DOI: 10.1111/cgf.142646
EUROGRAPHICS 2021 / N. Mitra and I. Viola
(Guest Editors)

Volume 40 (2021), Number 2

Physically-based Book Simulation with Freeform Developable Surfaces

Thomas Wolf[1], Victor Cornillère[1] and Olga Sorkine-Hornung[1]

[1]ETH Zurich, Switzerland

Abstract

Reading books or articles digitally has become accessible and widespread thanks to the large amount of affordable mobile devices and distribution platforms. However, little effort has been devoted to improving the digital book reading experience, despite studies showing disadvantages of digital text media consumption, such as diminished memory recall and enjoyment, compared to physical books. In addition, a vast amount of physical, printed books of interest exist, many of them rare and not easily physically accessible, such as out-of-print art books, first editions, or historical tomes secured in museums. Digital replicas of such books are typically either purely text based, or consist of photographed pages, where much of the essence of leafing through and experiencing the actual artifact is lost. In this work, we devise a method to recreate the experience of reading and interacting with a physical book in a digital 3D environment. Leveraging recent work on static modeling of freeform developable surfaces, which exhibit paper-like properties, we design a method for dynamic physical simulation of such surfaces, accounting for gravity and handling collisions to simulate pages in a book. We propose a mix of 2D and 3D models, specifically tailored to represent books to achieve a computationally fast simulation, running in real time on mobile devices. Our system enables users to lift, bend and flip book pages by holding them at arbitrary locations and provides a holistic interactive experience of a virtual 3D book.

1. Introduction

Mobile devices, such as tablets, smartphones or ebook-readers are common nowadays and make it easy for anyone to read in various environments. From the hardware perspective, many factors that elevate the reading experience are being continuously improved: e.g., portability, battery life and screen quality. On the other hand, reading applications have hardly changed, despite studies like [Cli19] showing that reading performance in terms of comprehension and recall appears worse on digital screens compared to physical books. The different sensorimotor cues of digital devices [MOV19] and missing visual anchors [HRL17] might be possible reasons for this effect. It is therefore plausible that a more realistic book representation on digital devices could diminish these issues by making digital reading a more stimulating sensorimotor experience and providing additional visual cues. Furthermore, improving the digital reading environment might also make it more attractive for traditional print publishers to share their publications on this medium, as well as make rare books more accessible and enable readers to experience them in a closer-to-reality way.

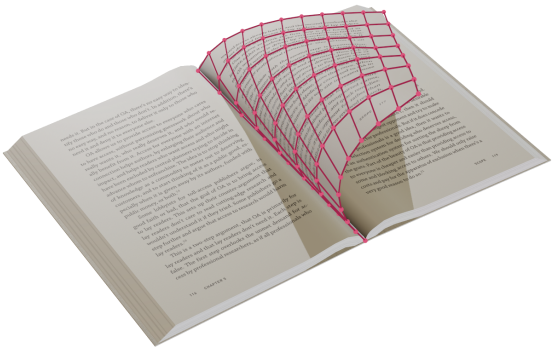


Figure 1: *Our book simulation employs a discrete developable surface to animate paper pages. The system comprises a fully interactive page and page blocks representing the rest of the book. This separation enables a realistic and interactive simulation running in real time on low compute power devices, such as mobile phones.*

We are motivated by these considerations and in this work, we propose a method for realistic, interactive simulation of physical books. For this purpose, we utilize the discrete model for freeform developable surfaces recently proposed by Rabinovich et al. [RHS18a]. Isometric deformations of developable surfaces exhibit paper-like properties and are thus well suited to represent book pages. We demonstrate how the static formulation of discrete developable surfaces can be used in a dynamic physics based simula-

tion framework. We employ our simulation method to approximate the behaviour of a life-like book page, which the user can interact with in real time. To ensure an overall realtime simulation speed for the entire book, we propose a factorized book representation: The page being manipulated by the user is simulated fully in 3D, and it seamlessly integrates with the rest of the book, whose simulation is carried out "en bloc" in 2D and extruded to create a realistic 3D appearance (Fig. 2). This allows our application to run smoothly on mobile devices with limited computational power. The simulation system is parameterized by essential characteristics of a book's physical composition, such as the material properties of the paper and the book binding, so that the appearance and dynamic behavior of the simulated book can closely imitate the physical original.

Our main contributions are as follows:

- The formulation of an interactive physics-based simulation of a developable surface in 3D, imitating paper behavior;
- The coupled decomposition of the global simulation of a book into a full 3D page and extruded page blocks representing the rest of the book;
- An application demonstrating intuitive user interaction for digital reading.

2. Related work

To frame the motivation for this work, we briefly review results from psychology literature comparing reading on paper with reading on digital devices, as well as related works in human computer interaction that deal with visual representation and user interface of digital books. We also discuss the prior art forming the algorithmic foundation of our method, namely cloth and thin shell simulation and developable surface models.

Reading on paper vs. reading on screens. Clinton et al. [Cli19] review 33 different studies between the years 2008-2018 concerning reading performance on paper compared to screens. When reading on displays, they note a significant negative impact on retention of information. Hou et al. [HRL17] attempt to explain these differences with two concepts: the *cognitive map mechanism* and the *medium materiality mechanism*. The idea of the former is that the way a text is embedded in a physical landscape can help (or hinder) the construction of a cognitive map to navigate, remember and understand the text. The medium materiality mechanism states that the cognitive processing is influenced by different material characteristics, such as screen flickering, display quality, etc. While the results do not sufficiently support the medium materiality mechanism, the cognitive map mechanism is confirmed by achieving worse reading comprehension and immersion, as well as feelings of fatigue when reading a partial view of a graphic novel page on a tablet compared to viewing the full page on a tablet or on paper. Hou and colleagues conclude that:

> "The present findings implied that reading on a screen could match that of reading from paper if the representation of the document on electronic reading devices resembles that of the printed book."

The medium materiality mechanism is also investigated by Kretzschmar et al. [KPH*13], who measure and compare brain activity, eye fixation duration and text comprehension of young and older individuals between e-readers, tablets and printed books. The study shows no difference in these metrics for younger people, but a faster fixation and less brain activity on digital devices for older people – indicating that less effort is needed for them to read on screens, likely due to the higher contrast of digital displays. However, both groups strongly preferred reading on paper.

Mangen et al. [MOV19] compared the sessions of 50 participants reading long texts of about one hour on Kindle DX and a pocket book, by asking factual questions regarding the text after the reading session. The mean number of correct responses was roughly the same for most categories of questions, with the exception of "time and temporality" category, where print readers performed significantly better. Additionally, on questions about where something happened in the text, participants of the Kindle group performed worse for questions concerning the beginning. Finally, the task of arranging the chronological events was solved substantially better by print readers.

Digital books. Most e-readers developed so far concentrate more on meta-features than on realism in their system design. Card et al. [CHMC04] propose a 3D book representation focused on advanced search, bookmarking and annotation features. Their model lacks in interactivity, as their page animations are simple handcrafted transitions played when the user presses on a page to see the next one. Hong et al. [HCC06] improve on the visual realism of page turning. In their system, the user clicks on a corner of the page, and a page turning animation is automatically played, computed by using a cone to deform the page in motion. Kim et al. [KKL13] build a 2D application optimized for comfortable book navigation, which relies heavily on intuitive touch gestures. A precomputed page swipe animation is used when pages are flipped over. Kotajima et al. [KT16] go a step further to make the navigation of a digital book on a tablet feel closer to printed books: they employ a combination of touch gestures and pressure sensors on the back. A stack of pages opens when both sensors register a higher pressure, and with touchscreen gestures, pages can then be flipped or bookmarked when one sensor is tapped twice. The page animations in this system are achieved by simple 2D texture manipulations.

Several publications attempt to bridge the gap between real printed books and their digital equivalents. Chu et al. [CWLB03; CBJW04] strive to build a realistic 3D model with page turning animations obtained from a costly simulation with a mass-spring system. However, in their system, when the user grabs a page, there is no real-time physical interaction due to the computational complexity. Instead, the system selects one of the pre-calculated animations and plays it as the user moves the page to the side. Liesaputra et al. [LWB09; LW12] also position their system as a realistic electronic book. Their page turning is interactive, but it is limited to rotating cropped 2D textures to give the illusion of a moving page.

None of the previous works tackle real-time 3D physical simulation and interactive manipulation of book pages.

Cloth and thin shell simulation. Book pages are thin shells that admit no stretch or compression. The simulation of thin shells such as cloth is addressed in a vast number of works in computer graphics, many of which follow the seminal approach by Baraff and

Witkin [BW98]. In their work, triangle meshes representing cloth are simulated using implicit Euler integration after linearizing the forces. To improve the dynamic behavior of the cloth, new energy terms have been designed, such as the bending energy for discrete shells proposed by Grinspun et al. [GHDS03]. Discrete shells use the sum of squared differences of the mean curvature between the rest shape and the deformed surface as a way of measuring cloth bending strain. Representing paper pages with these cloth simulation algorithms is challenging because of the required infinite stiffness to simulate an inextensible thin shell, making the simulation very unstable. These standard approaches also suffer from locking issues, preventing the cloth from bending in the required directions and necessitating remeshing [CB98; Ale12].

To improve the performance of cloth simulation and reduce numerical strain, Goldenthal et al. [GHF*07] model cloth using Chebyshev nets: regular quad meshes where both pairs of opposing edges in each quad should have the same length. They solve the constrained dynamic system by first performing an unconstrained step and then projecting onto the space of Chebyshev net constraints. The projection is modelled as an optimization problem and solved iteratively via repeated linearizations. The approach shows a stable simulation while exhibiting only low cloth strain. However, shearing is still inherently part of this model and is therefore not entirely suitable for simulating paper. We are inspired by their constrained simulation approach but employ a different surface model: discrete orthogonal geodesic nets [RHS18a] as opposed to Chebyshev nets, which enables realistic paper simulation.

Developable surfaces. A book page is a thin sheet of paper that can be described by a developable surface isometric to a planar rectangle. It can bend or crease but not stretch or compress, and all its configurations are the result of a series of bending operations on a planar sheet. Smooth developable surfaces can be locally characterized by vanishing Gaussian curvature at every point, but they are global by nature due to being ruled surfaces with constant normals along each ruling. As a result, their computational modeling and simulation has been a challenging subject of extensive research. One notable issue when simulating developable surfaces is locking, already discussed in the context of cloth simulation above [CB98; Ale12]. If the model for developable surfaces relies on its global characterization, such as the arrangement of the rulings [SVWG12; TBWP16], then smoothly transitioning from one developable shape to another in this representation may be impossible.

One solution to avoid this locking problem is to constantly remesh the surface. In the work of Schreck et al. [SRH*15], paper sheets are animated by first performing simulation without enforcing developability, followed by remeshing and finally enforcing local developability and isometry. The method shows crumpling and bending similar to real paper at interactive rates. However, no global notion of discrete developability is established, and the constant remeshing of the sheet increases the complexity of the simulation and creates lasting artifacts, such as folds that might not be desirable in a reading application.

For our purposes, we found the discrete developable model defined by Rabinovich et al. [RHS18a] to be well-suited. They build a class of surfaces called discrete orthogonal geodesic nets (or DOGs) that are regular quad grids, where discrete developability is enforced by constraining the angles around each vertex to be all equal. Since this formulation is local, it does not suffer from locking even at very low resolutions of the net, and the simplicity of the constraints makes DOGs attractive for our simulation system. Concurrently to our work, Jiang et al. [JWR*20] developed a new model for discrete developable surfaces that shares the advantages of Rabinovich et al. [RHS18a] and also offers accurate modeling of discrete isometry. This approach could be an interesting alternative way of representing developable surfaces for our setting.

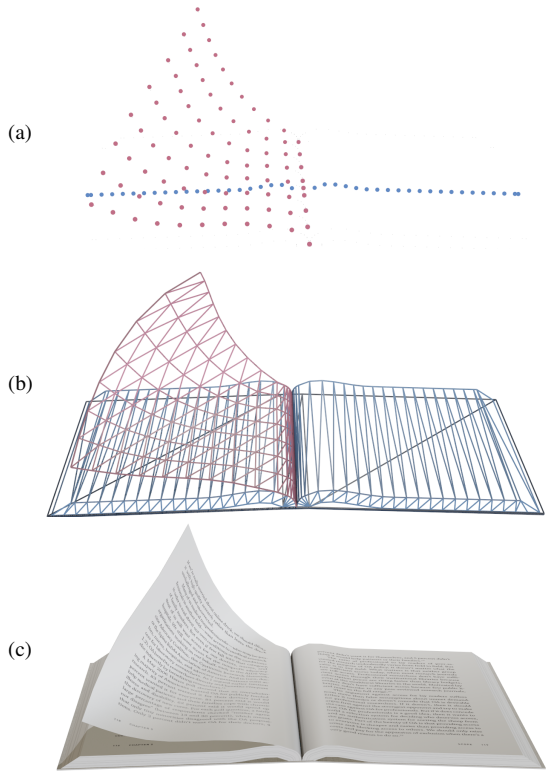


Figure 2: *Method overview.* (a) *Our simulation works with a set of nodes (point masses). They are divided into nodes modeling the active page (in red), simulated as a developable sheet in 3D, and nodes modeling the resting page blocks (in blue), as well as the book cover. Those nodes lie on a planar polyline representing the midline curve of the page blocks, and their Z coordinate remains constant throughout the simulation.* (b) *To visualize the book, we generate a triangle mesh out of the nodes by offsetting and extrusion.* (c) *Textured book rendering.*

3. Digital book model

To create a faithful digital simulation of a book, ideally we would like to simulate the book in its entirety: all individual pages, the covers and the binding with the appropriate book construction. However, such a simulation of a typical book with a few hundred pages is prohibitively expensive for an interactive application. We therefore focus on a typical reading scenario, where the book rests on a stable surface and the reader interacts with one page at a time

(or a stack of consecutive pages) by holding the edge and turning the page(s). The 3D shape of the *active page* may change dramatically during the interactive manipulation, while the shapes of the remaining pages typically stay rather still in comparison. Due to the binding along a straight line and the uniformity of the material of the book pages, the shape of these resting *page blocks* can be described by extruded generalized cylinders.

Based on these observations we allocate most simulation resources to the active page and represent it as a *freeform* discrete developable surface; the page blocks, on the other hand, are represented by their *planar midline curve*, which follows the bottom of the page block and is extruded both horizontally and away from the cover to create the 3D shape of the page block; see Fig. 2 for an illustration. Such a factorization enables a very efficient simulation even on low compute power devices like smartphones and tablets. The extrusion still allows to realistically model shear in the page blocks when pages shift; similarly, the active page surface can represent more than a single page, since a stack of manipulated pages can be created by extrusion (see Fig. 8).

We describe the interactive simulation framework for this lean, factorized representation in Sec. 3.1-3.3 and collision handling in Sec. 3.4. We then explain how we create the visual book representation (a triangle mesh) for rendering purposes in Sec. 3.5.

3.1. Simulation setup

We assume books of rectangular shape and set our coordinate axes in 3D such that the X-axis aligns with the horizontal edge of the book, the Y-axis is the direction along which the book pages are stacked, and the Z-axis is parallel to the binding. The planar curve representing a page block lives in the XY-plane, and we later extrude it in the Z direction, as well as offset in the XY-plane, to generate the 3D shape of the page block for rendering. We assume that the book lies on a surface in the XZ-plane, i.e., gravity is parallel to the Y-axis.

Our simulation routine continuously updates the 3D locations of N *simulation nodes* denoted by $x_i \in \mathbb{R}^3$; the vector of all node coordinates stacked together is denoted by $x \in \mathbb{R}^{3N}$. The nodes are the union of the following sets, simulated all together:

1. Vertices of the discrete developable surface representing the active page. We employ discrete orthogonal geodesic nets (DOGs), whose vertices are arranged in a regular quad grid connectivity [RHS18a].
2. Vertices of the polyline representing the midline curves of the page blocks lying to the left and to the right of the active page. This polyline is kept planar, i.e., the Z coordinates of its nodes are maintained constant throughout the simulation. An edge in the middle of this polyline represents the *binding* of the book, to which one side of each book page is glued.
3. If the book has a hard cover, we represent the cover separately via two straight segments emanating from the vertices of the midline corresponding to the binding. These long edges are visible in Fig. 4 below the page block curves.

The connectivity relationships between the nodes are modeled by the constraints and the potentials in our simulation, see Fig. 3, 4. Each node x_i has an associated mass m_i. We denote by M the diagonal mass matrix of our system. The active page nodes' masses sum up to the prescribed weight of a single page, and the masses of the midline nodes of each page block sum up to the weight of the stack of pages represented by the block. The weights are distributed uniformly among the nodes in each set.

Starting from a known configuration at time $t = 0$, the positions of the nodes x evolve according to the dynamics equations

$$\begin{aligned} M\ddot{x} &= -\nabla V(x) - \nabla C(x)^{\mathsf{T}} \lambda, \\ C(x) &= 0, \end{aligned} \tag{1}$$

where $V(x)$ is a weighted sum of all potentials (Sec. 3.3) and $C(x)$ is the vector of all constraint functions (Sec. 3.2); λ are the Lagrange multipliers used to solve our constrained optimization problem. Recall that the Z-coordinates of the midline curve nodes and the nodes of the active page that are attached to the binding remain fixed and are not updated in the simulation, unlike the remaining nodes of the active page, which are fully 3D.

We time-discretize the above equations and implicitly integrate the node positions x and their velocities v. Compared to [GHF*07], we treat not only the constraint forces $-\nabla C^{\mathsf{T}}\lambda$ implicitly, but also the potential forces $-\nabla V$ to achieve a stable simulation. Let $h = t^{(k)} - t^{(k-1)}$ be the time step size; discretizing $\ddot{x}$ in Eq. 1 leads to

$$\begin{aligned} v^{(k)} &= v^{(k-1)} + hM^{-1}\left(-\nabla V(x^{(k)}) - \nabla C(x^{(k)})^{\mathsf{T}}\lambda^{(k)}\right), \\ x^{(k)} &= x^{(k-1)} + h v^{(k)}, \\ C(x^{(k)}) &= 0. \end{aligned} \tag{2}$$

We combine these equations to obtain

$$F_k\left(x^{(k)}, \lambda^{(k)}\right) = 0, \quad \text{where} \tag{3}$$

$$F_k(x,\lambda) = \begin{bmatrix} x - x^{(k-1)} - h v^{(k-1)} + h^2 M^{-1}\left(\nabla V(x) + \nabla C(x)^{\mathsf{T}}\lambda\right) \\ C(x) \end{bmatrix},$$

which needs to be solved for $x^{(k)}, \lambda^{(k)}$.

Optimization. The nonlinear function F_k in Eq. 3 is differentiable, and its Jacobian is invertible for a small enough h. This enables us to find the roots via Newton's method at each time step k. Each Newton iteration can be written as:

$$\begin{bmatrix} x^{(k)}_{(j)} \\ \lambda^{(k)}_{(j)} \end{bmatrix} = \begin{bmatrix} x^{(k)}_{(j-1)} \\ \lambda^{(k)}_{(j-1)} \end{bmatrix} - DF\left(x^{(k)}_{(j-1)}, \lambda^{(k)}_{(j-1)}\right)^{-1} F_k\left(x^{(k)}_{(j-1)}, \lambda^{(k)}_{(j-1)}\right),$$

where

$$DF(x,\lambda) = \begin{bmatrix} I + h^2 M^{-1}(\nabla^2 V(x) + \sum_l \nabla^2 C_l(x)\lambda) & h^2 M^{-1}\nabla C(x)^{\mathsf{T}} \\ \nabla C(x) & 0 \end{bmatrix},$$

where $x^{(k)}_{(j)}, \lambda^{(k)}_{(j)}$ are the variable values in the j-th Newton iteration of time step k and $C_l(x)$, $l = 1, 2, \ldots$ are the individual components of the vector of constraint functions. As initial value, we use the node positions from the previous time step and $\lambda = 0$.

In practice, for small spatial discretization resolutions, we find that one Newton step is sufficient for visually pleasing results. This

leads to a simplified equation, where we can leave out the constraint Hessians, since the starting value of λ is 0, and we thus get the following update step:

$$x^{(k)} = x^{(k-1)} - DF\left(x^{(k-1)},0\right)^{-1} F_k\left(x^{(k-1)},0\right). \tag{4}$$

We note that $\nabla^2 V(x), \nabla C(x)$ and therefore $DF(x,\lambda)$ are sparse, leading to efficient computations, since our potentials and constraints have very local formulations, detailed in Sec. 3.2-3.3. The matrix $DF(x,\lambda)$ is in general not positive definite, hence we perform a line search to avoid bad Newton directions. We use the same merit function as in [RHS19]. Also note that we do not solve for the Z-coordinates of the midline nodes and the cover nodes. They remain constant during the simulation. We include them in our equations for the sake of notation simplicity.

3.2. Simulation constraints

Our vector of constraint functions $C(x)$ contains the various terms that impose a plausible book behavior in our simulation, including discrete developability of the active page C_{DOG} and inextensibility of the page block midline C_{iso2D}, as well as proper construction of the book C_{binding}, i.e., gluing of the pages to the binding:

$$C = \begin{bmatrix} C_{\text{DOG}} & C_{\text{iso2D}} & C_{\text{binding}} & C_{\text{fix}} \end{bmatrix}^{\mathsf{T}}. \tag{5}$$

The last term is a positional constraint that ensures that the book does not move around when turning pages. We detail the terms below.

Discrete developability constraints. To model the developability of the active page, we use a discrete orthogonal geodesic net (DOG), described in [RHS18a]. The nodes of the active page are arranged in regular quad grid topology, and the constraints that make the surface discrete-developable in this model are that for each node, all angles between consecutive emanating edges are equal, see Fig. 3. Therefore, the term C_{DOG} is a vector of angle-constraint functions, as detailed in [RHS18b]. Note that the DOG developability model does not include isometry, i.e., the developable surface may change its size while staying developable, which can be advantageous in freeform shape modeling, but is not desirable in our book simulation. The DOG model does not have sufficient amount of degrees of freedom to enforce isometry with hard constraints [RHS18a], so we include a soft isometry constraint in form of a potential (see Sec. 3.3).

Length constraints. The developability of the "virtual" pages in our page blocks is ensured by construction, since we extrude a planar curve and thus obtain a generalized cylinder. We disallow length changes of the midline curve in the simulation via the constraint C_{iso2D} and thereby enforce isometry. Let x_i, x_j be two adjacent nodes on the midline polyline, or the two nodes representing either hard book cover; we add a constraint

$$C_{\text{iso2D}(i,j)} = \|x_i - x_j\|^2 - \|x_i^0 - x_j^0\|^2 = 0, \tag{6}$$

where x_*^0 refers to the initial position of a node.

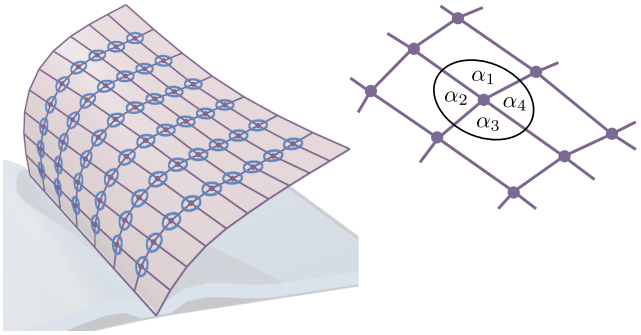


Figure 3: *Each inner node of the active page (in red) has a DOG constraint: all angles between each node's consecutive emanating edges are equal,* $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$.

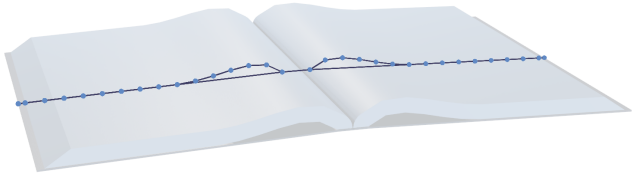


Figure 4: *The nodes of the page block midline and the book cover are connected via length constraints (dark blue edges).*

Binding constraints. Let x_{b_1}, x_{b_2} denote the two adjacent nodes on the midline polyline that represent the book binding (dark blue in Fig. 5). We wish to fixate one side of the active page to the binding at an appropriate location parameter $q \in [0,1]$ along this *binding edge*. In other words, if we project all simulation nodes to the XY plane, then each boundary vertex x_i of the active page on the side that is glued to the binding should end up in the location $(1-q)\,x_{b_1} + q\,x_{b_2}$ on the projected binding edge:

$$\begin{aligned} C_{\text{binding}(i),X} &= x_i^X - \left((1-q)\,x_{b_1}^X + q\,x_{b_2}^X\right) = 0, \\ C_{\text{binding}(i),Y} &= x_i^Y - \left((1-q)\,x_{b_1}^Y + q\,x_{b_2}^Y\right) = 0. \end{aligned} \tag{7}$$

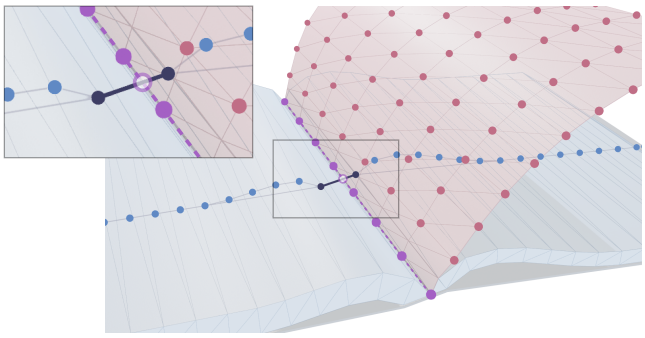


Figure 5: *We fix the XY-projection of the side nodes of the active page (in purple) to the binding edge (dark blue) to keep the active page attached to the rest of the book.*

Here, the superscripts x_*^X, x_*^Y denote the X or Y coordinates of a node, respectively. The parameter q is fixed for each frame. It is equal to the number of pages of the left page block plus half the number of active pages, divided by the total number of pages in the book. This puts the active page in its correct position based on the global reading progress.

Fix constraint. We fixate the whole book at the center of the binding edge:

$$C_{\text{fix}} = 0.5(x_{b_1} + x_{b_2}) - p_{\text{fix}} \tag{8}$$

to prevent the user from pulling the book away from its place. The book can still be rotated if desired.

3.3. Potentials

The potential function $V(x)$ contains a weighted sum of the different terms defining the paper and book construction properties and the behavior of our simulation system: the bending energy of the book pages V_{bending}, the binding glue energy V_{glue} that simulates the strength of the glue at the binding holding pages together, as well as the gravity term V_{gravity}. We add two soft constraint terms that cannot be modeled as hard constraints due to lack of degrees of freedom in our chosen discrete developable surface model, namely the isometry term V_{iso3D} for the active page and the positional objective V_{pos} for the grabbing interaction by the user.

$$V = V_{\text{bending}} + w_{\text{glue}} V_{\text{glue}} + V_{\text{gravity}} + w_{\text{iso3D}} V_{\text{iso3D}} + w_{\text{pos}} V_{\text{pos}}. \tag{9}$$

The potentials are governed by weighting parameters w_*; bending consists of two separately weighted terms, as described below.

Bending energy. The bending energy consists of two parts. For the nodes of the planar midline polyline representing the page blocks, we use a discrete Lapacian L with edge length weights. For the 3D nodes of the active page we use the DOG Laplacian $\tilde{L}$, as proposed in [RHS18b]. Only inner node sets $\mathcal{S}_{\text{inner-curve}}$, $\mathcal{S}_{\text{inner-DOG}}$ participate in the bending energy.

$$\begin{aligned} V_{\text{bending}}(x) = w_{\text{bend2D}} \sum_{i \in \mathcal{S}_{\text{inner-curve}}} \|L(x_i)\|^2 + \\ w_{\text{bend3D}} \sum_{i \in \mathcal{S}_{\text{inner-DOG}}} \|\tilde{L}(x_i)\|^2. \end{aligned} \tag{10}$$

Binding glue. Inspired by real-life bookmaking techniques, we add a term simulating the glue strength at the binding, which holds the pages upright even when the open book lies flat on a surface. The term measures the angles between the normal of the binding edge n and the edges of the midline polyline emanating from the binding edge (see Fig. 6):

$$V_{\text{glue}} = (\langle n,\, d_{\text{left}} \rangle - 1)^2 + (\langle n,\, d_{\text{right}} \rangle - 1)^2, \tag{11}$$

where d_{left}, d_{right} are the corresponding normalized midline edges.

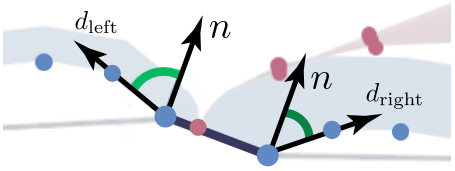


Figure 6: *To simulate the glue in the binding holding the pages up, we minimize the angles between the normal of the binding n and the tangents of the adjacent page blocks d_{left} and d_{right}.*

Gravity. In our system gravity aligns to the Y-axis, so the potential energy is expressed as

$$V_{\text{gravity}}(x) = g \sum_{i=0}^{N} m_i x_i^Y, \tag{12}$$

where g is the gravity acceleration.

Isometry objective. To keep the size of the active page stable, we use an isometry objective similar to the 2D length constraints described earlier. For each pair of nodes of the active page, x_i, x_j, that are neighbors on the DOG grid, we add the objective

$$V_{\text{iso3D}(i,j)} = \left(\|x_i - x_j\|^2 - \|x_i^0 - x_j^0\|^2 \right)^2, \tag{13}$$

where x_*^0 refers to the initial node position.

Positional objective. When the user grabs node x_i of the active page and drags it to the desired position $p \in \mathbb{R}^3$, a positional objective of the following form is added:

$$V_{\text{pos}} = \|x_i - p\|^2. \tag{14}$$

Since our application uses a screen, the user input is in the 2D screen space. We determine i by casting a ray from the camera and finding the node x_i of the active page that is one segment away from the page border and is closest to the ray. We avoid positional constraints on the boundary of the DOG, since these tend to produce smoothness artifacts. As the user drags the grabbed node around,

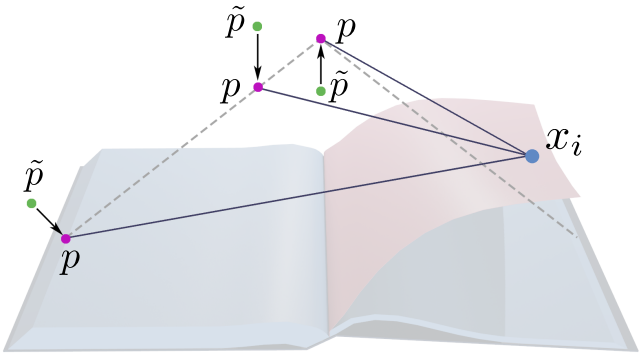


Figure 7: *Implementation of grabbing (positional) constraints. The user grabs a node (in blue) and drags it to various locations in screen space, transformed to 3D world coordinates (green points). As most such inputs do not correspond to physically achievable positional targets and lead to unnatural simulations, we project them to a tent function over the book (dotted gray line). These projections (purple points) are the positional targets we use in our simulation.*

we again cast a ray from this new screen position and find its intersection point $\tilde{p}$ with a plane passing through the original position of node i and with its normal being the forward direction of the camera. We then translate $\tilde{p}$ such that it is over the book and project it onto a tent function with the maximum over the binding to get the final target p, see Fig. 7. The purpose of this projection procedure is to avoid unrealistic positional targets that do not correspond to possible configurations of a book and could destabilize the simulation due to the quadratic scaling of the energy with the distance. The tent function enables a horizontal swipe on the screen to correspond to a curve in 3D space similar to the natural trajectory of a user turning a real book page. The shape of the tent function is determined empirically to induce a smooth, realistic-looking swiping animation. Other functions, such as arcs, could also be considered.

3.4. Collision handling

The collisions are handled after each simulation step. We first detect all intersections and then resolve them. We repeat these steps until convergence, but at most three times. To detect intersections between the active page and the page blocks, we project the nodes of the active page onto the XY dimensions of each segment of the midline polyline (we ignore the Z dimension since the page blocks are constant along it, so only the profile of the page block in the XY plane needs to be considered). We assume that the active page always lies above the page block and does not reach somewhere next to it. This is a reasonable assumption when using the book in a natural way. To account for the thickness of the page blocks and of the active page (which can represent several pages held by the user all at once), we check the distance of the nodes of midline polyline to the projected edges of the active page. When resolving the intersection, we check whether this distance is lower than the thickness of the page block plus half the thickness of the active page, and we correct the node positions of the active page by translating them along the segment normals. By using the initial configuration of the book, we can determine on which side of each node the page should be – this lets us avoid issues if a node passes through one of the blocks. In the same way, we can check and resolve the intersections between the page block and the cover, and between the cover and table. This collision handling method is computationally very inexpensive since it works in 2D and only involves point-line distances. We do not handle self-collisions, since they rarely occur in our setup and would result in a large performance impact. We also do not implement friction, as the only situation where it seems applicable is when the active page is bent such that its border touches a page block and the user lets go of the active page. However, we assume that the friction in that case would be negligible compared to the bending force of the active page. Given the purpose of our application, a user likely expects to see the page straightened quickly after releasing it, to facilitate reading comfort.

3.5. Meshing

We create a triangle mesh out of the nodes of the active page and the midline curve, which we can then render with the book materials and textures. The geometry of the book mesh is obtained via normal offsetting and extrusion. The nodes of the active page are offset by half the target thickness (which depends on the number of grabbed pages the active page represents) along the positive and negative directions of the vertex normals. The midline curve of the page blocks is offset in the XY plane by the node normals. We then extrude along the Z-axis to complete the 3D representation of the page blocks.

The 3D objects resulting from the above procedure do not yet possess a fully realistic geometry, as can be see in Fig. 9 (top): the page stacks lack curvature detail near the binding due to the uniform mesh density, and due to the orthogonal offsetting they also do not exhibit the typical shearing profile. We therefore employ local upsampling and length adaptation to correct these problems.

Upsampling at binding. We locally refine the offset curve of the midline polyline at the node that connects to the binding by selecting several offset directions instead of just one. The offset directions are uniformly distributed between the binding direction and the adjacent polyline segment's normal direction. This results in higher resolution of the mesh in this area and provides a more accurate, rounded profile (see Fig. 9, bottom).

Segment length adaptation. For the page blocks and for each line of edges from the binding to the outer end of the active page, we sum the lengths of these edge chains. If this sum differs from the actual width of the page (Fig. 9, top), we translate the end vertices along the last segment direction to correct the total length. For the page blocks, this means that the top and the bottom of each block now has the same length, creating a sheared end that is typical in real-life books (Fig. 9, bottom). This process only affects the meshing part and not the simulation itself.

4. Digital book reading application

We employ our digital book modeling method to implement an interactive reader application that photorealistically visualizes the book and enables interactive page leafing and finger bookmarking.

4.1. User interaction implementation

Most of the time, the book is at rest while the user reads a page. For reading comfort, the book should display no distracting movements in this state, so only the page blocks are simulated and rendered in this situation. We instantiate an active page when the user mouse-clicks or touch-presses one of the page blocks to grab the next (or

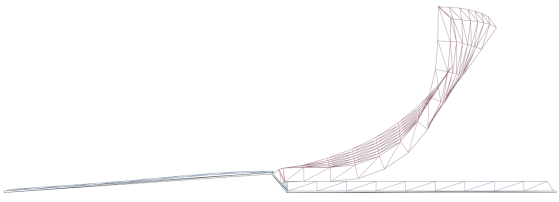


Figure 8: *The active page can represent a single book page, but also a stack of pages, in which case it serves as a "mid-layer" of the stack. In this example the stack consists of 45 pages out of an 80-page book.*

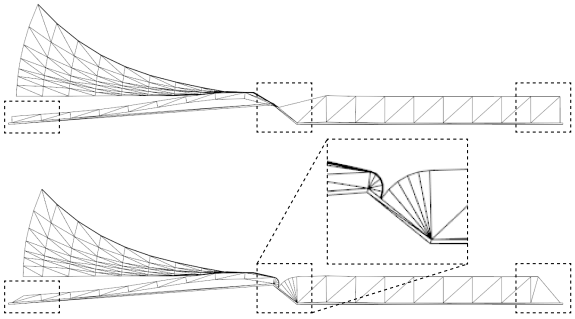


Figure 9: *Top: after the initial offsetting, the page blocks have an unnatural decreased thickness near the binding and a different surface area on the top side compared to the bottom side of the page block. Bottom: to correct these issues, we upsample the meshes at the binding to more accurately represent the high curvature, and adapt the lengths of the different page blocks and the active page mesh to produce the characteristic sheared appearance.*

Figure 10: *The user controls the active page by moving their finger on the screen. The page follows the thus-defined positional target while behaving like a real paper surface.*

the previous) page. The thickness of the page block meshes is automatically adapted to reflect the altered number of pages in each block. The user can bend the page and move it around, see Fig. 10; the required positional constraints are calculated as described in Sec. 3.3. When the page stops being actively moved, gravity brings it to rest on one of the page blocks; we then merge the page into the block, increasing the block's thickness.

Finger-bookmarking. Inspired by the finger-bookmarking from [KKL13], we let the user hold the current page while flipping through the next pages behind it. By saving the previously computed geometries of the active page, we can animate how pages are added and removed from the user's grip by instantiating single-page meshes, interpolated between those geometries (see Fig. 11 and the accompanying video). On a touch device, we use a simple swipe gesture of a second finger while still holding the middle page to initiate either an addition or a removal of a page, depending on whether the gesture was towards or away from the center. Every time we add or remove a page, we adapt the parameters for the number of pages held by the active page and the page blocks to adapt the mesh accordingly.

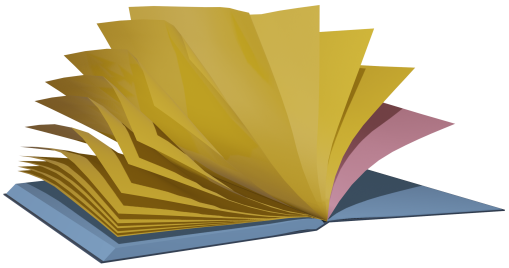


Figure 11: *We save a history of the active page geometries, strictly ordered, as shown in yellow. By interpolating between these geometries, we can animate how a user flips additional pages from the page block towards the red page or back. This lets users flip quickly through several pages at once. See the accompanying video for the full illustration of this feature.*

Figure 12: *Semi-translucent shader for the active page.*

4.2. Rendering

We employ a physically based rasterization shader using a pre-integrated environment light map and an additional point light as the light source. We do not consider light bounces and thus use shadow mapping to mimic shadows. For shading, we use a Cook-Torrance BRDF with an additional translucency term to extend it to a BTDF f (Fig. 12). We assume that part of the refracted light exits on the other side of the page and the rest is diffusely reflected.

$$f = k_d f_{\text{Lambert}} + k_s f_{\text{Cook-Torrance}} + k_t f_{\text{Lambert,back}}. \tag{15}$$

Let t be the thickness of the page and α a material parameter, then k_t is the amount of refracted light from the other side of the page:

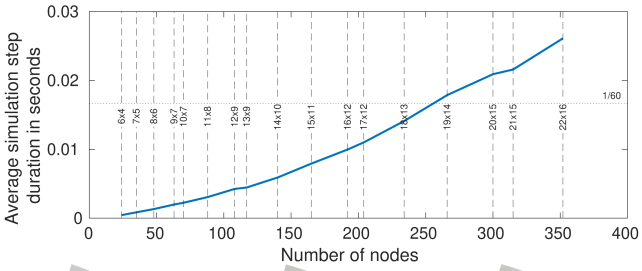
$$k_t = (1 - F_{\text{schlick,back}})\, e^{-\alpha t}. \tag{16}$$

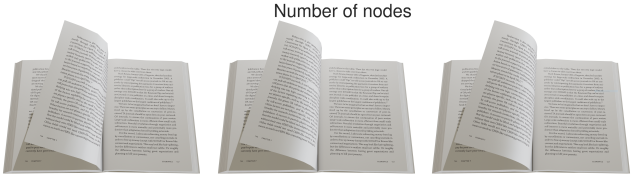
Here, $F_{\text{Schlick,back}}$ is Schlick's approximation to the Fresnel factor on the other side of the page.

We also remove the portion of the refracted light that gets transmitted through the page and thus is not diffusely reflected on the current side of the page:

$$k_d = (1 - F_{\text{Schlick}})\,(1 - e^{-\alpha t}). \tag{17}$$

Note that the light that does not get transmitted or diffusely reflected contributes to the specular reflection $k_s = F_{\text{Schlick}}$.





6×4 11×8 (ours) 22×16

Figure 13: *Performance impact and visual impact of different grid resolutions of the active page. Increasing the grid resolution beyond our chosen resolution of 11×8 results in longer simulation times while bringing little benefit to the visuals of our application.*

5. Experimental results

We implemented our system in C++ with OpenGL (ES) and libigl [JP*18]. We use Eigen3 [GJ*10] for the numerical computations and its SparseLU solver to solve the system in Sec. 3.1. We test our application on a laptop (i7 6600U, year 2015) natively and via WebAssembly. On iOS and Android, we test with an iPad Air (2014) and OnePlus 7T (2019). We provide an Android APK file in the additional material accompanying this paper. The application contains a full simulation of the book "Open Access" [Sub12].

Our application runs smoothly at interactive frame rates on the tested platforms. To gain performance, we exploit the fact that the sparsity pattern of our system matrix $DF(x,\lambda)$ stays constant, and therefore we can reuse the ordering permutation and symbolic factorization. We also notice that the constraint Hessian can be left out without any negative impact on the simulation and with a performance improvement of 7%, measured on a 750 frames benchmarking sequence with pre-recorded user inputs. Additionally, using only a single Newton iteration, as discussed in Sec. 3.1, reduces the average computation time per frame by 45% compared to two Newton iterations. In our experiments, the active page is a 11×8 quad grid. The page blocks are represented by 15 nodes in the midline. The grid resolution is chosen such that the active page looks sufficiently realistic while having a manageable performance impact, see Fig. 13. With these settings, the user can interact with the digital book in a life-like manner, flipping pages back and forth in a real-time simulation, as demonstrated in the accompanying video.

Our application is meant to be used as a digital book reader. To demonstrate the system, we use scans of the book "Il Cantone Ticino", printed in 1933 [Gal33], "Open Access", printed in 2012 and available digitally [Sub12], as well as an art catalog available digitally, see Fig. 14. We set the book properties, such as the bending and glue weights, as well as the page masses, based on empirical comparison with the printed books when available. Our default simulation parameters are $w_{\text{bend3D}} = 120$, $w_{\text{bend2D}} = 200$,

new print vintage book art catalog

Figure 14: *Various book types visualized with our method.*

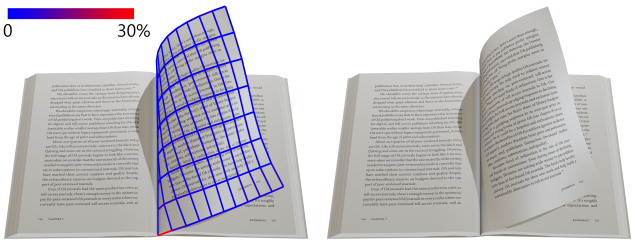
$w_{\text{glue}} = 600$, $w_{\text{iso3D}} = 4000$ and $w_{\text{pos}} = 400$. A single page has a unit mass, the width of a single page is set to 12 and the height is scaled according to the aspect ratio of the pages of the depicted book. The default single page thickness is set to 0.0075.

Since the active page uses a soft isometry objective (Sec. 3.3), the page area might vary during the simulation. When turning the active page, on average, the maximal stretch of an edge is 11% and the RMSE is 1.8%. However, even in extreme cases, as shown in Fig. 15, the stretch is not visible and the segment length adaptation of our meshing (Sec. 3.5) hides it completely in the final mesh.

In Fig. 16, we show the influence of the page bending and binding glue parameters on the shape of the digital book. A higher bending weight makes the book pages stiffer, while a higher binding glue term causes the pages to stay upright when the book is open and lying flat on a table.

5.1. Comparisons

We compare the page turning animations and the visual book representation to previous work in Fig. 17. In [LWB09], the authors opt to use a 2D book model with added shadows for an approximation of a 3D visualization effect. In the work of Chu et al. [CWLB03], the 3D page animation is pre-calculated from a mass-spring grid with a resolution of 18×16 and additional diagonal springs; this system is also used by Card et al. [CHMC04]. In contrast to this canned animation, our system enables real-time, interactive manipulation of the book pages and a photorealistic, fully three-dimensional book visualization.



Max stretch 30%, RMSE 3% same, without stretch visualization

Figure 15: *This book configuration is chosen to exhibit a deliberately high maximum stretch. The red edge at the bottom of the active page has a maximum stretch of 30%, while the RMSE over all edges is 3%. On the right, we show the same configuration without visualizing the isometry objective. The stretch does not impact the appearance of the page, as our meshing technique alleviates length disparities in the final mesh.*

Name	Effect	Commercial
flipbuilder.com	2D	yes
flipdocs.com	Static 3D	yes
fliphtml5.com	2D	yes
flippingbook.com	2D	yes
flipsnack.com	2D	yes
flipviewer.com	2D	yes
pageflip-books.com	2D, Static 3D	yes
pubhtml5.com	2D	yes
nextflipbook.com	2D	yes
turnjs.com	2D	no
yumpu.com	2D	yes

Table 1: *There are many page flip applications available online.* 2D effect *denotes a flip animation exhibiting only translation, rotation and cropping of a 2D texture on top of the book.* Static 3D effect *denotes a precomputed, non-interactive flip animation on a 3D mesh on top of the book. To the best of our knowledge, our method is the first to enable an interactive 3D simulation for book pages.*

In addition to academic works, there are many free and commercial systems that enable users to build a website or an application to visualize a book or magazine with page turning animations. To the best of our knowledge, all such software either uses a 2D effect similar to [LWB09], or a simple, precomputed, non-interactive 3D effect similar to [CWLB03]; see Table 1 for a summary of our findings. To the best of our knowledge, our method is the first to offer a fully interactive 3D page turning simulation.

To demonstrate the physical plausibility of our model, we compare our virtual book with its physical counterpart in Fig. 18 and the accompanying video. It can be observed that our digital book model reproduces the behavior of a real paper book with high fidelity; even the "popping" of a page from convex to concave shape is reproduced, as can be seen in the video recording in the additional material. We showcase the diversity of the animations produced by our interactive system in Fig. 19. Each page turning animation is different based on where the user grabs the page and which trajectory they take to go to the other side. It is even possible to grab a page, peek at the next one and then put the page down where it was, just like with a real book.

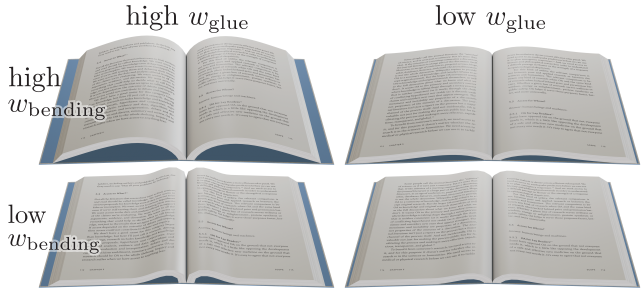


Figure 16: *By changing the weighting factors of w_{bend2D} and w_{glue}, we can simulate different book shapes. A higher bending weight creates stiffer pages, while a higher glue term keeps the pages more vertical at the binding.*

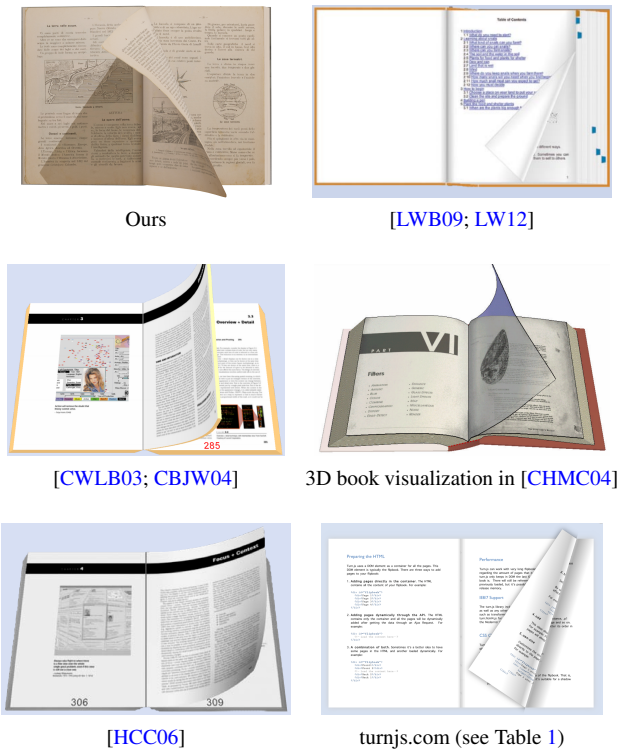


Figure 17: *Unlike other techniques and available applications, our method offers an immersive and interactive real-time manipulation of the book pages and a full 3D visualization of the book.*

6. Discussion, limitations and future work

On the technical side, our algorithm strongly relies on a physically faithful model for developable surfaces to provide a realistic simulation of paper sheets. The discrete orthogonal geodesic nets have proven suitable for this purpose, with the caveat that isometry is not inherently encoded in the model and has to be added as a soft constraint, i.e., a potential. The 4Q-isometry model offered in [RHS18a] is computationally much more expensive and appears to not be needed in practice. However, as mentioned, in the future it would be interesting to experiment with the method by Jiang et al. [JWR*20], which appeared concurrently to this work and provides a built-in discrete isometry modeling for developable surfaces.

Our digital book modeling covers many salient properties of physical books, but certainly not all. For example, we currently assume that the binding coincides with the book spine and is essentially a straight and rigid rectangle, and we model it with a single straight edge in our midline polyline. Many hardcover books use cloth binding that bends as pages are turned, and paperback spines are not entirely rigid and tend to deform and crease after extensive use. It is easily possible to extend our model by creating a binding *polyline* to mimic such book construction and even plastic effects

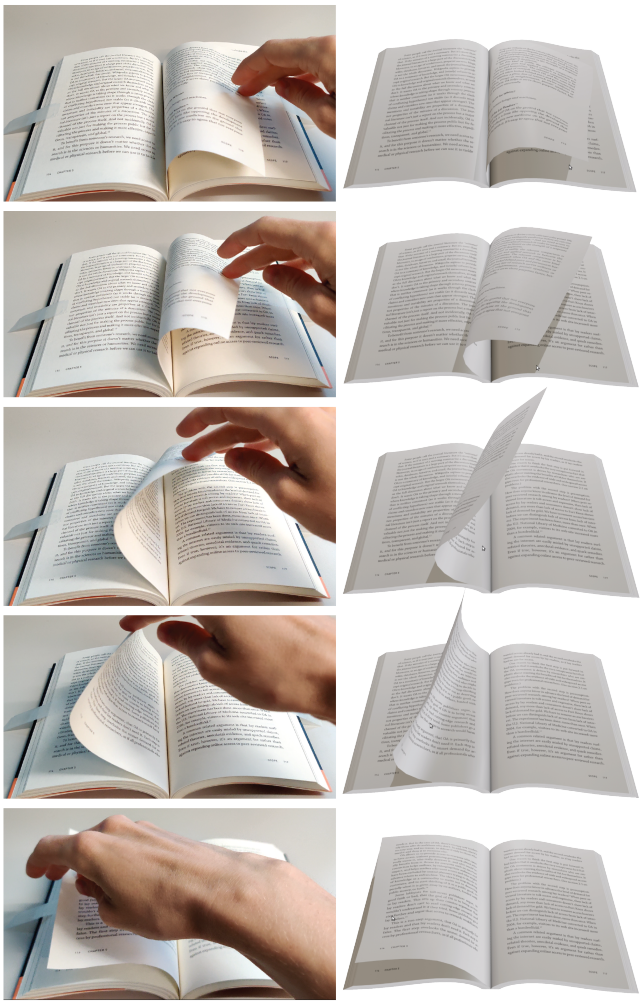


Figure 18: *We compare the printed book [Sub12] to its virtual counterpart while turning a page. The photographs are frames taken from a recorded video sequence only a few frames apart, and the virtual page images are from a single recorded sequence of a page flip in our system. The frames are chosen such that the page configurations roughly align.*

Figure 19: *To illustrate the diversity of the page turning animations of our method, each* column *shows every 5-th frame of a 60fps recording of a page flip from right to left. These different animations are produced by varying user interactions: grabbing the page in a different place, following a different trajectory from right to left.*

of paperbacks, if desired. Even more nuanced modeling is conceivable, such as varying the glue strength to imitate books made of multiple booklets bound together.

Our 3D book model looks overall convincing for modern books, but older books are often very degraded, with non-aligned pages and non-straight edges; the surface of their binding can exhibit shearing that we do not currently model. The aged appearance can be handled on the active page in our system by using transparency on the boundary of the page to show missing parts. However, the way we currently model the resting page blocks does not allow for realistic

rendering of the degradation (see inset). In the future, it is conceivable to employ more advanced meshing and rendering techniques to generate the blocks with irregular borders or bump maps to give the impression of a weathered book. Replacing the rendered page blocks with still images as background could also solve this issue, albeit at the cost of fixing the viewpoint.

In a similar vein, it would be interesting to extend the modeling of the developable surface of the page to include plastic effects such as wrinkling and creasing. The discrete surface model we employ can in fact accommodate creases, as discussed in [RHS18b; RHS19]; it is therefore conceivable to add an "earmarking" bookmarking feature to our method, but it would require adaptation to the dynamics simulation and the modeling of the page blocks. In general, it is worth investigating further to what degree the realism of the modeling and simulation of the virtual book contributes to reading experience and performance.

We have tested our application with standard mouse or touchscreen interfaces, which are currently the prevalent devices for digital reading. In the future, it would be interesting to extend our

method to immersive virtual reality environments with 3D interaction, possibly using a physical stand-in book for tactile feedback.

7. Conclusion

Reading a printed book is arguably a more immersive experience than modern e-readers. With our system, we aim to reproduce this experience in a virtual setting. Using recent advances in discrete developable surface modeling, we design a physically-based simulation that emulates the behavior of paper pages. Our application runs in real-time even on mobile devices thanks to the decomposition of our virtual book into a fully interactive 3D page and a 2D simulation controlling the rest of the book. We demonstrate that our virtual book offers a realistic visualization and user experience. It supports features such as "finger-bookmarking" that lets a reader flip quickly through several pages, and we imagine that it can be extended to support more advanced book navigation techniques.

Acknowledgments

We are grateful Aude Delerue and Patrick Roppel for the inspiration for this project and the insightful discussions. We also thank Michael Rabinovich for helpful discussions.

References

[Ale12] ALESSIO, QUAGLINO. "Membrane locking in discrete shell theories". PhD thesis. Niedersächsische Staats-und Universitätsbibliothek Göttingen, 2012 3.

[BW98] BARAFF, DAVID and WITKIN, ANDREW. "Large Steps in Cloth Simulation". *Proc. SIGGRAPH*. 1998, 43–54 3.

[CB98] CHAPELLE, DOMINIQUE and BATHE, KLAUS-JÜRGEN. "Fundamental considerations for the finite element analysis of shell structures". *Computers & Structures* 66.1 (1998), 19–36 3.

[CBJW04] CHU, YI-CHUN, BAINBRIDGE, DAVID, JONES, MATT, and WITTEN, IAN H. "Realistic Books: A Bizarre Homage to an Obsolete Medium?": *Proc. ACM/IEEE-CS Joint Conference on Digital Libraries*. 2004, 78–86 2, 10.

[CHMC04] CARD, STUART K., HONG, LICHAN, MACKINLAY, JOCK D., and CHI, ED H. "3Book: a scalable 3D virtual book". *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. 2004, 1095–1098 2, 9, 10.

[Cli19] CLINTON, VIRGINIA. "Reading from paper compared to screens: A systematic review and meta-analysis". *Journal of Research in Reading* 42.2 (2019), 288–325 1, 2.

[CWLB03] CHU, YI-CHUN, WITTEN, I. H., LOBB, R., and BAINBRIDGE, D. "How to turn the page [digital libraries]". *Joint Conference on Digital Libraries*. 2003, 186–188 2, 9, 10.

[Gal33] GALLI, ANTONIO. *Il Cantone Ticino: con brevi nozioni di geografia generale / testo-atlante ad uso delle scuole*. Lugano: A. Arnold, 1933, 77 p. : ill. 9.

[GHDS03] GRINSPUN, EITAN, HIRANI, ANIL N., DESBRUN, MATHIEU, and SCHRÖDER, PETER. "Discrete Shells". *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2003, 62–67 3.

[GHF*07] GOLDENTHAL, RONY, HARMON, DAVID, FATTAL, RAANAN, et al. "Efficient Simulation of Inextensible Cloth". *ACM Trans. Graph.* 26.3 (2007) 3, 4.

[GJ*10] GUENNEBAUD, GAËL, JACOB, BENOÎT, et al. *Eigen v3*. http://eigen.tuxfamily.org. 2010 9.

[HCC06] HONG, LICHAN, CARD, STUART K, and CHEN, JINDONG. "Turning pages of 3D electronic books". *3D User Interfaces (3DUI'06)*. IEEE. 2006, 159–165 2, 10.

[HRL17] HOU, JINGHUI, RASHID, JUSTIN, and LEE, KWAN MIN. "Cognitive map or medium materiality? Reading on paper and screen". *Computers in Human Behavior* 67 (2017), 84–94 1, 2.

[JP*18] JACOBSON, ALEC, PANOZZO, DANIELE, et al. *libigl: A simple C++ geometry processing library*. https://libigl.github.io/. 2018 9.

[JWR*20] JIANG, CAIGUI, WANG, CHENG, RIST, FLORIAN, et al. "Quad-Mesh Based Isometric Mappings and Developable Surfaces". *ACM Trans. Graph.* 39.4 (2020) 3, 10.

[KKL13] KIM, SANGTAE, KIM, JAEJEUNG, and LEE, SOOBIN. "BezelFlipper: Design of a Light-Weight Flipping Interface for e-Books". *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. 2013, 1719–1724 2, 8.

[KPH*13] KRETZSCHMAR, FRANZISKA, PLEIMLING, DOMINIQUE, HOSEMANN, JANA, et al. "Subjective Impressions Do Not Mirror Online Reading Effort: Concurrent EEG-Eyetracking Evidence from the Reading of Books and Digital Media". *PLOS ONE* 8.2 (Feb. 2013), 1–11 2.

[KT16] KOTAJIMA, YUTO and TANAKA, JIRO. "Book-Like Reader: Mirroring Book Design and Navigation in an E-Book Reader". *Human-Computer Interaction. Interaction Platforms and Techniques*. Ed. by KUROSU, MASAAKI. Springer International Publishing, 2016, 192–200 2.

[LW12] LIESAPUTRA, VERONICA and WITTEN, IAN H. "Realistic electronic books". *International Journal of Human-Computer Studies* 70.9 (2012), 588–610 2, 10.

[LWB09] LIESAPUTRA, V., WITTEN, I. H., and BAINBRIDGE, D. "Creating and Reading Realistic Electronic Books". *Computer* 42.2 (2009), 72–81 2, 9, 10.

[MOV19] MANGEN, ANNE, OLIVIER, GÉRARD, and VELAY, JEAN-LUC. "Comparing Comprehension of a Long Text Read in Print Book and on Kindle: Where in the Text and When in the Story?": *Frontiers in Psychology* 10 (2019), 38 1, 2.

[RHS18a] RABINOVICH, MICHAEL, HOFFMANN, TIM, and SORKINE-HORNUNG, OLGA. "Discrete Geodesic Nets for Modeling Developable Surfaces". *ACM Transactions on Graphics* 37.2 (2018) 1, 3–5, 10.

[RHS18b] RABINOVICH, MICHAEL, HOFFMANN, TIM, and SORKINE-HORNUNG, OLGA. "The Shape Space of Discrete Orthogonal Geodesic Nets". *ACM Trans. Graph.* 37.6 (2018) 5, 6, 11.

[RHS19] RABINOVICH, MICHAEL, HOFFMANN, TIM, and SORKINE-HORNUNG, OLGA. "Modeling Curved Folding with Freeform Deformations". *ACM Trans. Graph.* 38.6 (2019) 5, 11.

[SRH*15] SCHRECK, CAMILLE, ROHMER, DAMIEN, HAHMANN, STEFANIE, et al. "Non-smooth developable geometry for interactively animating paper crumpling". *ACM Trans. Graph.* 35.1 (Dec. 2015), 10:1–10:18 3.

[Sub12] SUBER, PETER. *Open Access*. The MIT Press, 2012. ISBN: 0262517639 9, 11.

[SVWG12] SOLOMON, JUSTIN, VOUGA, ETIENNE, WARDETZKY, MAX, and GRINSPUN, EITAN. "Flexible developable surfaces". 31.5 (2012), 1567–1576 3.

[TBWP16] TANG, CHENGCHENG, BO, PENGBO, WALLNER, JOHANNES, and POTTMANN, HELMUT. "Interactive Design of Developable Surfaces". *ACM Transactions on Graphics* 35 (2016), 1–12 3.