparing both states (closed and opened) per distortion-type. All energies were measured after five relaxation iterations. It can be seen that all distortions are higher during systole (closed state), while especially angle distortion protrudes. This is probably because angle energy is not mitigated by the optimizer as much as the other energies.

All in all, these results show that during mitral valve analysis, using the proposed 2D-view, a physician should be able to rely on the distances along mesh edges and the displayed leaflet area.

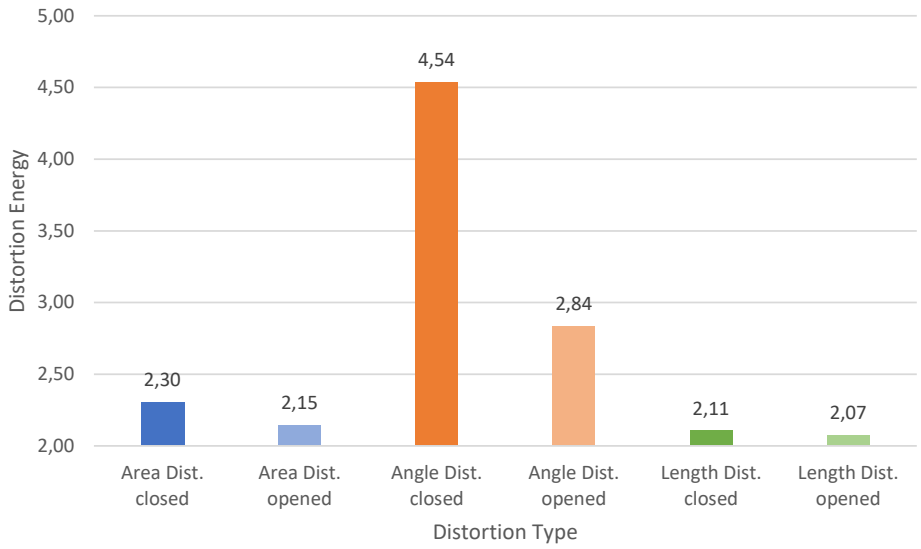

Figure 28: Average distortion energies of 10 mitral valve data-sets. For each of these sets, the closed (systole) and opened (diastole) states were sampled independently. States in-between were omitted.

7.2 Survey Design

The second part of the evaluation is aimed towards assessing usability and general usefulness of the proposed 2D-view. For this purpose a survey was created, which is designed to be conducted with the target audience, e.g., cardiac surgeons. This means a high degree of knowledge concerning mitral valve anatomy, pathologies and treatment possibilities is assumed. The participants were also already familiar with the 3D representation of the mitral valve. The goal of this survey is to find out whether the expected benefits of the 2D-view hold true and in which aspects the prototype can be improved. The survey is split into the following parts:

Introductory Video. First of all, participants are acquainted with the 2D-view. A short video presents the prototype and explains the design of the 2D-representation, as well as its possible advantages. Some of the developed color-maps are also introduced, especially the ones devised to help in pathology-analysis (see Section 5.4.3).

Point Localization. The participants are then presented with a task: They are shown ten images of the prototype, displaying the combined view, i.e., 3D on the left and 2D on the right. In each picture, one of the views contains a marked point. The participants are now required to localize this point in the opposing view. In order to measure how close they are to the actual position, a grid is used, from which they are to pick the cell they think most likely to contain the marked point. Figure 29 shows one of these images. This task should show how well the 2D/3D-correspondence can be understood. The images, presented to the participants, are all shown with the anterior leaflet in blue and the posterior leaflet in red. Half of the images do also display the (clinical) valve segments, as introduced in Section 5.4.1. This should provide results regarding the improvement of readability through a texture-overlay like this.

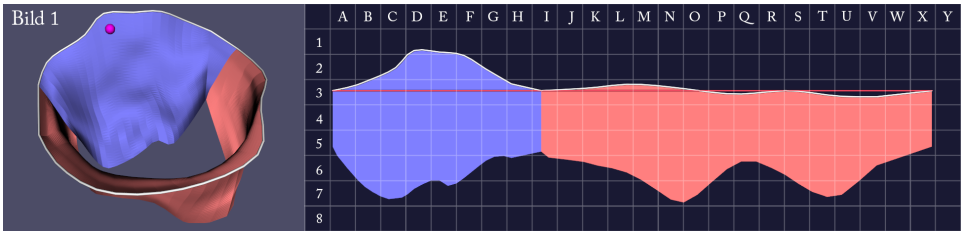

Figure 29: Image as it can be found in the first task of the survey. Participants are required to pick the cell they think most likely to contain the marked point.

Pathology Identification. At this point, the participants should be roughly familiarized with the 2D-view. They are now given a second assignment, which is more closely related to actual valve analysis. For the purpose of this task the developed prototype was modified to provide a controlled environment. The participants are to classify 40 mitral valves and assign each of them one of three pathological categories: normal, prolapsed or functional mitral regurgitation (FMR). The valves will be subsequently presented to the participants in two *alternating* formats. Half of the models will be shown in a 3D-only-view, displayed in a grey color. The other half will be shown in the combined 3D/2D-view, as is the current state of the prototype. In this view, the participants will also have the option to choose between the tenting- and coaptation-color-map (see Section 5.4.3), as these are designed to directly facilitate valve analysis. Here, the segments of the valve are displayed as well. All 3D models, regardless of the view-format,

can be interacted with (moved, rotated and scaled). What the participants do not know is that there are only 20 distinct mitral valve models. Each model will actually be shown twice, once in the 3D-only-view and once in the combined view, in a randomized order. This makes a direct comparison of the participants' actions, regarding specific valves, in both views possible. All shown models were pre-classified by experts at the DKFZ.

For each pathological classification the participants are also instructed to rate their own confidence on a scale from one (not confident) to five (very confident). The entry of the classification and confidence is done after the participant decides he or she has concluded the analysis of the currently shown valve, by pressing the *enter*-key. The time required for the analysis is measured in the background for each valve. Only after pressing *enter* again, the next valve model is shown. The first two valves (one for each view-format) serve as an example to help the participants get used to the program controls.

Questionnaire. The last part of the survey requires participants to fill out a small questionnaire. They are asked to submit their agreement on specific statements regarding usability and usefulness of the prototype. For this a scale from one (strongly disagree) to five (strongly agree) is provided. They are also given open questions referring to what they think could be possible applications of the 2D-view and how the prototype could be improved in the future.

7.3 Survey Results

The survey was conducted on two participants at Heidelberg University Hospital: an expert in valve segmentation and visualization and a cardiac surgeon. Scope and timeframe of this thesis did not allow a more ambitious execution of the proposed survey. Nonetheless, the results should provide valuable feedback for possible future adaptations and improvements of the 2D-view. All results will be listed now, then discussed.

Avg. euclidean distance	0.33
Avg. distance with segments off	0.74
Avg. distance with segments on	0.0
Range of results	0.0 - 2.24

Table 1: Results of the first task (point localization).

Accuracy	3D-view only	3D/2D-view	Change
Normal	75%	58%	- 17%
Prolapsed	79%	92%	+ 13%
FMR	0%	50%	+ 50%
Total	**74%**	**79%**	**+ 5%**

Table 2: Results of the second task regarding classification accuracy. The table shows the percentage of correctly classified mitral valves, comparing the two view-modes. The column *change* shows the difference between the two view-modes.

Time	3D-view only	3D/2D-view	Change
Normal	12.17s	22.10s	+ 9.93s
Prolapsed	11.78s	12.10s	+ 0.31s
FMR	24.50s	18.00s	- 6.50s
Total	**12.79s**	**15.56s**	**+ 2.77s**

Table 3: Results of the second task regarding time. The table shows the average number of seconds participants required to analyze the mitral valves, comparing the two view-modes. The column *change* shows the difference between the two view-modes.

Perceived Confidence	3D-view only	3D/2D-view	Change
Correct classification	4.39	4.53	+ 0.14
Incorrect classification	3.40	3.63	+ 0.23

Table 4: Results of the second task regarding the participant's perceived confidence on a scale of 1 to 5. The two view-modes are compared for correctly and incorrectly classified valves. The column *change* shows the difference between the two view-modes.

Outcomes of the point-localization task can be found in Table 1. Euclidean distances between the picked and the correct cell were measured. It can be seen that the average distance is below 1, which means participants chose the correct cell most of the time. All wrong picks are within an euclidean distance of 2.24 (i.e., a maximum of 2-3 cells away from the target), leading to the assumption that all participants gained an understanding of the 2D/3D-correspondence. Furthermore, it can be seen that valves, which also displayed the segment-texture, were evaluated correctly in all cases (average distance of 0.0). This is a strong indicator that the implemented localization-feature does facilitate intuitive understanding of the 2D-view.

For the second task, which focused on the identification of mitral valve pathologies, several factors were measured. The most important aspect in this regard is classification accuracy, i.e., the ratio of correctly categorized

pathologies. The results can be seen in Table 2. Both participants showed a slightly higher classification accuracy (about 5% on average), when they had the 2D-view and color-maps available.

Prolapsed valves appear to be easily identifiable, especially in the combined view-mode, where over 90% were correctly recognized. Looking at the example in Figure 30, one can see two reasons why accuracy might have increased when the combined view (right) was displayed: Open parts of the valve are identifiable at a glance using the coaptation-color-map (top right). The lower rim of a closed valve should be continuously green. Additionally, large red areas in the tenting-color-map (bottom right) are an indicator for a prolapse, as they highlight parts of the valve that are above the approximated annulus-plane. Both these aspects could be observed in the colorless 3D-model alone, yet only if it is correctly rotated.

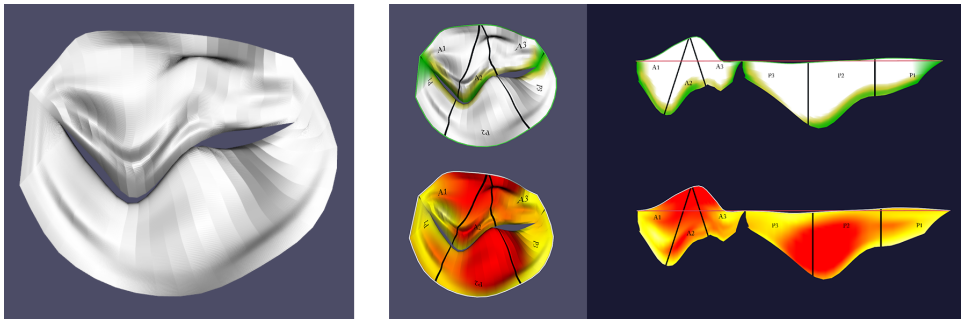

Figure 30: A prolapsed valve as it appeared in the second task in 3D-only (left) and the combined view (right).

It was difficult for participants to differentiate between normal valves and functional mitral regurgitation (FMR). No FMR was correctly classified in the pure 3D-view and the accuracy of declaring a valve to be normal was only about 60%, when the combined view-mode was displayed. These results probably suffer from a low sample size, as there was only a very small number of valves from the FMR category included in the evaluation data-set. Still, the combined view might provide an advantage when dealing with FMR, as the coaptation-color-map makes open regions of the valve visible in the 2D-view, while their severity might be difficult to assess in 3D only (see Figure 31). However, the 2D-view in combination with color-maps must also be interpreted with care, since it might lead to false positives. This can be seen in its 17% lower detection accuracy for normal valves. Significant assertions would require the study to be conducted with a higher sample size of valves or more participants.

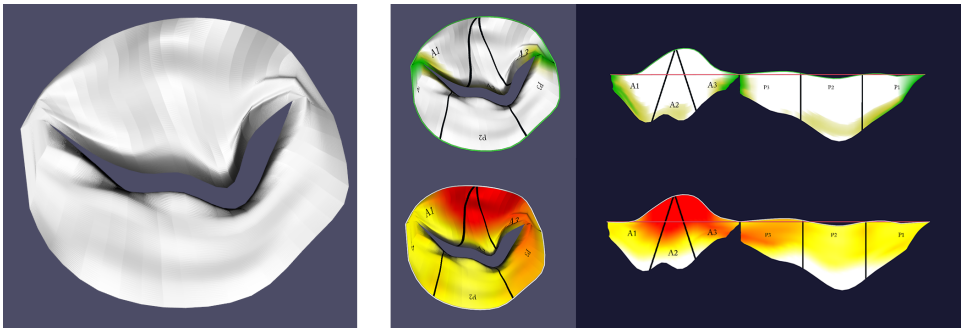

Figure 31: A valve suffering from FMR as it appeared in the second task in 3D-only (left) and the combined view (right).

The second measurement assessed the time it took participants to evaluate individual valves. One of the predictions made earlier claimed the 2D-view could save time, as physicians might be able to identify pathological indicators more quickly, when the need for a variety of viewing angles was omitted. However, the results (see Table 3) show that participants actually required a few more seconds to analyze the valves when using the combined 3D/2D-view. While the difference in time is almost negligible, it still raises the question why the combined view did not enhance the time required for analysis. A possible explanation is that the 2D-view, including the switchable color-maps and valve-segments visualization, exposes the user to an increased degree of information, which results in a longer processing time. The largest difference can be observed with non-pathological valves, reinforcing the assumption that the 2D-view might make it more difficult to correctly classify normal valves.

Participants were also asked to specify their confidence with each classification on a scale from 1 (not confident) to 5 (very confident). The results can be seen in Table 4. Comparison between the two view-modes shows that participants were slightly more confident with their decisions when using the combined view. This trend, however, holds true for both, correct and incorrect classifications.

After the second task was concluded, participants filled out a questionnaire, which mainly consisted of providing approval-ratings for specific statements concerning the 2D-view. A scale from 1 (strongly disagree) to 5 (strongly agree) was given. Each statement belonged to one of the following categories:

- The correspondence between the 3D- and 2D-view is clear (average rating: 5.0).
- Different valves result in comparable 2D-views (average rating: 4.0).
- Pathologies of the mitral valve are easier to identify when the 2D-view is provided (average rating: 4.5).

- Analysis of the mitral valve is facilitated through the 2D-view (average rating: 4.5).

As all statements were positively viewed, with approval-ratings of 4 to 5, it is safe to say that the prototype was well received. A possible enhancement of the study design would be to incorporate more direct assertions within this questionnaire, since the statements used were of a more general nature and thus do not provide feedback on individual aspects of the prototype.

The last part of the survey posed some open questions regarding possible appliances and improvements of the 2D-view. Participants stated they would benefit from the proposed view, especially during valve analysis, procedure planning and when comparing valves before and after surgical intervention. They also brought up features they would like to see, which mainly included quantifications of different aspects, such as area of the leaflets, height of the saddle-horn or length of the annulus. This would be easy to realize in a future adaptation of the prototype, as these values only need to be read from the already computed models and displayed on screen.

8 Conclusion

This chapter concludes the thesis at hand. It features a summary of this work (section 8.1) and provides an outlook on possible future enhancements and adaptations (section 8.2).

The core of this work was the development of a software prototype, designed to aid the pathophysiological analysis of patient-specific 4D mitral valve models through visualization of a flattened 2D version of the valve's leaflets. Such a view allows an inspection of the valve surface, without the need for different viewing angles and the drawbacks that come with perspective projection. The final prototype displays an interactive 3D- and 2D-view of the mitral valve, where the latter is generated per model in real-time, through methods of global parameterization.

8.1 Summary and Contributions

After a thematic introduction in Chapter 1, Chapter 2 comprises the medical background, including heart anatomy, physiology and functions of the heart valves, in particular the mitral valve. Possible malfunctions of the mitral valve are reviewed as well. This includes mitral valve prolapse, a condition that describes a non-physiological bulging of a leaflet into the left atrium, which can lead to mitral regurgitation (a "leakage" of the valve). Mitral regurgitation is the most common pathology of this particular valve and often requires complex surgical intervention.

Chapter 3 introduces the technical side of background information necessary in order to understand the development and implementation of the 2D-view. Surface parameterization is reviewed as a general concept, applied to a variety of mesh processing tasks. The typical goal of parameterization in a mesh processing context is creating a mapping that allows describing points on a mesh's surface in 3D with only two coordinates, commonly labeled uv-coordinates. Aspects of differential geometry are summarized as a toolset used to describe and perform parameterization. It is explained that the targeted parameterization will inevitably introduce distortions, which means the proportions of the surface will differ in 3D and 2D. Furthermore, it is shown that distortion can be divided into area- and angle-distortion. In the following sections, different established methods for parameterization of surface meshes are outlined, including *barycentric mapping*, *conformal mapping* and approaches based on *distortion analysis*.

Chapter 4 concludes the foundation of this work. Drawing on the previously discussed medical and technical backgrounds, a composition of the two is introduced through a survey of works, where parameterization was used in context of medical visualization. Last but not least, approaches of segmenting 3D heart valve models from medical image modalities are reviewed, as these methods are a direct requirement for this work's ad-

vances. The 4D data-sets used here were segmented at the DKFZ using transesophageal echocardiography scans of the mitral valve.

Chapter 5 presents the concepts of the developed parameterization and visualization, independent from a specific implementation. It provides a preliminary overview of the used data-sets and the objectives targeted by this thesis (i.e., the basis and goals of this work). Then, the parameterization method is explained in detail. It is composed of three steps: First, the annulus is parameterized as a curve, while relative distances to an approximated annulus-plane and its arc length are kept. Only then are the leaflets flattened in 2D. A relaxation method is applied, inspired by methods of simulating the physical behavior of semi-elastic tissue or cloth. This approach provides a way of minimizing distortions introduced by the parameterization.

After the transformation into uv-space is explained, section 5.4 describes visualizations, which are able to show different properties of the valve, characterized by per-vertex scalars and displayed through specified colormaps. These can help facilitate 3D/2D-correspondence, visualize areas of high distortion and convey medically relevant information. Two colormaps, belonging to the last category, were developed. One of them displays the severeness of prolapsed leaflet parts, the other marks the area where the two leaflets touch, making it possible to see whether the valve is completely closed or not at one glance.

Chapter 6 delves deeper into the actual implementation and prototype design. The Visualization Toolkit (VTK) is the main tool used for data reading, display and saving, in order to comply with the data formats and execution pipeline used by the DKFZ. From a software engineering perspective, the prototype is split into functional parts regarding file-handling, visualization and parameterization. Data-flow is minimized across interfaces of these parts. They can be addressed independently and could be substituted or utilized individually in a future adaptation.

Chapter 7 assesses the accomplishments actually made by the developed prototype in two evaluations. The first one relies on measurements only, quantifying introduced distortions, working under the assumption that a less distorted 2D-view provides a more realistic and thus better representation of the real-world valve. It could be shown that the implemented method minimizes distortions of triangle area and edge-lengths, however, distortion from a perspective of mesh angles could not be further enhanced. Open valves appear to exhibit a generally lower distortion than closed valves.

The second evaluation consisted of a survey conducted with experts at Heidelberg University Hospital. Due to the low sample size, results should only be interpreted as preliminary indications. They appear to hint at a slightly increased pathology detection accuracy, especially in prolapsed valves, when using the developed prototype, instead of a simple 3D-view.

One of the overarching goals of this work was to achieve visual correspondence between 3D- and 2D-views of a valve. In the conducted survey, users found this criteria to be satisfied, which was matched by the results of a point-localization task.

8.2 Future Work

Although the software developed in this work is fully functional and usable, it lacks certain qualities associated with a finished product. It does not contain a typical user-interface nor an integrated file-manager. For the most part user interaction is restricted to key strokes and the mitral valve models need to be located inside a specified folder. Therefore, the implementation is always referred to as a *prototype*. In the foreseeable future, parts of the developed classes could be included within the MITK as a plugin or extension, providing a 2D-view of mitral valve segmentations.

The visualization itself could still be improved as well. As already mentioned, quantifications of different values could provide further information during pathology analysis. Such measurements could include the area of individual leaflets or the area of valve surface above and below the annulus plane. It might also be beneficial to provide the option to mark several points on the valve and display the distance between them, in order to facilitate the assessment of gaps during systole. Similar ideas include showing the height of the saddle-horn (compared to the annulus-plane) or the annulus' length. The 2D-view could incorporate a scale or ruler, extending the uv-axes, to make dimensions more distinctly comparable.

The focus on the cardiac cycle could also be enhanced. An integral pathophysiological analysis of the mitral valve is only possible when it is observed over time. While an animation of the valve in 3D is conceivable and could be extended to the 2D, this would be a very demanding project, as vertices on the valve do not unambiguously correspond to each other across different time-steps. It could be possible, however, to use the parameterization in an attempt to re-mesh the valve geometry and make it better suited for animation, as the constructed uv-space uses a partly landmark based approach. Several 2D-views could also be blended to one image when stacked upon another. This could provide cues when identifying variations of, e.g., leaflet area or annulus shape over a cardiac cycle.

In more general terms, different parameterization methods could also be tried and tested. The method proposed here reduces area-distortion more than angle-distortion. It could be interesting to compare the results of different parameterization algorithms, for example, approaches more focused on angles. At the beginning of this work a barycentric method was tried, which resulted in a different 2D-view, more akin to a closed valve. An example can be seen in Figure 32. Additional geometry is added to the valve, which is then mapped to a sphere using a barycentric method. Af-

terwards, the added geometry is removed, leaving the original mesh in a less distorted state. Later on, this approach was not pursued, as it sometimes yielded unexpected and inconsistent results, particularly with open valve models. However, it could serve as a basis for further inquiries in this direction.

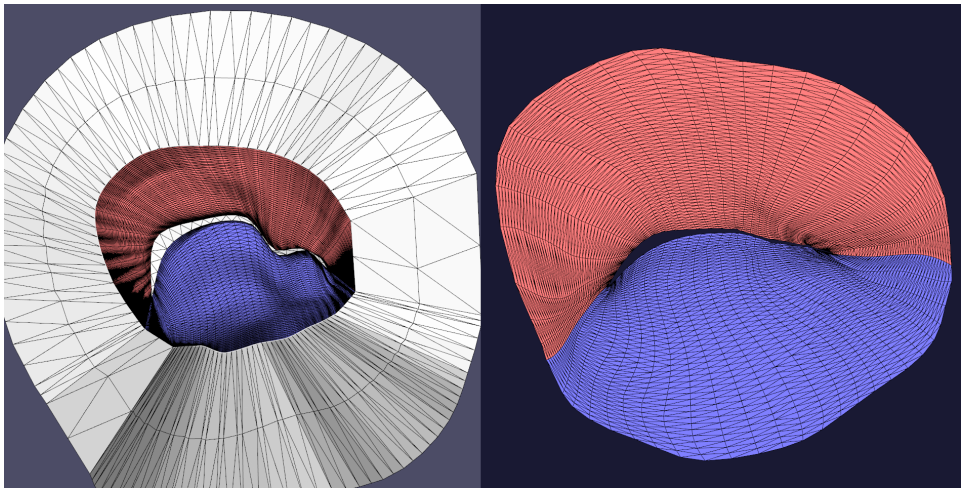

Figure 32: A different approach of constructing a 2D-view, through circular barycentric mapping with geometry extensions.

The evaluation, as described in Chapter 7, should be repeated with a larger number of participants, if more significant results are desired. It would be interesting to see if the outcomes remain consistent after participants become more acquainted with using the 2D-view. It is entirely possible speed and accuracy will increase once users are more accommodated with the proposed valve visualization. The evaluation could also be enhanced through additional tasks related to therapeutic decision making or pre- and post-surgical intervention comparison, as these are the probable use cases most often stated.

References

[1] A. Aggarwal, V. S. Aguilar, C.-H. Lee, G. Ferrari, J. H. Gorman, R. C. Gorman, and M. S. Sacks. Patient-Specific Modeling of Heart Valves: From Image to Simulation. *Functional Imaging and Modeling of the Heart*, pages 141–149, 2013.

[2] J. H. Ahlberg, E. N. Nilson, and J. L. Walsh. *The theory of splines and their applications*. Academic Press, 1967.

[3] A. V. Bartroli, R. Wegenkittl, A. Konig, and E. Groller. Nonlinear virtual colon unfolding. pages 411–579, 2001.

[4] W. Böcker, H. Denk, P. U. Heitz, G. Höfler, H. Kreipe, and H. Moch. *Pathologie*. Elsevier, 2012.

[5] C. Bennis, J. M. Vézien, and G. Iglésias. Piecewise Surface Flattening for non-distorted Texture Mapping. *ACM SIGGRAPH Computer Graphics*, 25(4):237–246, 1991.

[6] A. I. Bobenko and T. Hoffmann. Conformally Symmetric Circle Packings: A Generalization of Doyle's Spirals. *Experimental Mathematics*, 10(1):141–150, 2001.

[7] A. I. Bobenko and T. Hoffmann. Minimal surfaces from circle patterns: Geometry from combinatorics. *Annals of Mathematics*, 164:231–264, 2001.

[8] R. O. Bonow, B. A. Carabello, K. Chatterjee, A. C. de Leon, D. P. Faxon, M. D. Freed, W. H. Gaasch, B. W. Lytle, R. A. Nishimura, P. T. O'Gara, R. A. O'Rourke, C. M. Otto, P. M. Shah, and J. S. Shanewise. ACC/AHA 2006 Guidelines for the Management of Patients With Valvular Heart Disease. *Circulation*, 114(5):e84–e231, 2006.

[9] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy. *Polygon Mesh Processing*. CRC Press Taylor and Francis Group, 2010.

[10] P. Burlina, C. Sprouse, D. DeMenthon, A. Jorstad, R. Juang, F. Contijoch, T. Abraham, D. Yuh, and E. McVeigh. Patient-Specific Modeling and Analysis of the Mitral Valve Using 3D-TEE. *Information Processing in Computer-Assisted Interventions*, pages 135–146, 2010.

[11] A. Carpentier, D. H. Adams, and F. Filsoufi. *Carpentier's Reconstructive Valve Surgery*, volume Innere Organe. Saunders/Elsevier, 1 edition, 2010.

[12] C. Clouchoux, O. Coulon, D. Rivière, A. Cachia, J. F. Mangin, and J. Régis. Anatomically Constrained Surface Parameterization for Cortical Localization. *Med. Image Comput. Comput. Assist. Interv.*, 8(2):344–351, 2005.

[13] C. A. Conti, M. Stevanella, F. Maffessanti, S. Trunfio, E. Votta, A. Roghi, O. Parodi, E. G. Caiani, and A. Redaelli. Mitral valve modelling in ischemic patients: Finite element analysis from cardiac magnetic resonance imaging. pages 1059–1062, 2010.

[14] W. Daniel, H. Baumgartner, C. Gohlke-Bärwolf, P. Hanrath, D. Horstkotte, K. C. Koch, A. Mügge, H. J. Schäfers, and F. Flachskampf. Klappenvitien im Erwachsenenalter. *Clin Res Cardiol*, 95:620–641, 2006.

[15] P. Degener, J. Meseth, and R. Klein. An Adaptable Surface Parameterization Method. *Proceedings of the 12th International Meshing Roundtable*, pages 201–213, 2003.

[16] M. Eck, T. D. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *Proceedings of SIGGRAPH '95*, pages 173–182, 1995.

[17] S. Engelhardt, N. Lichtenberg, S. Al-Maisary, R. De Simone, H. Rauch, J. Roggenbach, S. Müller, M. Karck, H.-P. Meinzer, and I. Wolf. Towards Automatic Assessment of the Mitral Valve Coaptation Zone from 4D Ultrasound. *Functional Imaging and Modeling of the Heart*, pages 137–145, 2015.

[18] B. Fischl, M. I. Sereno, and D. A. M. Cortical surface-based analysis. II: Inflation, flattening, and a surface-based coordinate system. *Neuroimage*, 9(2):195–207, 1999.

[19] M. S. Floater. Parameterization and Smooth Approximation of Surface Triangulations. *Computer Aided Geometric Design*, 14(3):231–250, 1997.

[20] M. S. Floater. Mean Value Coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003.

[21] C. F. Gauß. Disquisitiones generales circa super cies curvas. *Commentationes Societatis Regiæ Scientiarum Gottingensis Recentiores*, 6:99–146, 1827.

[22] S. J. Gortler, C. Gotsman, and T. D. Discrete One-Forms in Meshes and Applications to 3D Mesh Parameterization. *Computer Aided Geometric Design*, 23(2):83–112, 2006.

[23] S. Grbic, R. Ionasec, D. Vitanovski, I. Voigt, Y. Wang, B. Georgescu, N. Navab, and D. Comaniciu. Complete valvular heart apparatus model from 4D cardiac CT. *Medical Image Anlaysis*, 16(5):1003–1014, 2012.

[24] K. C. Gurijala, R. Shi, W. Zeng, X. Gu, and A. Kaufman. Colon Flattening Using Heat Diffusion Riemannian Metric. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2848–2857, 2013.

[25] P. E. Hammer, D. P. Perrin, P. J. del Nido, and R. D. Howe. Image-based mass-spring model of mitral valve closure for surgical planning. *Proc. SPIE*, 6918, 2008.

[26] G. Herold. *Innere Medizin*. Gerd Herold, 2018.

[27] W. Hong, F. Qiu, and A. Kaufman. A pipeline for computer aided polyp detection. *IEEE Trans. Vis. Comput. Graph.*, 12(5):861–869, 2006.

[28] K. Hormann and G. Greiner. MIPS: An Efficient Global Parameterization Method. *Curve and Surface Design: Saint-Malo 1999*, pages 153–162, 2000.

[29] K. Hormann, K. Polthier, and A. Sheffer. Mesh Parameterization: Theory and Practice. *SIGGRAPH Asia 2008 Course Notes*, 2008.

[30] Z. Hu, D. Metaxas, and L. Axel. Computational Modeling and Simulation of Heart Ventricular Mechanics from Tagged MRI. *Functional Imaging and Modeling of the Heart*, pages 369–383, 2005.

[31] R. I. Ionasec, I. Voigt, B. Georgescu, Y. Wang, H. Houle, F. Vega-Higuera, N. Navab, and D. Comaniciu. Patient-Specific Modeling and Quantification of the Aortic and Mitral Valves from 4D Cardiac CT and TEE. *IEEE Transactions on Medical Imaging*, 29(9):1636–1651, 2010.

[32] D. Julius, V. Kraevoy, and A. Sheffer. D-Charts: Quasi-Developable Mesh Segmentation. *Computer Graphics Forum*, 24(3):581–590.

[33] Kitware, Inc. Online at: `https://www.vtk.org/`, accessed July 2018.

[34] G. Kós and T. Várady. Parameterizing complex triangular meshes. *Curve and surface design: Saint-Malo 2002*, pages 265–274, 2003.

[35] M. Krone, F. Frieß, K. Scharnowski, G. Reina, S. Fademrecht, T. Kulschewski, J. Pleiss, and T. Ertl. Molecular Surface Maps. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):701–710, 2017.

[36] E. T. Y. Lee. Choosing nodes in parametric curve interpolation. *Computer-Aided Design*, 21(6):363–370, 1989.

[37] Y. Lee, H. S. Kim, and S. Lee. Mesh parameterization with a virtual boundary. *Computers and Graphics*, 26(5):677–686, 2002.

[38] B. Lévy. Dual Domain Extrapolation. *ACM Trans. Graph.*, 22(3):364–369, 2003.

[39] N. Lichtenberg. Semi-Automatic Segmentation of the Mitral Valve Leaflets on 4D Ultrasound Images. Master's thesis, University of Koblenz-Landau, 2015.

[40] B. Lévy and J.-L. Mallet. Non-Distorted Texture Mapping For Sheared Triangulated Meshes. *Proceedings of SIGGRAPH '98*, pages 343–352, 1998.

[41] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics*, 21(3):362–371, 2002.

[42] J. Maillot, H. Yahia, and A. Verroust. Interactive Texture Mapping. *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, pages 27–34, 1993.

[43] J. Marino and A. Kaufman. Planar Visualization of Treelike Structures. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):906–915, 2016.

[44] J. Marino, W. Zeng, X. Gu, and A. Kaufman. Context preserving maps of tubular structures. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1997–2004, 2011.

[45] D. Mathieu, M. Mark, and A. Pierre. Intrinsic Parameterizations of Surface Meshes. *Computer Graphics Forum*, 21(3):209–218, 2002.

[46] S. Nadeem, J. Marino, X. Gu, and A. Kaufmann. Corresponding Supine and Prone Colon Visualization Using Eigenfunction Analysis and Fold Modeling. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):751–760, 2017.

[47] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.

[48] N. Ray, W. C. Li, B. Lévy, A. Sheffer, and P. Alliez. Periodic Global Parameterization. *ACM Trans. Graph.*, 25(4):1460–1485, 2006.

[49] P. V. Sander, S. J. Gortler, J. Snyder, and H. Hoppe. Signal-Specialized Parametrization. *Proceedings of the 13th Eurographics Workshop on Rendering (EGWR '02)*, pages 87–98, 2002.

[50] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture Mapping Progressive Meshes. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 409–416, 2001.

[51] R. J. Schneider, D. P. Perrin, N. V. Vasilyev, G. R. Marx, P. J. del Nido, and R. D. Howe. Mitral annulus segmentation from 3D ultrasound using graph cuts. *IEEE Transactions on Medical Imaging*, 29(9):1676–1687, 2010.

[52] M. Schünke, E. Schulte, U. Schumacher, M. Voll, and K. Wesker. *Prometheus: LernAtlas der Anatomie*, volume Innere Organe. Thieme, Stuttgart, 4 edition, 2015.

[53] V. Schomaker, J. Waser, R. E. Marsh, and G. Bergman. To fit a plane or a line to a set of points by least squares. *Acta Crystallographica*, 12(8):600–604, 1957.

[54] A. Sheffer and E. de Sturler. Parameterization of Faceted Surfaces for Meshing using Angle-Based Flattening. *Engineering with Computers (2001)*, 17:326–337, 2001.

[55] A. Sheffer and J. C. Hart. Seamster: Inconspicuous Low-Distortion Texture Seam Layout. *Proceedings of IEEE Visualization 2002*, pages 291–298, 2002.

[56] A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov. ABF++ : Fast and Robust Angle Based Flattening. *ACM Transactions on Graphics*, 24(2):311–330, 2005.

[57] Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun. Designing Quadrangulations with Discrete Harmonic Forms. *Proceedings of the 4th Eurographics Symposium on Geometry Processing (SGP 2006)*, pages 201–210, 2006.

[58] R. Toro and Y. Burnod. Geometric atlas: modeling the cortex as an organized surface. *Neuroimage*, 20(3):1468–1484, 2003.

[59] Z. G. Turi. Mitral Valve Disease. *Circulation*, 109:e38–e41, 2004.

[60] W. T. Tutte. Convex Representations of Graphs. *Proceedings of the London Mathematical Society*, 10:304–320, 1960.

[61] S. Vera, M. A. González Ballester, and D. Gil. Anatomical Parameterization for Volumetric Meshing of the Liver. *Proc. SPIE*, 9036, 2014.

[62] E. L. Wachspress. *A Rational Finite Element Basis*. Academic Press, New York, 1975.

[63] Y. Wang, X. Gu, K. M. Hayashi, T. F. Chan, P. M. Thompson, and S.-T. Yau. Brain Surface Parameterization Using Riemann Surface Structure. *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, 2:1061–1066, 2005.

[64] Wikimedia Commons. Online at: `https://commons.wikimedia.org/wiki/File:Mitralinsuff_TEE.jpg`, accessed July 2018.

[65] Wikimedia Commons. Online at: `https://de.wikipedia.org/wiki/Herzklappe#/media/File:Blausen_0469_HeartValves.png`, accessed May 2018.

[66] Wikimedia Commons. Online at: `https://en.wikipedia.org/wiki/File:Latidos.gif`, accessed May 2018.

[67] Wikimedia Commons. Online at: `https://es.wikipedia.org/wiki/Archivo:RPost_op_view_mitral_valve_repair_Dr_S%C3%A1ez_de_Ibarra.jpg`, accessed May 2018.

[68] R. Zayer, B. Lévy, and H.-P. Seidel. Linear Angle Based Parameterization. *Proceedings of the 5th Eurographics Symposium on Geometry Processing (SGP 2007)*, pages 135–141, 2007.

[69] W. Zeng, J. Marino, K. C. Gurijala, X. Gu, and A. Kaufman. Supine and Prone Colon Registration Using Quasi-Conformal Mapping. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1348–1357, 2010.

[70] L. Zhu and T. Jiang. Parameterization of 3D Brain Structures for Statistical Shape Analysis. *Proc. SPIE*, 5370, 2004.