Efficient Position-Based Deformable Colon Modeling for Endoscopic Procedures Simulation

Marcelo Martins
Lucas Morais
Luciana Nedel
marcelo@ufrgs.br
lzmorais@inf.ufrgs.br
nedel@inf.ufrgs.br
Federal Univ. of Rio Grande do Sul
Porto Alegre, Brazil

Rafael Torchelsen
rafael.torchelsen@inf.ufpel.edu.br
Federal University of Pelotas
Pelotas, Rio Grande do Sul, Brazil

Anderson Maciel
anderson.maciel@tecnico.ulisboa.pt
Instituto Superior Técnico,
Universidade de Lisboa
Lisbon, Portugal
Federal Univ. of Rio Grande do Sul
Porto Alegre, Brazil

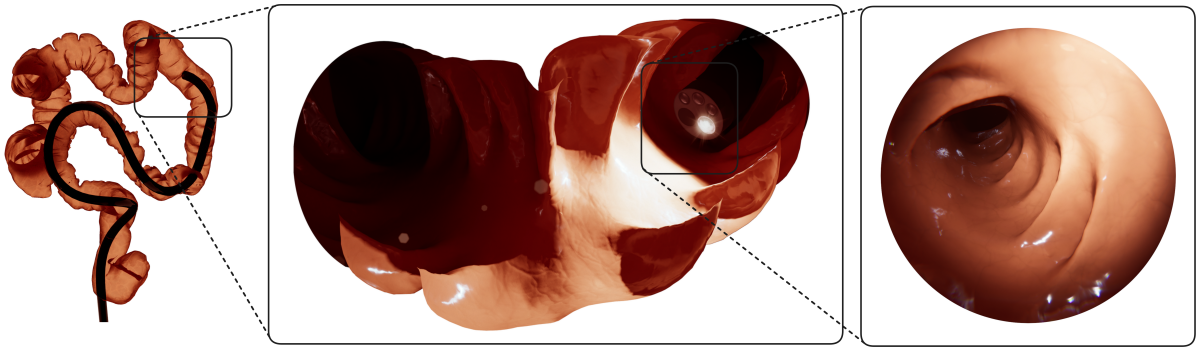

Figure 1: On the left, the colon mesh undergoes deformation in response to the endoscope movement. In the center, a kinematic perspective of the endoscope highlights light reflections on the walls of the colon. On the right is a frame of the operator's view, as seen when using our colonoscopy simulator.

ABSTRACT

Current endoscopy simulators oversimplify navigation and interaction within tubular anatomical structures to maintain interactive frame rates, neglecting the intricate dynamics of permanent contact between the organ and the medical tool. Traditional algorithms fail to represent the complexities of long, slender, deformable tools like endoscopes and hollow organs, such as the human colon, and their interaction.

In this paper, we address longstanding challenges hindering the realism of surgery simulators, explicitly focusing on these structures. One of the main components we introduce is a new model for the overall shape of the organ, which is challenging to retain due to the complex surroundings inside the abdomen. Our approach uses eXtended Position-Based Dynamics (XPBD) with a Cosserat rod constraint combined with a mesh of tetrahedrons to retain the colon's shape. We also introduce a novel contact detection algorithm for tubular structures, allowing for real-time performance. This comprehensive representation captures global deformations and local features, significantly enhancing simulation fidelity compared to previous works.

Results showcase that navigating the endoscope through our simulated colon seemingly mirrors real-world operations. Additionally, we use real-patient data to generate the colon model, resulting in a highly realistic virtual colonoscopy simulation. Integrating efficient simulation techniques with practical medical applications arguably advances surgery simulation realism.

SIGGRAPH Conference Papers '24, July 27-August 1, 2024, Denver, CO, USA

ACM ISBN 979-8-4007-0525-0/24/07.
https://doi.org/10.1145/3641519.3657454

CCS CONCEPTS

• **Computing methodologies** → **Physical simulation**; **Collision detection**; *Virtual reality*; • **Software and its engineering** → *Virtual worlds training simulations*; • **Applied computing** → Consumer health.

KEYWORDS

Interactive simulation, Surgery simulation, Position-based dynamics, Real-time collision detection, Cosserat rods

ACM Reference Format:
Marcelo Martins, Lucas Morais, Luciana Nedel, Rafael Torchelsen, and Anderson Maciel. 2024. Efficient Position-Based Deformable Colon Modeling for Endoscopic Procedures Simulation. In *Special Interest Group on Computer*

Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24), July 27-August 1, 2024, Denver, CO, USA. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3641519.3657454

1 INTRODUCTION

Colorectal cancer (CRC) is the second leading cause of cancer-related deaths. Projections indicate that there will be almost 1.1 million deaths attributable to CRC by 2030 [Siegel et al. 2020]. Early detection of polyps, adenomas, and CRCs significantly prevents and lowers mortality rates. Colonoscopy is the gold standard technique for detecting and eliminating these polyps [Dik et al. 2014], leading to a 67% decrease in the risk of death [Doubeni et al. 2016].

However, mastering colonoscopy requires extensive training, and Virtual Reality (VR) simulators have proven to enhance performance in Patient-based Assessments [Koch et al. 2012]. These simulators are crucial for training novice surgeons and improving patients' comfort and safety. Additionally, VR simulators offer the advantage of accelerating the training process for students in the early stages of practical learning by enabling the repeated practice of procedures [Van der Wiel et al. 2016]. The effectiveness of virtual endoscopic simulators frequently rivals or surpasses that of traditional training methods [Zhang et al. 2021].

The ability of a simulator to accurately mimic the deformation of soft tissues is a crucial factor in its realism. Still, the simulation of biological tissues presents numerous research challenges, such as delivering plausible visual behavior while keeping high performance and quick response times [Zhang et al. 2017]. The pivotal factor influencing the success of these virtual simulators lies predominantly in the mechanical model employed for simulating both the endoscope tube and the colon. While the colon can be conceptualized as a soft tissue, the endoscope is primarily a tensioned steel rod within a flexible tube.

Typically, simulation models concentrate on replicating the effects of these critical characteristics. This paper presents novel solutions for improving virtual tissue behavior during a tubular-based surgery simulation. Our contribution is three-fold:

- A technique to simulate the computationally demanding deformation of the colon shape resulting from its interaction with the endoscope.
- A collision detection algorithm designed for dynamic and intricate tubular surfaces. Our approach ensures real-time performance and is finely tuned to navigate intense collision situations efficiently.
- A streamlined workflow to facilitate patient data integration, eliminating the need for manual colon modeling. This enhances efficiency and opens up novel possibilities for medical training.

2 RELATED WORK

Preceding related works primarily emphasized simulating the endoscope, often oversimplifying the representation of the colon. The endoscope's relative *ease* of representation is understandable, given its nature as a controllable steel rod **within** soft tissues. Consequently, most collision reactions and shape deformations occur in the soft tissue. Based on these observations, we directed our focus toward modeling the colon.

2.1 Endoscope Model

While our contributions are centered on the colon, and we limit the scope of the details about the endoscope, we still need to understand how the endoscope modeling approach affects the context of colonoscopy. Overall, the endoscope tube comprises a series of interlinked, rigid, cylindrical, and hollow segments, with a few cables crossing the middle of the segments to control the stiffness and direction of the tube [Torii 2003]. To model it, recent works [Korzeniowski et al. 2016; Morais et al. 2023] have used Cosserat rods elements [Spillmann and Teschner 2007], and direct position-based solvers [Deul et al. 2018; Macklin et al. 2016], respectively. We extend these previous works by applying the Cosserat theory and Position-Based Dynamics (PBD) [Fang et al. 2023; Müller et al. 2007] to approach the problem.

2.2 Soft-Tissue Model

Several soft tissue modeling approaches are applied in surgical simulation [Berndt et al. 2017; Chang et al. 2011; Duan et al. 2016; France et al. 2005; Pan et al. 2015; Visser et al. 2010]. However, this line of research predominantly encompasses significantly thicker tissues than the colon. The most common methods are force-based. These include the Finite Element Method (FEM) analysis and the Mass-Spring Method (MSM) lattices.

FEM approximates continuous functions as discrete models involving a finite number of points and sub-domains in the problem's domain [Tekkaya and Soyarslan 2018]. Solving partial differential equations with linear FEM is computationally efficient, but it does not handle large deformations well. Yan and Canny [1999], Nebel [2001] and [Guan et al. 2014] presented deformable tissue models for surgical simulation using nonlinear FEM. Yan et al. [2007] improved the approach with a meshing algorithm based on Delaunay's criterion. However, these methods are generally unsuitable for real-time [Ye et al. 2018]. Chentanez et al. [2009] proposed a linearization on the FEM formulation to accelerate their coupled needle-tissue simulator to less than 100 ms per frame; still not real-time.

The Mass-Spring Method (MSM) is widely used for simulating deformable objects such as cloth and soft tissue [Mollemans et al. 2003; Patete et al. 2013]. MSM is based on mass points connected by springs, where Hooke's law governs the conventional model. Models based on MSM are more accessible to implement and provide more straightforward calculations, allowing real-time interaction. Nonlinearity emerges from linear springs without additional cost. Recently, Omar et al. [2022] presented a model that integrates traditional Hooke's law with a nonlinear volume force derived from the conical spring methodology. MSM, however, lacks topological and material properties accuracy, requires complex tuning of parameters, and is often unstable in terms of structure and numerical computation [Yan et al. 2007].

Previous force-based methods also exploited the concept of using a skeleton core to speed up deformation computation and incorporate behaviors. Zhang et al. [2010] used a skeleton to preserve volume when bending. Other authors used skeletons with soft layers to control the animation of humanoid characters [Capell et al. 2002; Kim and Pollard 2011]. When the static result of several forces is more important than the dynamics of deformations, a grid

schema can also be applied [Mitchell et al. 2015] to attain real-time responses.

Position Based Dynamics (PBD) [Müller et al. 2007] has appeared as a scalable alternative that offers visually appealing results in games and interactive real-time applications. Unlike force-based methods, Position-Based Dynamics (PBD) operates by iteratively solving position constraints to achieve the desired behavior, and its versatility has led to widespread use [Fang et al. 2023].

The eXtended Position Based Dynamics (XPBD) [Macklin et al. 2016] enhanced the PBD method by applying an implicit time integration, increasing its stability and solving the iteration count and time step-dependent constraint stiffness. Additionally, XPBD can provide estimates of constraint forces, making it suitable for applications requiring haptic user feedback and even allowing unified simulation with fluids and rigid bodies [Müller et al. 2020]. Our approach aligns with this trend, using XPBD as the framework for simulating the colon and the endoscope. However, we incorporate several components, as explained later. One is blending a Cosserat rod and tetrahedra constraints with a spline; another is a specialized collision detection algorithm.

2.3 Collision Detection

In addition to soft tissue deformation, collision detection **between deformable structures** has long been a primary bottleneck in interactive simulation [Teschner et al. 2005; Wang and Cao 2021]. Typically, accuracy conflicts with achieving the high framerates demanded for interaction. When force feedback is needed, the performance requirements become even more stringent, often leading to constraints on one side that compromise results on the other.

When working with deformable meshes, we face the challenge of constantly updating pre-computed collision structures to represent the object's geometry accurately in its current state. Numerous algorithms have been proposed to tackle this challenge. Those proposed by Choi et al. [2017], Liang et al. [2018], and Kim et al. [2019] have employed bounding volume hierarchies (BVHs) to improve collision detection efficiency. Other techniques used Spatial Partitioning, such as hash tables [Maciel et al. 2007; Maule et al. 2010; Ye et al. 2016], spatiotemporal coherence [Maciel and De 2008] and K-D tree [Schauer and Nüchter 2015; Serpa and Rodrigues 2019], along with distance fields [Liu and Kim 2013].

Concerning collision detection for colonoscopy simulation, Yi et al. [2006] used a uniform cubic subdivision of space with adaptations for continuous collision detection. France et al. [2005] proposed a different approach, employing a chain of spheres in the broad phase for both the endoscope and the colon meshes. Korzeniowski et al. [2016] used an AABB BVH between the colon and the endoscope. Other recent works rely on collision detection methods in commercial software such as Nvidia PhysX within the Unity game engine [Morais et al. 2023].

Almost all the works mentioned used a colon mesh that is either rigid or has minimal freedom of movement, a scenario where collisions occur between two rigid or deformable and one rigid object. Methods exist that are competent to handle them but do not tolerate a training simulation of a real-life situation where both main components are deformable. Furthermore, we have no knowledge of collision detection techniques that have leveraged the unique characteristics of the tubular structures in our context to improve performance. Our approach, described in Sec. 3.2, addresses both aspects.

3 METHODS

To improve the living tissues' dynamic behavior during the simulation of colonoscopies, we propose a pipeline (see Fig. 2) and outline its critical stages, starting with acquiring the colon mesh from patient data. The mesh is reconstructed from computed tomography (CT) images, usually obtained from patients undergoing colonography. Automatic segmentation is possible due to colon insufflation and the dye contrast used in the preparation for the scan. The colon centerline is then extracted and used to configure a spline skeleton for the deformable colon (details are in Sec. 3.1). Our endoscope model is also based on a spline skeleton. Having both structures modeled on the same basis allows for a targeted collision detection approach. This new collision method is detailed in Sec. 3.2. Finally, to process the deformation of these elements, we propose an XPBD formulation in Sec. 3.3. We delineate each pipeline step below.

Due to page limits, we have omitted details on the user interface and the endoscope modeling. Finer points about the software architecture and the choices made for the graphics output can be found as supplemental material.

3.1 Patient-Based Colon Modeling

Our processing pipeline begins with obtaining a triangle mesh of the colon surface. This can be a reproduction made by a designer anatomist or a reconstruction from medical imaging. While other medical imaging techniques could also provide anatomical data [Hong et al. 2014], we consider colonography CT data ideal. We followed an approach using open tools [Paulo et al. 2018] capitalizing on The Cancer Imaging Archive[1].

Given the surface mesh, we enhance the model with structures necessary to deform the colon. The next step is to extract a set of points close to the centerline of the colon from the rectum to the cecum. There are several methods to estimate the centerline, including employing mesh shrinking techniques [Paulo et al. 2018] or a pathfinding algorithm. We opted to discretize the mesh into voxels and used them as input for the A* pathfinding algorithm. This is not crucial to the pipeline; applying mesh shrinking will yield similar results. We sample points regularly spaced from the centerline and interpolate them using a centripetal Catmull-Rom spline [Catmull and Rom 1974] that will serve as a backbone for the colon shape. The density of the control point sampling must be high enough to preserve the relationship of the spline with the high-frequency features.

The spline provides a continuous path throughout the colon but lacks volume. Conversely, the mesh provides volume but poses challenges in detecting multiple collisions due to its complexity. We thus propose to enrich the spline with volumetric information, creating a structure we call *volumetric spline*, or simply *volspline*. The volumetric spline is illustrated in Fig. 3. It lies at the foundation of our collision detection and simulation model presented in Sec. 3.2 and 3.3 below. The *volspline* is a data structure composed of a

[1] https://www.cancerimagingarchive.net

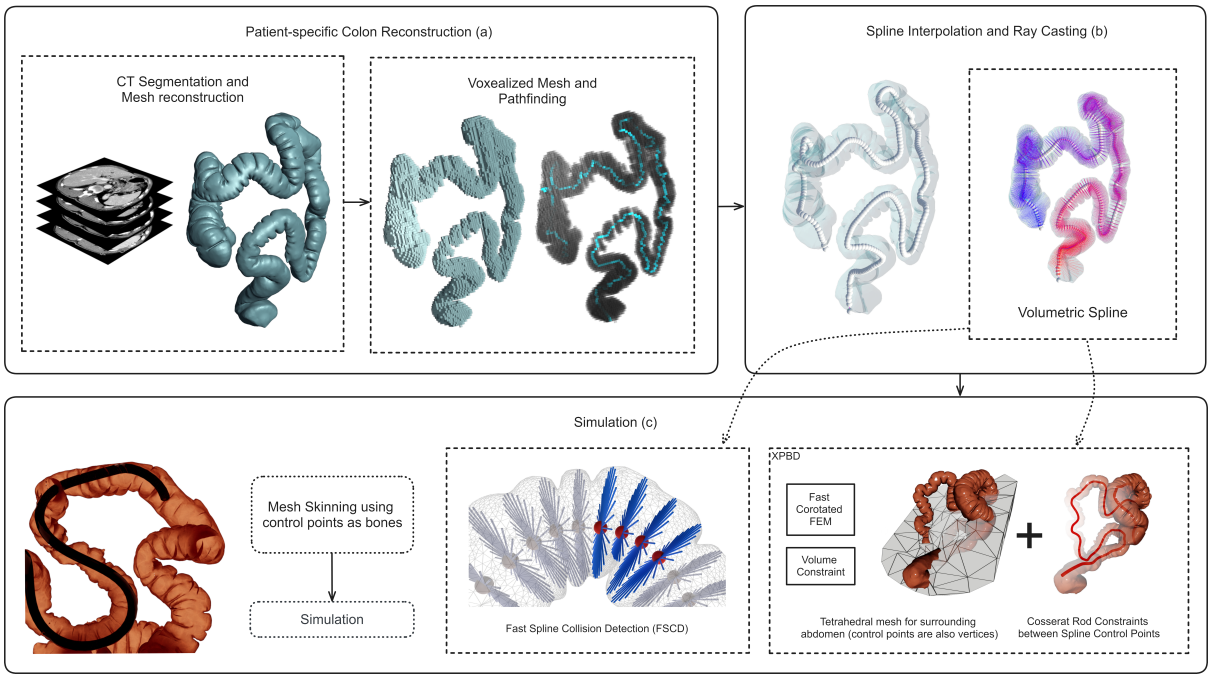

Figure 2: The simulation pipeline is divided into three main parts: a) acquisition and reconstruction of the colon mesh from a real patient CT, including the colon centerline estimation; b) building a volumetric spline skeleton from the reconstructed mesh and centerline; c) simulate the colon deformation using an XPBD formulation and a tailored collision detection method.

set of Radiating Control Points (RCPs) that serve a dual purpose: they act as control points and encompass additional topological information about the mesh. For each RCP $\mathbf{P}_i$ in the set $\mathbf{P}$, rays are cast from the point toward the surrounding mesh. We refer to the intersection points between the rays and the mesh as radial points $\mathbf{V}_j$. Each RCP $\mathbf{P}_i$ is associated with m radial points. These radial points collectively form an array $\mathbf{V} = [\mathbf{V}_0, \mathbf{V}_1, \cdots, \mathbf{V}_{m-1}]$. The blue rays in Fig. 3 (bottom row) represent them.

We also modeled the endoscope as a *volspline*, with a set of RCPs $\mathbf{U}_k$, as illustrated in green in Fig. 3 (top row). An endoscope is more straightforward to model than the colon since its volume can be defined by a radius parameter $\mathbf{r}$ (purple line in Fig. 3). Furthermore, it does not need reconstruction, as its generalized cylindrical shape can be procedurally modeled.

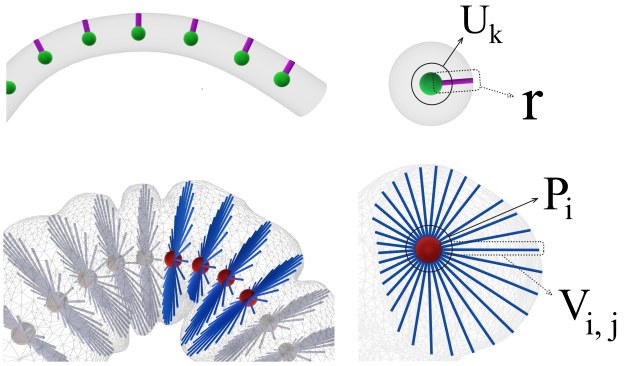

Figure 3: The volumetric spline representation for the endoscope (top) and colon (bottom), in front view (left), and cross-sections (right).

3.2 Fast Spline Collision Detection (FSCD)

The FSCD algorithm is tailored to detect multiple intersections between two deformable tubular structures that permanently touch while moving inside each other, such as an endoscope inside the colon. The method identifies collisions by comparing two volumetric splines, as in Fig. 4. We thus have an inner and an outer *volspline*.

The process of computing the colliding points between the two *volsplines* starts by sampling the two curves with any number of $\mathbf{s}$ steps. By interpolating s steps between $\mathbf{U}_k$ and $\mathbf{U}_{k+1}$, and $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$, we can create as many RCPs as needed to refine the intersection test, allowing us to adjust the precision in regions where more accuracy is sought. An intersection test is triggered for each sample pair in the set of $\mathbf{s}^2$ pairs (see optimizations in Sec. 3.2.1). The number of steps defines the resolution of the approach and may affect the accuracy. The intersection test returns a set of colliding tuples. Each tuple contains the colliding point in each *volspline* and the respective spline segment IDs, where a segment is a rigid link associated with a control point upon which the collision response will be applied. This means the detection is processed at a higher resolution than the control points, but the response is given at the control points.

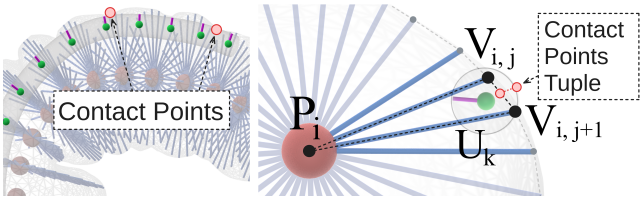

Figure 4: A front view and a cross-section close-up of a collision between two *volsplines* as detected by the FSCD method.

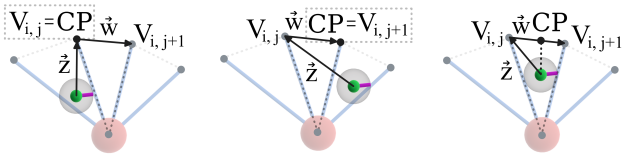

Figure 5: The images portray the three scenarios expressed in Eq. 1 in the same orded, where $\vec{w}$ is the vector from $V_{i,j}$ to $V_{i,j+1}$, and $\vec{z}$ is the vector between U_k and $V_{i,j}$.

As the inner *volspline* has a uniform radius all along, we can see each RCP $\mathbf{U}_k$ as a sphere (Fig. 4-right). We cannot make that assumption for the outer *volspline*, as $\mathbf{P}_i$'s have variable radial vectors. Therefore, we build triangles grouping each radial vector with its adjacent, as illustrated by the dashed line in Fig. 4-right. This reduces the problem to the intersection check between the sphere and the side of the triangle representing the outside mesh wall.

For processing the intersection test between the $\mathbf{U}_k$ sphere (inner *volspline*) and the outer edge of the outer *volspline* triangle (see Fig. 5) we compute $\vec{\mathbf{z}} = \mathbf{V}_{i,j} - \mathbf{U}_k$ and then project $\vec{\mathbf{z}}$ on $\vec{\mathbf{w}}$ to obtain the closest point **CP** on $\vec{\mathbf{w}}$ to $\mathbf{U}_k$ using the following equations:

$$\mathbf{CP} = \begin{cases} \mathbf{V}_{i,j}, & \text{if } \vec{\mathbf{w}} \cdot \vec{\mathbf{z}} \leq 0 \\ \mathbf{V}_{i,j+1}, & \text{if } \vec{\mathbf{w}} \cdot \vec{\mathbf{w}} < \vec{\mathbf{w}} \cdot \vec{\mathbf{z}} \\ \mathbf{V}_{i,j} + q \cdot \vec{\mathbf{w}} & \text{otherwise.} \end{cases} \tag{1}$$

At the last condition, the closest point comes over $\vec{\mathbf{w}}$, where **CP** is obtained as a linear interpolation between $\mathbf{V}_{i,j}$ and $\mathbf{V}_{i,j+1}$ with q defined by Eq. 2:

$$q = \frac{\vec{\mathbf{w}} \cdot \vec{\mathbf{z}}}{\vec{\mathbf{w}} \cdot \vec{\mathbf{w}}} \quad \text{as } q \in (0, 1). \tag{2}$$

Finally, we use Eq. 3 to check if a collision occurs:

$$\|\mathbf{CP} - \mathbf{U}_k\| \leq r. \tag{3}$$

This is repeated for all $\mathbf{m}$ triangles ($\mathbf{P}_i$, $\mathbf{V}_{i,j}$, $\mathbf{V}_{i,j+1}$) of each RCP. When an intersection is found, we create a tuple with **CP** and $\|\mathbf{CP} - \mathbf{U}_k\| \cdot r + \mathbf{U}_k$, representing the exact collision location of both the inner and outer *volsplines*. We also include the IDs i and k of the respective colliding RCPs.

3.2.1 Optimizations. The core algorithm's complexity increases at a quadratic rate of the spline sampling size and number of control points. Even if the product of these numbers is not expected to be higher than around a few thousand for a human colon, we added a broad phase [Hubbard 1993] on the top of the FSCD to filter out pairs $[i, k]$ that are unlikely to collide.

Our broad phase uses a pair of AABB trees, one for each spline. Although AABBs may not produce high-precision bounding volumes, they fit the problem well and are very efficient to build [Bergen 1997]. The AABB-tree leaves correspond to a pair of RCPs of the *volspline*. We construct our trees bottom-up, creating a parent node encompassing the leaves and processing iteratively until we have a single root box. Next, we use a depth-first search between the two trees for candidate verification. Only the pairs of leaf boxes that intersect pass to the narrow phase, where the respective segments are sampled on the curve.

3.3 Colon Deformation Model

An interactive simulation of a colon for a colonoscopy simulation demands a physically plausible response and real-time performance despite the nonlinear complexity of the problem. We employed XPBD and a novel arrangement of constraints to simulate the colon and endoscope deformations. XPBD was chosen for its demonstrated robustness, speed, and convergence capabilities [Cetinaslan 2021; Cetinaslan and Chaves 2019; Deul et al. 2018].

3.3.1 XPBD Solver. We used a solver that employs a nonlinear variation of the Gauss-Seidel method [Deul et al. 2018], which is based on the XPBD algorithm [Macklin et al. 2016]. The solver overcame the slow converge problem for stiff deformables of XPBD by applying a modified version of Baraff's linear-time solver [Baraff 1996].

3.3.2 The colon as a Cosserat Rod. A natural approach to model the colon with XPBD is to exploit tetrahedral elements and constraints such as volume conservation. We did not proceed that way because the number of tetrahedra required for a coherent behavior is computationally demanding and impractical for real-time use [keyong Lee et al. 2022]. We sought alternatives and devised an approach combining the Cosserat rods theory with our *volspline*.

While a Cosserat rod is generally 1D, fitting it to a *volspline* yields volumetric and structural features to the rod. The approach relies on the assumption that the rod maintains coherence between the continuous structure of the colon with a simpler and more computationally affordable convergence [Chang et al. 2011]. The typical Cosserat rod constraint is a combination of the zero-stretch and elastic bending and twisting constraints that couples two rigid segments in one six-dimensional constraint [Deul et al. 2018], which is defined by:

$$C(\mathbf{X}_1, q_1, \mathbf{X}_2, q_2) = \begin{pmatrix} \mathbf{R}(q_1)\mathbf{p}_1 + \mathbf{X}_1 - \mathbf{R}(q_2)\mathbf{p}_2 - \mathbf{X}_2 \\ \frac{2}{l_i} \text{Vec}\left[\hat{q}_1 q_2 - \hat{q}_1^{rest} q_2^{rest}\right] \end{pmatrix}, \tag{4}$$

where $\mathbf{p}$ represents a point in the local coordinates of a segment, which is transformed to world coordinates by applying the rotation matrix $\mathbf{R}(q)$ and translation $\mathbf{X}$. The term l_i represents the average length of segments i and $i+1$, and the operator Vec [] extracts the imaginary part of a quaternion. In this context, q represents the rotation quaternions for each segment. These quaternions are crucial for constraining the bending and twisting between segments,

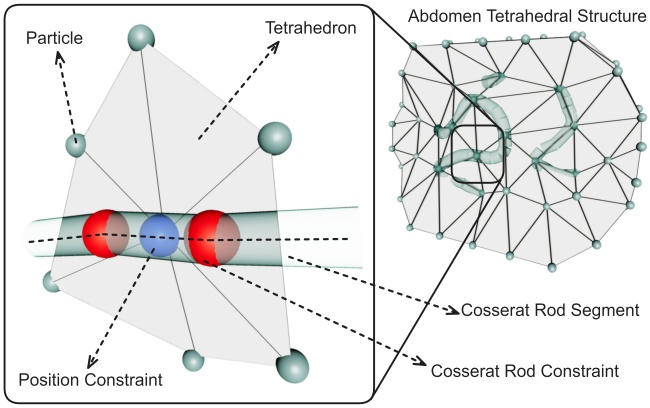

Figure 6: Geometric representation of the Cosserat rod segments (colon) held in place by tetrahedral constraints (mesentery and surrounding soft tissues).

capturing the local deformations, and enabling the calculation of strain energies associated with these deformations. In our version of the rod, we also relaxed the inextensible characteristic of the Cosserat to allow for the colon tissue's physiological stretching and shrinking behavior.

3.3.3 Keeping the colon in place. While combining the *volspline* with an elastic Cosserat rod provides structure and flexibility for a simulated colon tube, we still needed to account for the abdominal structures that keep the colon in place – the mesentery and surrounding soft tissues.

The mesentery is a fold of tissue that attaches the intestines to the posterior abdominal wall, providing support and stability. Additionally, other organs and structures in the abdomen, along with connective tissues, help maintain the colon's position. These structures work together to ensure that the colon remains in the correct location while allowing for the necessary movement and flexibility for its function during the digestive process [Byrnes et al. 2019].

Simplistic solutions, such as using distance constraints or springs, fail because they lead to inappropriate behavior due to the constraint's unidimensionality. We propose using tetrahedral constraints to model the bulk of the abdominal tissue around the colon. Contrary to an ordinary approach that would model the thin colon walls with tetrahedra, we model the surrounding medium instead.

To avoid the need for constant collision calculations between the tetrahedron structure and the Cosserat rod (colon), we constructed the tetrahedra around the Cosserat segments so that the segment is positioned in the same place as a particular component of the tetrahedron, as shown in Fig. 6. The node and the segment are linked using a zero-stretch position constraint.

To model the tetrahedra, we use the Fast Corotational constraint, along with the Volume Constraint, which models the stretching part of corotated linear elasticity [Barbic 2012]. The Fast Corotational constraint is derived from the method proposed by Kugelstadt et al. [Kugelstadt et al. 2018] and presents a tetrahedron in a tetrahedral finite element discretization of a volumetric object. Volume and corotational constraints were defined to work together. Both have their compliance determined based on Young's modulus and Poisson's ratio. The constraint equations are reproduced in the supplemental file.

3.3.4 Contact constraint. The FSCD method (Sec. 3.2) identifies the contact points where the endoscope interfaces with the colon. Following this detection, collision resolution is achieved by establishing contact constraints. The collision tuple output by the FSCD returns two segments of the *volspline*. Such a segment is treated like a rigid body from the perspective of collisions. As typical XPBD collision constraints work with particles and rigid bodies, they can be directly applied. When a collision is identified, both rigid bodies involved – the endoscope segment and the colon segment – are penalized using a contact constraint.

Different from most constraints, which are formulated mathematically as equalities $C(\mathbf{X}) = 0$, contact constraints are formulated as inequalities $C(\mathbf{X}) \geq 0$, leading to a linear complementarity problem. For more details on solving and stabilizing the contact constraint, we recommend the works from Bender et al. [2014] and Cline and Pai [2003]. For testing collisions between rigid bodies, we used:

$$C_c(\mathbf{X}) = \hat{\mathbf{n}}^T((\mathbf{X}_1 + \mathbf{R}(\boldsymbol{\vartheta}_1)\mathbf{c}_1) - (\mathbf{X}_2 + \mathbf{R}(\boldsymbol{\vartheta}_2)\mathbf{c}_2)), \geq 0, \tag{5}$$

where $\boldsymbol{\vartheta}$ represents the orientation vector of a rigid body and $\hat{\mathbf{n}}$ the collision normal. For this constraint, the obtained Jacobian is

$$\mathbf{J}_c(\mathbf{X}) = \hat{\mathbf{n}}^T[\mathbf{I} - (\mathbf{c}_{1_{world}})^\star - \mathbf{I}(\mathbf{c}_{2_{world}})^\star], \tag{6}$$

where $\mathbf{c}_i$ is the point of contact in each body and $()^*$ computes the skew-symmetric cross-product matrix of a given three-dimensional vector.

4 RESULTS

Our results are discussed in three parts: the feasibility of the colon deformation model, the performance and accuracy of collision detection, and the overall performance.

4.1 Deformation Plausibility

The simulation must replicate various scenarios encountered during colonoscopy procedures, ranging from simple to complex. Ideally, in *in vivo* training, complex scenarios should be initially avoided and then gradually incorporated. However, controlling this during exercise on a patient is challenging, being one of the aspects where a virtual simulation is preferable. There, the training can start with handcrafted colon models that capture the overall shape of the colon but deliberately avoid situations leading to more challenging conditions. Fig. 11 illustrates this and a comparison with an X-ray.

Generating colons of diverse complexities, from simple handcrafted models to complex cases derived from patient data, all utilizing the same deformation method, has never been explored before to the best of our knowledge. Typically, only simple colon models were employed, primarily due to the absence of an effective method for simulating deformation in more challenging cases. This opens up a new range of training possibilities, such as maintaining the anatomical landmarks that allow self-localization by the surgeon [Saito et al. 2020]. Figs. 10 and 12 show results from our

simulation of a colon extracted from patient data. Fig. 13 delineates the pressure distribution on the endoscope tube. The color map is created by verifying the norm of the vector from the colon contact point to the endoscope contact point before solving the contact constraint. Perforations can be simulated by adding a threshold to the norm. Finally, Fig. 14 depicts a graphics comparison with actual colonoscopy.

4.1.1 Simulation Parameters. We tried various configurations for timesteps and iterations, finding a good balance with a timestep of 0.01 seconds and 100 steps per second, allowing a 1-to-1 simulation time. The model demonstrated satisfactory convergence with 15 iterations. The endoscope was modeled with approximately 200 segments and Young's modulus of 57,000 Pa. The colon comprised 298 segments with Young's modulus of 1,000 Pa. The Young's modulus for the abdomen was set to 800 Pa, with Poisson's Ratio of 0.49 and a set of about 3.7k tetrahedra.

We used 1 RCP (Fig. 4-left) with 32 radial vectors (Fig. 4-right) for each segment of the endoscope and colon Cosserat rods in the FSCD (collision detection method) implementation. For the initial three segments of the endoscope, where the camera is located and can be locally turned by the surgeon, we employed an 8-step sampling resolution. For the remaining segments, a 4-step sampling resolution proved to be sufficient.

4.2 Performance

The tests were performed on an Intel Core i7-13700KF system, with an NVIDIA GeForce RTX 4070, and 32GB of RAM at a 4K resolution. We evaluated the performance during the endoscope insertion (Fig. 10), which was sufficient to reach the cecum region (the point of most significant stress for the application's performance due to the highest number of contact constraints).

Fig. 7 presents a graph of such measurements showing specific times for the simulation components, while Fig. 8 depicts the average performance in frames per second (FPS). The application maintained above 110 FPS throughout the simulation, with an overall average of 171 FPS. In moments of higher stress, particularly after passing through the sigmoid, traversing the transverse colon, and entering the cecum (between 150 and 250 seconds), it kept an average of 130 FPS.

We emphasize that this is a stress test generating over 1.2k contact points along the entire extent of the colon walls and the endoscope duct. We will discuss this aspect next since it represents one of our main bottlenecks.

4.3 Collision Detection Performance

As depicted in Fig. 7, our simulation demands an efficient collision detection algorithm. Past studies [Korzeniowski et al. 2016] relied on the SOLID framework [Bergen 1997]. To showcase the substantial performance gains of our solution, without compromising simulation quality, we present a scene featuring a hanging spring within another (see Figure 9). This example induces collisions throughout the springs' length, as the outer spring moves vertically and the inner spring is dragged.

The performance of both collision detection algorithms is illustrated in Table 1. To demonstrate the growth in computational cost, we tested the same scene with three mesh resolutions. At 1, 30, and 60 seconds, we measured the isolated cost of detecting all collisions for each algorithm. As the figure depicts, both have linear computational growth.

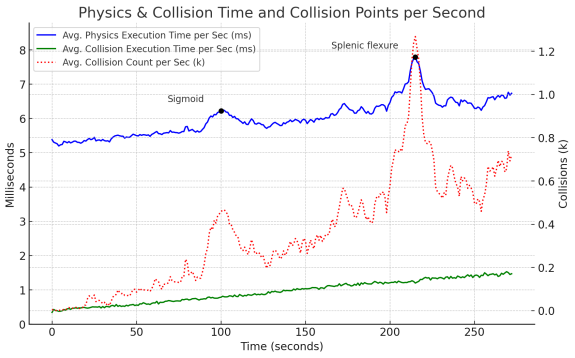

Figure 7: Average physics execution time (blue) and collision execution time (green) per second. The red line indicates the number of collisions detected in each second.

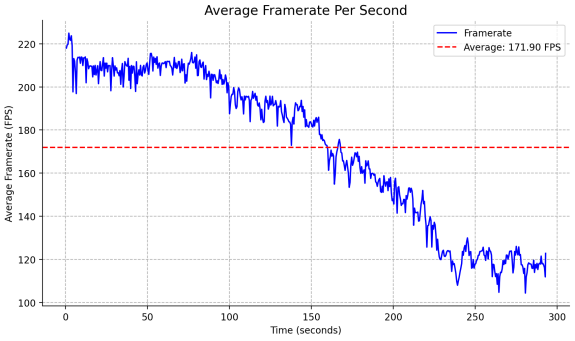

Figure 8: Framerate (blue) and average framerate (red) per second during an insertion procedure (see Fig. 10). Note that after passing the splenic flexure (around the 220th second) – a critical point –, the framerate tends to stabilize, oscillating between 100 and 105 FPS.

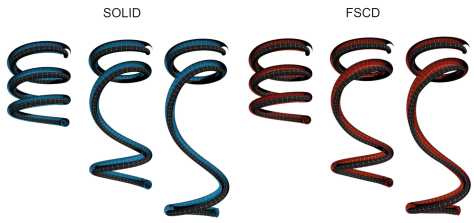

Figure 9: A visual comparison was conducted between simulations utilizing our colon deformation scheme with the FSCD and the SOLID collision detection methods. Captures at 1s, 10s, and 60s durations reveal that both results are equivalent.

Table 1: Comparison of the isolated collision detection performance for the simulation in Figure 9. The first column shows the triangle resolution in each model (inner and outer models).

	FSCD			SOLID		
	1s	**30s**	**60s**	**1s**	**30s**	**60s**
6k	0.37ms	0.42ms	0.44ms	0.47ms	0.98ms	1.20ms
19k	1.06ms	1.16ms	1.21ms	0.85ms	3.49ms	3.58ms
32k	1.77ms	2.03ms	2.02ms	1.36ms	5.67ms	5.65ms

In our comparison, we increased the number of SCPs in proportion to the count of triangles of the object to showcase an optimized situation for the SOLID algorithm and a suboptimal one for ours. The computational cost of the SOLID is directly linked to the number of triangles, and by doing so, we intended to emphasize its performance in an advantageous scenario. However, we had the option to maintain the same number of control points, considering the sampling capacity of the FSCD, which is specifically designed to handle this type of scenario. This, in turn, would result in similar performance between the low- and high-resolution tests for our approach.

5 CONCLUSIONS AND LIMITATIONS

Our exploration of colon and endoscope simulation has yielded an innovative deformation method, representing a leap forward in visual fidelity and stability compared to preceding works. This culminated in the development of a functional interactive application. This platform enables us to enhance our algorithms' output by gaining professional insights. Additionally, our work allows for creating patient-specific scenarios for simulating colonoscopy while preserving physiological landmarks crucial for navigating the endoscope within the colon environment.

While our method was designed to preserve the colon's shape, achieving this required simplification due to the intricate nature of the organic mechanisms at play. To strengthen and validate our results, a comprehensive comparative study involving actual deformation is essential, especially in the context of medical training. Undertaking such an endeavor is complex, and we envision exploring this avenue in our future works. Moreover, a crucial facet of validation pertains to assessing the educational impact of our approach. Remote education is a transformative prospect, particularly in regions where educational centers are sparse and economic disparities hinder access.

In addition to the highlighted contributions of our specific case study, we emphasize the versatility of our deformation model, particularly the adaptable collision detection component. This adaptability extends its utility to diverse scenarios involving the simulation of tubular and deformable structures, encompassing upper endoscopy, bronchoscopy, and angioscopy procedures. Furthermore, its versatility extends beyond the medical field, making it suitable for simulating scenarios in plumbing systems, subsea ducts, and other situations where the insertion of a probe into either a flexible or rigid tube and training is needed.

ACKNOWLEDGMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, Fundação de Amparo a Pesquisa do Estado do Rio Grande do Sul (FAPERGS) through grant 21/2551-0002087-3, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) through grants 311251/2020-0 and 311259/2022-7 and by Portuguese national funds through FCT, *Fundação para a Ciência e a Tecnologia*, under grant 2022.09212.PTDC (XAVIER Project), and project UIDB/50021/2020 (DOI:10.54499/UIDB/50021/2020) under the auspices of the UNESCO Chair on AI&VR of the University of Lisbon.

REFERENCES

David Baraff. 1996. Linear-time dynamics using Lagrange multipliers. In *Proc. of the 23rd annual conference on Computer graphics and interactive techniques*. 137–146.

Jernej Barbic. 2012. SIGGRAPH 2012 Course Notes FEM Simulation of 3D Deformable Solids: A practitioner's guide to theory, discretization and model reduction. Part 2: Model Reduction (version: August 4, 2012). (2012).

Jan Bender, Kenny Erleben, and Jeff Trinkle. 2014. Interactive simulation of rigid body dynamics in computer graphics. In *Computer Graphics Forum*, Vol. 33. 246–270.

Gino van den Bergen. 1997. Efficient collision detection of complex deformable models using AABB trees. *Journal of graphics tools* 2, 4 (1997), 1–13.

Iago Berndt, Rafael Torchelsen, and Anderson Maciel. 2017. Efficient Surgical Cutting with Position-Based Dynamics. *IEEE Computer Graphics and Applications* 37 (5 2017), 24–31. Issue 3. https://doi.org/10.1109/MCG.2017.45

K. Byrnes, D. Walsh, P. Dockery, K. McDermott, and J. Coffey. 2019. Anatomy of the mesentery: Current understanding and mechanisms of attachment. *Seminars in cell and developmental biology* (2019). https://doi.org/10.1016/j.semcdb.2018.10.004

Steve Capell, Seth Green, Brian Curless, Tom Duchamp, and Zoran Popović. 2002. Interactive skeleton-driven dynamic deformations. *ACM Trans. Graph.* 21, 3 (jul 2002), 586–593. https://doi.org/10.1145/566654.566622

Edwin Catmull and Raphael Rom. 1974. A class of local interpolating splines. In *Computer aided geometric design*. Elsevier, 317–326.

O. Cetinaslan. 2021. ESPEFs: Exponential Spring Potential Energy Functions for Simulating Deformable Objects. *Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games* (2021). https://doi.org/10.1145/3487983.3488303

O. Cetinaslan and Rafael O. Chaves. 2019. Energy Embedded Gauss-Seidel Iteration for Soft Body Simulations. *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (2019), 147–154. https://doi.org/10.1109/SIBGRAPI.2019.00028

Jian Chang, Xiaosong Yang, Jun J. Pan, Wenxi Li, and Jian J. Zhang. 2011. A fast hybrid computation model for rectum deformation. *The Visual Computer* 27 (2 2011), 97–107. Issue 2. https://doi.org/10.1007/s00371-010-0533-z

Nuttapong Chentanez, Ron Alterovitz, Daniel Ritchie, Lita Cho, Kris K. Hauser, Ken Goldberg, Jonathan R. Shewchuk, and James F. O'Brien. 2009. Interactive simulation of surgical needle insertion and steering. *ACM Trans. Graph.* 28, 3, Article 88 (jul 2009), 10 pages. https://doi.org/10.1145/1531326.1531394

A Ram Choi, Sung Min Kim, and Mee Young Sung. 2017. Controlling the contact levels of details for fast and precise haptic collision detection. *Frontiers of Information Technology & Electronic Engineering* 18, 8 (2017), 1117–1130.

Michael B Cline and Dinesh K Pai. 2003. Post-stabilization for rigid body simulation with contact and constraints. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, Vol. 3. IEEE, 3744–3751.

Crispin Deul, Tassilo Kugelstadt, Marcel Weiler, and Jan Bender. 2018. Direct Position-Based Solver for Stiff Rods. *Computer Graphics Forum* 37 (9 2018), 313–324. Issue 6. https://doi.org/10.1111/cgf.13326

Vincent K Dik, Leon MG Moons, and Peter D Siersema. 2014. Endoscopic innovations to increase the adenoma detection rate during colonoscopy. *World journal of gastroenterology: WJG* 20, 9 (2014), 2200.

Chyke A Doubeni, D. Corley, V. Quinn, Christopher D. Jensen, A. Zauber, M. Goodman, Jill R Johnson, S. Mehta, T. A. Becerra, Wei K. Zhao, J. Schottinger, V. P. Doria-Rose, T. Levin, N. Weiss, and R. Fletcher. 2016. Effectiveness of screening colonoscopy in reducing the risk of death from right and left colon cancer: a large community-based study. *Gut* 67 (2016), 291–298. https://doi.org/10.1136/gutjnl-2016-312712

Yuping Duan, Weimin Huang, Huibin Chang, Wenyu Chen, Jiayin Zhou, Soo Kng Teo, Yi Su, Chee Kong Chui, and Stephen Chang. 2016. Volume Preserved Mass–Spring Model with Novel Constraints for Soft Tissue Deformation. *IEEE Journal of Biomedical and Health Informatics* 20 (2016), 268–280. Issue 1. https://doi.org/10.1109/JBHI.2014.2370059

Junheng Fang, Lihua You, Ehtzaz Chaudhry, and Jian Zhang. 2023. State-of-the-art improvements and applications of position based dynamics. *Computer Animation and Virtual Worlds* (2 2023). https://doi.org/10.1002/cav.2143

Laure France, Julien Lenoir, Alexis Angelidis, Philippe Meseure, M-P Cani, François Faure, and Christophe Chaillou. 2005. A layered model of a virtual human intestine for surgery simulation. *Medical image analysis* 9, 2 (2005), 123–132.

Qiu Guan, Xiaochen Du, Yan Shao, Lili Lin, and Shengyong Chen. 2014. Three-dimensional simulation of scalp soft tissue expansion using finite element method. *Computational and mathematical methods in medicine* 2014 (2014).

DongHo Hong, Wallapak Tavanapong, Johnny Wong, JungHwan Oh, and Piet C De Groen. 2014. 3D reconstruction of virtual colon structures from colonoscopy images. *Computerized Medical Imaging and Graphics* 38, 1 (2014), 22–33.

Philip M Hubbard. 1993. Interactive collision detection. In *Proceedings of 1993 IEEE Research Properties in Virtual Reality Symposium*. IEEE, 24–31.

Stephen Kavic and Marc Basson. 2001. Complication of endoscopy. *American journal of surgery* 181 (05 2001), 319–32. https://doi.org/10.1016/S0002-9610(01)00589-X

Do keyong Lee, Tae won Kim, Yoo-Joo Choi, and Min Hong. 2022. Volumetric Object Modeling Using Internal Shape Preserving Constraint in Unity 3D. *Intelligent Automation and Soft Computing* (2022). https://doi.org/10.32604/iasc.2022.020674

Junggon Kim and Nancy S. Pollard. 2011. Fast simulation of skeleton-driven deformable body characters. *ACM Trans. Graph.* 30, 5, Article 121 (oct 2011), 19 pages. https://doi.org/10.1145/2019627.2019640

Minsang Kim, Nak-Jun Sung, Sang-Joon Kim, Yoo-Joo Choi, and Min Hong. 2019. Parallel cloth simulation with effective collision detection for interactive AR application. *Multimedia Tools and Applications* 78, 4 (2019), 4851–4868.

Arjun D Koch, Jelle Haringsma, Erik J Schoon, Rob A De Man, and Ernst J Kuipers. 2012. Competence measurement during colonoscopy training: the use of self-assessment of performance measures. *Official journal of the American College of Gastroenterology (ACG)* 107, 7 (2012), 971–975.

Przemyslaw Korzeniowski, Alastair Barrow, Mikael H Sodergren, Niels Hald, and Fernando Bello. 2016. NOViSE: a virtual natural orifice transluminal endoscopic surgery simulator. *International Journal of Computer Assisted Radiology and Surgery* 11 (12 2016), 2303–2315. Issue 12. https://doi.org/10.1007/s11548-016-1401-8

Tassilo Kugelstadt, Dan Koschier, and Jan Bender. 2018. Fast corotated FEM using operator splitting. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 149–160.

Tang Liang, Wei-guo SONG, Tian-cheng HOU, Lei-lei LIU, Wei-xing CAO, and ZHU Yan. 2018. Collision detection of virtual plant based on bounding volume hierarchy: A case study on virtual wheat. *Journal of integrative agriculture* 17, 2 (2018), 306–314.

Sun Gyo Lim, Kwang Jae Lee, Kwang Suh, Seung Yeop Oh, Soon Kim, Junhwan Yoo, and Jeong Wi. 2013. Preoperative Colonoscopy for Detection of Synchronous Neoplasms after Insertion of Self-Expandable Metal Stents in Occlusive Colorectal Cancer: Comparison of Covered and Uncovered Stents. *Gut and liver* 7 (05 2013), 311–316. https://doi.org/10.5009/gnl.2013.7.3.311

Fuchang Liu and Young J Kim. 2013. Exact and adaptive signed distance fieldscomputation for rigid and deformablemodels on gpus. *IEEE transactions on visualization and computer graphics* 20, 5 (2013), 714–725.

Anderson Maciel, Ronan Boulic, and Daniel Thalmann. 2007. Efficient collision detection within deforming spherical sliding contact. *IEEE Transactions on Visualization and Computer Graphics* 13, 3 (2007), 518–529.

Anderson Maciel and Suvranu De. 2008. An efficient dynamic point algorithm for line-based collision detection in real time virtual environments involving haptics. *Computer animation and virtual worlds* 19, 2 (2008), 151–163.

Miles Macklin, Matthias Müller, and Nuttapong Chentanez. 2016. XPBD: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*. 49–54.

Marilena Maule, Anderson Maciel, and Luciana Nedel. 2010. Efficient collision detection and physics-based deformation for haptic simulation with local spherical hash. In *2010 23rd Conference on Graphics, Patterns and Images*. IEEE, 9–16.

Nathan Mitchell, Court Cutting, and Eftychios Sifakis. 2015. GRIDiron: an interactive authoring and cognitive training foundation for reconstructive plastic surgery procedures. *ACM Trans. Graph.* 34, 4, Article 43 (jul 2015), 12 pages.

Wouter Mollemans, Filip Schutyser, Johan Van Cleynenbreugel, and Paul Suetens. 2003. Tetrahedral mass spring model for fast soft tissue deformation. In *International Symposium on Surgery Simulation and Soft Tissue Modeling*. Springer, 145–154.

Lucas Zanusso Morais, Victor Kunde Bergmann, Eduarda Abreu Carvalho, Raquel Zimmer, Marcelo Gomes Martins, Luciana Porcher Nedel, Anderson Maciel, and Rafael Piccin Torchelsen. 2023. An enhanced interactive endoscope model based on position-based dynamics and Cosserat rods for colonoscopy simulation. *Computers & Graphics* 116 (2023), 345–353.

Matthias Müller, Miles Macklin, Nuttapong Chentanez, Stefan Jeschke, and Tae-Yong Kim. 2020. Detailed rigid body simulation with extended position based dynamics. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 101–112.

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation* 18 (4 2007), 109–118. Issue 2. https://doi.org/10.1016/j.jvcir.2007.01.005

Jean-Christophe Nebel. 2001. Soft tissue modelling from 3D scanned data. In *Deformable avatars*. Springer, 85–97.

Mohd N Omar, Nasrul H Johari, Mohd H A Hassan, and Mohd A Azizan. 2022. Modeling of Soft Tissue Deformation Using Mass Spring Method with Nonlinear Volume Force. In *Human-Centered Technology for a Better Tomorrow*. Springer, Singapore, 75–90.

Jun J. Pan, Jian Chang, Xiaosong Yang, Hui Liang, Jian J. Zhang, Tahseen Qureshi, Robert Howell, and Tamas Hickish. 2015. Virtual reality training and assessment in laparoscopic rectum surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery* 11 (2015), 194–209. Issue 2.

Paolo Patete, Maria Ida Iacono, Maria Francesca Spadea, Giovanna Trecate, Daniele Vergnaghi, Luca Tommaso Mainardi, and Guido Baroni. 2013. A multi-tissue mass-spring model for computer assisted breast surgery. *Medical engineering & physics* 35, 1 (2013), 47–53.

Soraia Figueiredo Paulo, Nuno Figueiredo, Joaquim Armando Jorge, and Daniel Simões Lopes. 2018. 3D Reconstruction of CT Colonography Models for VR/AR Applications using Free Software Tools. (2018).

Mark B. Pochapin. 2011. *Colonoscopy Video Tour: Journey Through a Healthy Colon*. AmCollege Gastro. https://youtu.be/yXCYHjbKenA?t=160

Hiroaki Saito, Tetsuya Tanimoto, Tsuyoshi Ozawa, Soichiro Ishihara, Mitsuhiro Fujishiro, Satoki Shichijo, Dai Hirasawa, Tomoki Matsuda, Yuma Endo, and Tomohiro Tada. 2020. Automatic anatomical classification of colonoscopic images using deep convolutional neural networks. *Gastroenterology Report* 9, 3 (12 2020), 226–233.

Johannes Schauer and Andreas Nüchter. 2015. Collision detection between point clouds using an efficient kd tree implementation. *Advanced Engineering Informatics* 29, 3 (2015), 440–458.

Ygor Rebouças Serpa and Maria Andréia Formico Rodrigues. 2019. Flexible Use of Temporal and Spatial Reasoning for Fast and Scalable CPU Broad-Phase Collision Detection Using KD-Trees. *Computer Graphics Forum* (2019).

S. G. Shah, J. Brooker, C. Thapar, C. Williams, and B. Saunders. 2002. Patient pain during colonoscopy: an analysis using real-time magnetic endoscope imaging. *Endoscopy* 34 6 (2002), 435–40. https://doi.org/10.1055/S-2002-31995

Rebecca L. Siegel, Kimberly D. Miller, Ann Goding Sauer, Stacey A. Fedewa, Lynn F. Butterly, Joseph C. Anderson, Andrea Cercek, Robert A. Smith, and Ahmedin Jemal. 2020. Colorectal cancer statistics, 2020. *CA: A Cancer Journal for Clinicians* 70 (5 2020), 145–164. Issue 3. https://doi.org/10.3322/caac.21601

J. Spillmann and M. Teschner. 2007. CoRdE: Cosserat Rod Elements for the Dynamic Simulation of One-Dimensional Elastic Objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '07)*. Eurographics Association, Goslar, DEU, 63–72.

A. Erman Tekkaya and Celal Soyarslan. 2018. *Finite Element Method*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–8.

Matthias Teschner, Stefan Kimmerle, Bruno Heidelberger, Gabriel Zachmann, Laks Raghupathi, Arnulph Fuhrmann, M-P Cani, François Faure, Nadia Magnenat-Thalmann, Wolfgang Strasser, et al. 2005. Collision detection for deformable objects. In *Computer graphics forum*, Vol. 24. Wiley Online Library, 61–81.

Yuichi Torii. 2003. U.S. Patent No. 6641528B2. U.S. Patent and Trademark Office, Washington, DC. https://patents.google.com/patent/US6641528B2.

SE Van der Wiel, R Küttner Magalhães, Carla Rolanda Rocha Gonçalves, Ma Dinis-Ribeiro, MJ Bruno, and AD Koch. 2016. Simulator training in gastrointestinal endoscopy–From basic training to advanced endoscopic procedures. *Best Practice & Research Clinical Gastroenterology* 30, 3 (2016), 375–387.

Hans De Visser, Josh Passenger, David Conlan, Christoph Russ, David Hellier, Mario Cheng, Oscar Acosta, Sébastien Ourselin, and Olivier Salvado. 2010. DEVELOPING A NEXT GENERATION COLONOSCOPY SIMULATOR. *International Journal of Image and Graphics* 10 (4 2010), 203–217. Issue 2.

Monan Wang and Jiaqi Cao. 2021. A review of collision detection for deformable objects. *Computer Animation and Virtual Worlds* 32, 5 (2021), e1987.

Zhuang Yan and J Canny. 1999. Real-time Simulation of Physically Realistic Global Deformation. *IEEE Vis' 99. San Francisco, California* (1999), 24–29.

Zhennan Yan, Lixu Gu, Pengfei Huang, Sizhe Lv, Xiao Yu, and Xianming Kong. 2007. Soft tissue deformation simulation in virtual surgery using nonlinear finite element method. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 3642–3645.

Xiufen Ye, Xinkui Mei, and Shuguo Xiao. 2018. Filling model based soft tissue deformation model. In *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 1655–1659.

Xiufen Ye, Jianguo Zhang, Peng Li, Tian Wang, and Shuxiang Guo. 2016. A fast and stable vascular deformation scheme for interventional surgery training system. *Biomedical engineering online* 15 (2016), 1–14.

SY Yi, HS Woo, WJ Ahn, JY Kwon, and DY Lee. 2006. New colonoscopy simulator with improved haptic fidelity. *Advanced Robotics* 20, 3 (2006), 349–365.

Jinao Zhang, Yongmin Zhong, and Chengfan Gu. 2017. Deformable models for surgical simulation: a survey. *IEEE reviews in biomedical engineering* 11 (2017), 143–164.

Shaoting Zhang, Andrew Nealen, and Dimitris Metaxas. 2010. Skeleton Based As-Rigid-As-Possible Volume Modeling. In *Eurographics 2010 - Short Papers*, H. P. A. Lensch and S. Seipel (Eds.). The EG Association.

Wenlan Zhang, Xin Liu, and Bin Zheng. 2021. Virtual reality simulation in training endoscopic skills: A systematic review. *Laparoscopic, Endoscopic and Robotic Surgery* 4 (2021), 97–104. Issue 4. https://doi.org/10.1016/j.lers.2021.09.002

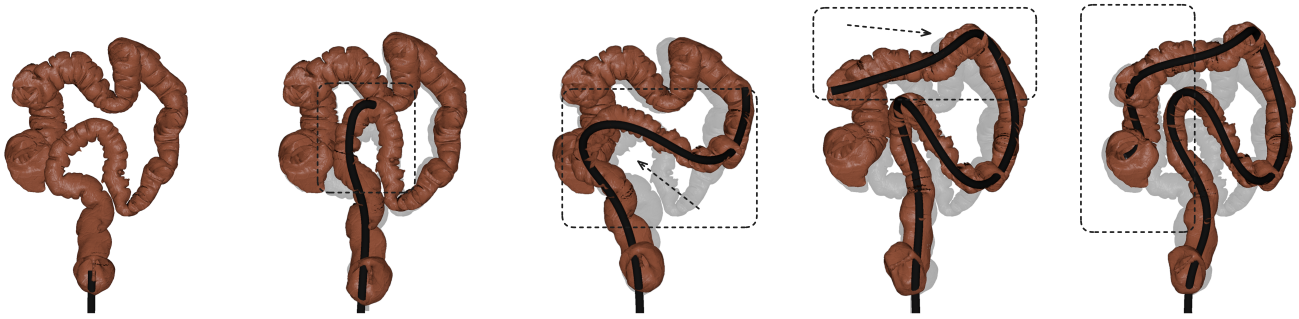

Figure 10: Progress of the endoscope insertion: (a) in the rectum where there is no perceived deformation; (b) the tip of the endoscope bending to the right while pressing on the upper part of the sigmoid, which deforms in consequence; (c) the sigmoid tightening that challenges the passage of the scope; (d) the scope moving towards the transverse colon, where entanglement also occurs; and (e) the endoscope reaches the cecum. Note that the colon retains its overall position in the abdomen.

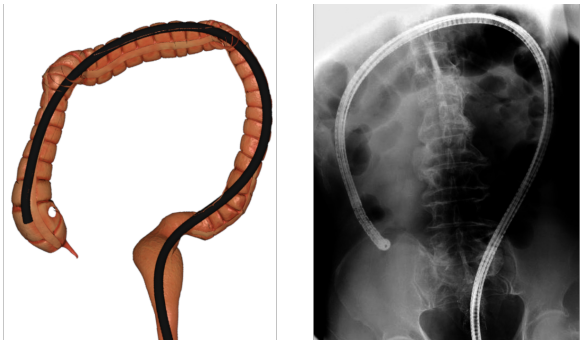

Figure 11: Comparison with an X-ray image demonstrates the similarity of the resulting shape. The X-ray image by Lim et al. [2013] is available under the terms of the Creative Commons Attribution Non-Commercial License.

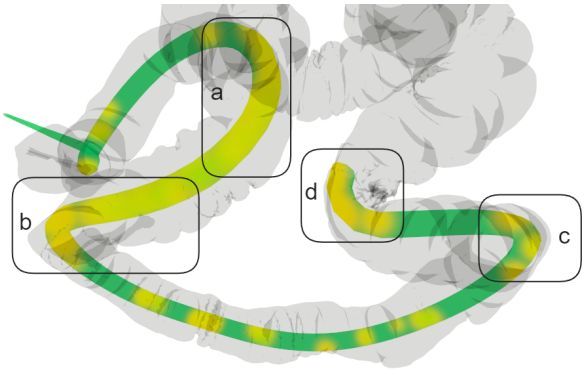

Figure 13: Warmer colors indicate the areas of higher compression between the colon and the endoscope. Letters *a* to *d* mark the regions where medical literature [Kavic and Basson 2001] indicates higher chances of perforation, correlating with our compression results.

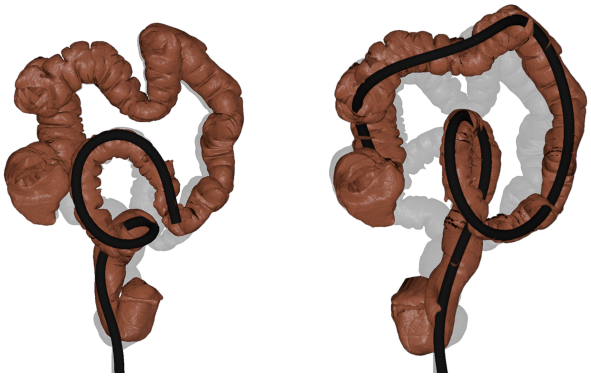

Figure 12: Two types of loops typically occur during surgery, that need to be addressed by the surgeon (alpha-loop (right) and inverted alpha-loop (left)). The formation of loops in the sigmoid is critical in colonoscopy, representing 77% of patient complaints related to pain [Shah et al. 2002].

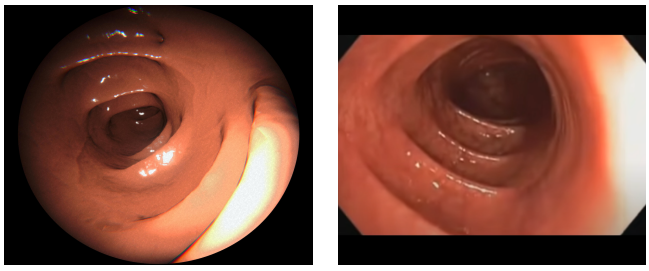

Figure 14: Rendering effects of the intra-colon environment. (a) depicts our final rendering with post-processing effects. In (b), a video frame from an actual colonoscopy [Pochapin 2011] is shown for comparison.