Animated Surfaces in Physically-based Simulation

Von der Fakultät für Informatik, Elektrotechnik und
Informationstechnik der Universität Stuttgart
zur Erlangung der Würde eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

Vorgelegt von

Markus Huber

aus Spaichingen

Hauptberichter:	Prof. Dr. Daniel Weiskopf
Mitberichter:	Prof. Dr. Bernhard Eberhardt
	Prof. Dr. Nils Thuerey
Tag der mündlichen Prüfung:	05.02.2018

Visualisierungsinstitut
der Universität Stuttgart

2018

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

BDF	backward differential formula
CFL	Courant–Friedrichs–Lewy
CG	conjugate gradient
CPU	central processing unit
CSF	continuum surface force
CUDA	compute unified device architecture
DFSPH	divergence-free Smoothed Particle Hydrodynamics
DOP	discrete oriented polytope
DTW	dynamic time warping
EOS	equation of state
FEM	finite element method
FV	feature vector
GPU	graphics processing unit
IIF	inter-particle interaction force
IISPH	implicit incompressible Smoothed Particle Hydrodynamics
IVP	initial value problem
MC	marching cubes
MLS	moving least squares

ODE	ordinary differential equation
PCISPH	predictive-corrective incompressible Smoothed Particle Hydrodynamics
PDE	partial differential equation
PPE	pressure Poisson equation
RAM	random-access memory
SESPH	state equation Smoothed Particle Hydrodynamics
SPH	Smoothed Particle Hydrodynamics
VRAM	video random access memory

Units

fps	frames per second
GHz	Gigahertz
GB	Gigabyte

List of Symbols

Geometry, Functions, and Matrices

Symbol	**Unit**	**Explanation**
$\mathbf{x}$	m	point in $\mathbb{R}^3$ with $\mathbf{x} = (x,y,z)^T$
$\mathbf{I}$		identity matrix
$\delta(.)$		Dirac (delta) function
h	m	kernel smoothing length
$W(r,h)$		kernel function with distance r and support radius h
S	m^2	surface
H	deg	mean curvature

Operators

Symbol	**Explanation**
$\nabla\cdot$	divergence: $\nabla\cdot = \left(\frac{\partial}{\partial x_1} + ... + \frac{\partial}{\partial x_n}\right)$
∇	gradient: $\nabla = \left(\frac{\partial}{\partial x_1}, ..., \frac{\partial}{\partial x_n}\right)^T$
∇^2	Laplacian: $\nabla^2 = \nabla \cdot \nabla = \left(\frac{\partial^2}{\partial x_1^2} + ... + \frac{\partial^2}{\partial x_n^2}\right)$
$\frac{D}{Dt}$	substanial derivative: $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla$
$\|\|.\|\|$	Euclidean norm
$\dot{\mathbf{x}}$	first time derivative $\dot{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial t}$
$\ddot{\mathbf{x}}$	second time derivative $\dot{\mathbf{x}} = \frac{\partial^2 \mathbf{x}}{\partial^2 t}$

Physics

Symbol	Unit	Explanation
$\mathbf{F}$	N	force
m	kg	mass
V	m^3	volume
$\mathbf{a}$	$\mathrm{m\,s^{-2}}$	acceleration
$\mathbf{v}$	$\mathrm{m\,s^{-1}}$	velocity
$\mathbf{g}$	$\mathrm{m\,s^{-2}}$	gravitational acceleration
t	s	time
Δt	s	time step
$\mathbf{J}$	$\mathrm{N\,s}$	impulse

Fluid Physics

Symbol	Unit	Explanation
$\mathscr{V}$	m^3	control volume
$m_{\mathscr{V}}$	kg	mass of control volume
$\mathbf{v}$	$\mathrm{m\,s^{-1}}$	velocity of a fluid element with $\mathbf{v} = (u,v,w)^T$
$\mathbf{n}$	m	normal on surface of control volume
ρ	$\mathrm{kg\,m^{-3}}$	mass density
ρ_0	$\mathrm{kg\,m^{-3}}$	rest density
$\mathbf{f}$	$\mathrm{N\,m^{-3}}$	force per unit volume
$\mathbf{f}^b$	$\mathrm{N\,m^{-3}}$	body force
$\mathbf{f}^s$	$\mathrm{N\,m^{-2}}$	surface force
$\boldsymbol{\sigma}$	$\mathrm{N\,m^{-2}}$	stress tensor
σ	$\mathrm{N\,m^{-2}}$	normal stress
τ	$\mathrm{N\,m^{-2}}$	shear stress
p	$\mathrm{N\,m^{-2}}$	pressure
$\mathbf{T}$	$\mathrm{N\,m^{-2}}$	deviatoric stress tensor
μ	$\mathrm{kg\,m^{-1}\,s^{-1}}$	dynamic viscosity
ν	$\mathrm{m^2\,s^{-1}}$	kinematic viscosity
κ	$\mathrm{m^2\,s^{-2}}$	(pressure) stiffness constant
γ	1	pressure constant

Particles

Symbol	Unit	Explanation
$\mathbf{x}_i$	m	position in world coordinates of particle i

Symbol	Unit	Explanation
$\mathbf{v}_i$	$\mathrm{m\,s^{-1}}$	velocity of particle i
$\mathbf{v}_i^*$	$\mathrm{m\,s^{-1}}$	intermediate velocity of particle i
V_i	$\mathrm{m^3}$	volume of particle i
m_i	kg	mass of particle i
ρ_i	$\mathrm{kg\,m^{-3}}$	density of particle i
ρ_i^*	$\mathrm{kg\,m^{-3}}$	intermediate density of particle i
p_i	$\mathrm{N\,m^{-2}}$	pressure of particle i
$\mathbf{F}_i$	N	net force acting on particle i
$\mathbf{F}_i^b$	N	body force acting on particle i
$\mathbf{F}_i^p$	N	pressure force acting on particle i
$\mathbf{F}_i^v$	N	viscosity force acting on particle i
$\mathbf{F}_i^{st}$	N	surface tension force acting on particle i
φ	$\mathrm{s^{-2}}$	surface tension coefficient for the IIF-based model
γ	$\mathrm{kg^{-1}\,s^{-2}}$	surface tension coefficient for the combined model
κ	$\mathrm{kg\,m^{-5}\,m^{-2}}$	surface tension coefficient for the CSF model
$C(r,h)$	1	kernel function for repulsions in surface tension for a distance r and support radius h
$\mathbf{n}_i$	m	normal of particle i
c_i		color field at particle i

Deformable Bodies and Cloth

Symbol	Unit	Explanation
$\mathcal{B}$		deformable body
V	$\mathrm{m^3}$	volume of deformable body
A	$\mathrm{m^2}$	boundary area of deformable body
$\mathbf{x}_i$	m	position of material particle i of deformable body in $\mathbb{R}^3$
$\bar{\mathbf{x}}_i$	m	rest position of material particle i of deformable body in $\mathbb{R}^3$
$\boldsymbol{\varphi}(\mathbf{x})$	1	configuration mapping
$\mathbf{F}$	1	deformation gradient tensor
$\mathbf{E}$	1	Green strain tensor
$\mathbf{C}$	1	right Cauchy-Green deformation tensor
$\mathbf{U}$	1	pure stretch tensor
$\mathbf{R}$	1	rotation tensor
$\mathbf{u}$	m	displacement field
ε_C	1	Cauchy strain

Symbol	Unit	Explanation
ε_{CR}	1	co-rotational strain
$\mathbf{f}^b$	N	body force
$\mathbf{t}$	N	traction force
$\mathbf{f}$	N	net force
$\mathbf{f}^e$	N	elastic force
$\mathbf{f}^v$	N	viscosity force
$\boldsymbol{\sigma}$	$\mathrm{N\,m^{-2}}$	Cauchy stress tensor
$\boldsymbol{\mathcal{C}}$	1	elasticity tensor
E	$\mathrm{N\,m^{-2}}$	Young's modulus
G	$\mathrm{N\,m^{-2}}$	Shear modulus
ν	1	Poisson's ratio
u		cloth material direction: weft
v		cloth material direction: warp

Discretized Cloth

Symbol	Unit	Explanation
e		(triangular) element
$\mathbf{u}_i$	m	displacement of node i
$\mathbf{M}$	kg	diagonal mass matrix
$\mathbf{K}$	$\mathrm{N\,m^{-1}}$	stiffness matrix
$\mathbf{D}$	$\mathrm{kg\,s^{-1}}$	damping matrix
k^B	$\mathrm{N\,m^{-1}}$	bending stiffness coefficient
τ	m	cloth thickness
N_i	1	FEM shape function for node i
A_e	$\mathrm{m^2}$	area of (triangular) element e
θ	deg	bending angle of two neigboring elements
$\mathbf{f}_i^{B_l}$	N	linear bending force on node i
$\mathbf{f}_i^{B_n}$	N	nonlinear bending force on node i
$\mathbf{R}^{B_l}$	m	linear bending vector
b		bending element (two adjacent elements sharing an edge)
θ	deg	angle between normals of two adjacent mesh elements
$\mathbf{F}_N$	N	cloth-fluid contact normal force
$\mathbf{F}_F$	N	cloth-fluid contact friction force
μ	1	friction control parameter
δ	$\mathrm{kg\,s^{-1}}$	friction control parameter

Cloth Retrieval

Symbol	**Explanation**
$\mathcal{S}$	simulation
$\mathcal{S}'$	simulation segment
ς_i	simulation state *i*
$d(.)$	distance function
$\mathbf{A}$	feature vector of a cloth animation
a_i^t	attribute *i* of cloth surface at time *t*
A	set of attributes
w	attribute weight

Acknowledgements

First of all, I would like to thank my advisors Daniel Weiskopf and Bernd Eberhardt for their ongoing support and their invaluable advice during my doctoral studies. By sharing their knowledge and experience in numerous fruitful discussions, they guided me toward my goal. It was a great pleasure to be their PhD student, I had a great time.

I would also like thank Nils Thuerey for reviewing my thesis. Furthermore, I am grateful to the Promotionskolleg Digital Media for funding parts of my research.

Further, I would like to thank all my collaborators and co-authors for their contributions to my research: Otilia Dumitrescu, Thomas Ertl, Manuel Hirschler, Daniel Kauker, Michael Krone, Ulrich Nieken, Guido Reina, and Stefan Reinhardt.

I am very thankful to the colleagues and friends at VISUS for a great time. In particular, I greatly enjoyed the numerous coffee breaks with Julian Heinrich, Corinna Vehlow, Marco Ament, Fabian Beck, and Filip Sadlo. I also had a great time with my office mates Marco Ament, Michael Bußler, Grzegorz Karch, Haris Mumtaz, Filip Sadlo, and Liang Zhou. Furthermore, I enjoyed the time with my colleagues of the Promotionskolleg Dieter Morgenroth, Stefan Reinhardt, Benjamin Wollet, Sebastian Herholz, Jan Fröhlich, Lena Gieseke, and David Körner. Many thanks go to Stefan Reinhardt and Sebastian Boblest for proofreading parts of my thesis.

I want to express my deepest gratitude to my family for their never-ending support. I also want to thank all my friends, in particular Martin and Julian, for always being there and cheering me up. Finally, would like to thank Jana for her understanding and support. Thank you for going through mostly nice and sometimes rough periods with me.

Abstract

Physics-based animation has become a ubiquitous element in all application areas of computer animation, especially in the entertainment sector. Animation and feature films, video games, and advertisement contain visual effects using physically-based simulation that blend in seamlessly with animated or live-action productions. When simulating deformable materials and fluids, especially liquids, objects are usually represented by animated surfaces. The visual quality of these surfaces not only depends on the actual properties of the surface itself but also on its generation and relation to the underlying simulation. This thesis focuses on surfaces of cloth simulations and fluid simulations based on Smoothed Particle Hydrodynamics (SPH), and contributes to improving the creation of animations by specifying surface shapes, modeling contact of surfaces, and evaluating surface effects of fluids.

In many applications, there is a reference given for a surface animation in terms of its shape. Matching a given reference with a simulation is a challenging task and similarity is often determined by visual inspection. The first part of this thesis presents a signature for cloth animations that captures characteristic shapes and their temporal evolution. It combines geometric features with physical properties to represent accurately the typical deformation behavior. The signature enables calculating similarities between animations and is applied to retrieve cloth animations from collections by example.

Interactions between particle-based fluids and deformable objects are usually modeled by sampling the deformable objects with particles. When interacting with cloth, however, this would require resampling the surface at large planar deformations and the thickness of cloth would be bound to the particle size. This problem is addressed in this thesis by presenting a two-way coupling technique for cloth and fluids based on the simulation mesh of the textile. It allows robust contact handling and intuitive control of boundary conditions. Further, a solution for intersection-free fluid surface reconstruction at contact with thin flexible objects is presented.

The visual quality of particle-based fluid animation highly depends on the properties of the reconstructed surface. An important aspect of the reconstruction method is that it accurately represents the underlying simulation.

This thesis presents an evaluation of surfaces at interfaces of SPH simulations incorporating the connection to the simulation model. A typical approach in computer graphics is compared to surface reconstruction used in material sciences.

The behavior of free surfaces in fluid animations is highly influenced by surface tension. This thesis presents an evaluation of three types of surface tension models in combination with different pressure force models for SPH to identify the individual characteristics of these models. Systematic tests using a set of benchmark scenes are performed to reveal strengths and weaknesses, and possible areas of applications.

German Abstract —Zusammenfassung—

Physikalisch basierte Animationen sind ein allgegenwärtiger Teil in jeglichen Anwendungsbereichen der Computeranimation, insbesondere dem Unterhaltungssektor. Animations- und Spielfilme, Videospiele und Werbung enthalten visuelle Effekte unter Verwendung von physikalisch basierter Simulation, die sich nahtlos in Animations- oder Realfilme einfügen. Bei der Simulation von deformierbaren Materialien und Fluiden, insbesondere Flüssigkeiten, werden die Objekte gewöhnlich durch animierte Oberflächen dargestellt. Die visuelle Qualität dieser Oberflächen hängt nicht nur von den Eigenschaften der Fläche selbst ab, sondern auch von deren Erstellung und der Verbindung zu der zugrundeliegenden Simulation. Diese Dissertation widmet sich Oberflächen von Textil- und Fluidsimulationen mit der Methode der Smoothed Particle Hydrodynamics (SPH) und leistet einen Beitrag zur Verbesserung der Erstellung von Animationen durch die Beschreibung von Oberflächenformen, der Modellierung von Kontakt von Oberflächen und der Evaluierung von Oberflächeneffekten von Fluiden.

In vielen Anwendungen gibt es eine Referenz für eine Oberflächenanimation, die ihre Form beschreibt. Das Abgleichen einer Referenz mit einer Simulation ist eine große Herausforderung und die Ähnlichkeit wird häufig visuell überprüft. Im ersten Teil der Dissertation wird eine Signatur für Textilanimationen vorgestellt, die charakteristische Formen und ihre zeitliche Veränderung erfasst. Sie ist eine Kombination aus geometrischen Merkmalen und physikalischen Eigenschaften, um das typische Deformationsverhalten genau zu repräsentieren. Die Signatur erlaubt es, Ähnlichkeiten zwischen Animationen zu berechnen, und wird angewendet, um Textilanimationen aus Kollektionen anhand eines Beispiels aufzufinden.

Interaktionen zwischen partikelbasierten Fluiden und deformierbaren Objekten werden gewöhnlich durch das Abtasten des deformierbaren Objekts mit Partikeln modelliert. Bei der Interaktion mit Textilien würde dies jedoch ein neues Abtasten bei großen planaren Deformation erfordern und die Stärke des Textils wäre an die Partikelgröße gebunden. Mit diesem Problem befasst sich diese Dissertation und stellt eine Technik für die wechselseitige Kopplung zwischen Textilien und Fluiden vor, die auf dem Simulationsnetz des Textils beruht. Diese erlaubt eine robuste Kontaktbehandlung und intuitive Kontrolle von Randbedingungen. Des Weiteren wird ein Lösungsansatz für eine durchdringungsfreie Oberflächenrekonstruktion beim Kontakt mit dünnen flexiblen Objekten präsentiert.

Die visuelle Qualität von partikelbasierten Fluidanimationen hängt stark von den Eigenschaften der rekonstruierten Oberfläche ab. Wichtig bei Rekonstruktionsmethoden ist, dass sie die zugrundeliegende Simulation genau repräsentieren. Die Dissertation präsentiert eine Evaluierung von Oberflächen an Grenzflächen, die den Zusammenhang zum Simulationsmodell miteinbezieht. Ein typischer Ansatz aus der Computergrafik wird mit der Oberflächenrekonstruktion in der Werkstoffkunde verglichen.

Das Verhalten von freien Oberflächen in Fluidanimationen wird stark von der Oberflächenspannung beeinflusst. In dieser Dissertation wird eine Evaluierung von drei Oberflächenspannungsmodellen in Kombination mit verschiedenen Druckmodellen für SPH präsentiert, um die Charakteristika der jeweiligen Modelle zu identifizieren. Es werden systematische Tests mit Hilfe von Benchmark-Tests durchgeführt, um Stärken, Schwächen und mögliche Anwendungsbereiche deutlich zu machen.

CHAPTER 1

Introduction

Reproducing physical phenomena is one of the most exciting and challenging disciplines in computer graphics. Especially physically-based simulation techniques are a major field in computer graphics research and gained much interest in many areas of application. One of the most important and prominent fields that makes use of physics-based animation is in the production of visual effects, for instance in the context of feature films or computer games. With the present techniques, it is possible to create animations of physical effects of almost arbitrary complexity that go beyond the capabilities of traditional key frame or procedural animation techniques. In many applications, visual quality has already reached a level of realism, where animations are indistinguishable from physical reality. Typical examples are the destruction of buildings and structures with rigid body simulations, floodings and explosions with fluid simulation, cloth or hair animations with elastic object simulations, and the interaction between objects being simulated by the different simulation techniques.

The fundamental process of creating animated surfaces with physically-based simulation techniques is outlined in Figure 1.1. Beginning with an observed physical phenomenon, a mathematical model that governs the phenomenon is established, usually the underlying physical laws. The mathematical model is typically given by a set of ordinary differential equations (ODEs) or partial differential equations (PDEs). Solving these governing equations numerically requires a suitable discrete representation of the domain, such as a grid or particles, that forms the basis for a numerical solution. The numerical approximation commonly involves a spatial and temporal discretization of the governing equations in continuous forms to discrete forms and allows for

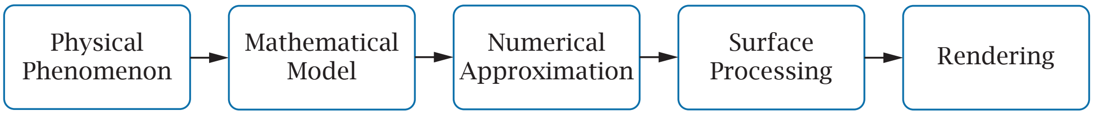


Figure 1.1: The process of creating an animation based on surfaces using physically-based simulation.

a computational solution. This component of the process usually involves iterative updates for a given time frame. Depending on the discretization of the domain, a surface is obtained either directly by the representation or has to be converted from an implicit representation. Further processing of the surface is optional and rendering of selected frames can be performed for a final animation given by a sequence of images.

Many research areas from scientific computing such as applied mathematics, numerical analysis, and computational mechanics provide essential techniques for physically-based simulations and, thereby, make it possible to reach a high level of accuracy. However, for computer graphics purposes, transferring those techniques is often not sufficient as there are special requirements and challenges. Making the simulation methods robust in terms of numerical computation and external manipulation is essential as well as computational efficiency. Creating high-quality animations within several hours of computation on standard workstations and real-time techniques for computer games are of major interest. For visual quality, accuracy is not necessarily the most suitable measure, as the most important aspect is visual quality in terms of the perception of the consumer. Another important topic from a computer graphics perspective is extensibility and manipulation of the physically-based simulations. For some applications, visual realism is the main goal, but often, artistic freedom or meeting a predefined objective is more important for the resulting animation. For animators, it should be possible to use the tools to bend or even sacrifice physical correctness to create goal-oriented animations or enrich them with artistic features.

The topic of thesis is in the context of animated surfaces based on physically-based simulations with focus on cloth and fluid animations. For computer graphics purposes, surfaces provide a good basis, especially since they are convenient for further processing and well-suited for rendering. With solid bodies, the surface is usually given explicitly and can be utilized directly for subsequent tasks. In contrast, fluid surfaces are usually given implicitly, and they are either converted into explicit representations or processed based on their implicit functions. This thesis addresses open challenges in the field of animated surfaces resulting from the aforementioned simulations. In particular, the improvement of surface representations of the simulated objects is of major interest in terms of whether the surface meets a given reference, whether interfaces between simulations are handled in a suitable manner, and how simulation models influence the resulting shapes of the surface.

In the following, important terms and concepts in the context of animated surfaces are briefly discussed. Subsequently, the research challenges and contributions of this thesis are presented in Section 1.2 and Section 1.3.

1.1 ANIMATED SURFACES

Surface representations can be classified in two major categories [29]: parametric and implicit representations. Given a 2D parameter domain $\Omega \subset \mathbb{R}^2$, parametric surfaces S_P are defined by

$$\boldsymbol{\mathcal{F}} : \Omega \rightarrow S_P, \tag{1.1}$$

$$S_P = \boldsymbol{\mathcal{F}}(\Omega) \subset \mathbb{R}^3, \tag{1.2}$$

where $\boldsymbol{\mathcal{F}}$ maps the parameter domain to the surface S_P. In contrast, an implicit surface S_I is defined as the zero set of a scalar function $\mathcal{F}$ and can be expressed as

$$\mathcal{F} : \mathbb{R}^3 \rightarrow \mathbb{R}, \tag{1.3}$$

$$S_I = \{\mathbf{x} \in \mathbb{R}^3 | \mathcal{F}(\mathbf{x}) = 0\}. \tag{1.4}$$

For arbitrary shapes, it is usually more convenient to partition the domain Ω into small segments and use a piecewise definition that approximates the surface. Depending on the size and number of segments, the quality of the approximation can be controlled. In computer graphics, triangles and quadrilaterals are the most common types of segments for parametric surface definitions. For implicit surfaces, the domain is typically subdivided by means of hexahedra or tetrahedra.

Both representations exhibit properties that are advantageous or disadvantageous for the use of animated surfaces. Deforming and animating parametric surfaces is easily obtained by displacement of the surface points, which is easy to control. Large deformations, however, can lead to undesired distortions and could require time-consuming remeshing. Also, changes in topology and (self-) intersections need to be taken care of. Implicit surfaces are not bound to a fixed topology and, therefore, remain consistent. On the downside, manipulation and tracing the surface require more effort.

For cloth simulations, an explicit surface representation is already given by the finite element discretization, usually in the form of a regular or irregular triangle mesh. The number of nodes in the mesh and connectivity usually do not change during simulation and animation is given by changes in position of the nodes. Large deformations of individual elements (triangles) do not occur since typical textiles show only small stretch deformations and (self-) collision handling avoids intersections. To capture important details of textiles, mesh resolution is usually sufficient and quality can be further increased by using subdivision schemes such as Loop subdivision [70].

In case of fluids, the surface is given implicitly by an isosurface of either a distance field or a density field. The implicit surface can be rendered directly by means of raycasting or be converted into an explicit mesh. Conversion is usually done using the marching cubes (MC) algorithm [71] or one of its variants. The result is a triangle mesh for each frame of the animation. Quality is mainly given by the resolution of the MC grid that has to be chosen by a trade-off between mesh resolution and computational efficiency. The implicit representation is particularly suitable for liquids as topology changes such as splits and merges of droplets are handled naturally.

1.2 RESEARCH CHALLENGES

Despite much progress in physics-based simulation methods to recreate real-world phenomena, the research field is still of high importance. There are many challenges, especially in the creation of animated surfaces that conform to a reference and to gain knowledge of simulation results. Describing the evolving shape of cloth simulations, modeling the interaction between fluids and deformable surfaces, and improving the understanding of simulation results are addressed in this thesis. The specific challenges of these topics related to animated surfaces are given below.

Given a simulation system, the straightforward approach to create an animation is to setup the initial state, in particular defining the geometry or domain of the dynamic object, setting initial positions of all objects in the scene, and choosing a set of parameters for the simulation models and the material. Then, the simulation is started to produce a sequence of simulation states. Based on the defined parameters, the user has a rough idea how the result will look like. However, even experienced animators cannot predict the outcome of the simulation in detail, and if it corresponds to their expectations. Especially for cloth simulations, the characteristic shape of the underlying mesh of the textile and its temporal evolution are of major interest for the resulting animation. It is not always possible to draw a link between the simulation settings and the shape of resulting animated surface, which has a big impact on the final visual quality of the animation. To this end, a descriptor for the shape of cloth in a simulation that captures the changes during animation is required to perform an analysis of the simulation result. This descriptor enables a variety of applications, such as measuring how good an animation is in agreement with a given reference.

The interaction between dynamic objects is a common scenario and requires dedicated models for boundary handling to achieve natural interaction. Coupling particle-based fluids with cloth simulations is especially challenging as cloth is modeled as a thin surface with a wide range of possible material properties. At first, penetration of the highly deformable cloth surface by the fluid has to be prevented. Depending on the fabrics' material that ranges from impermeable to porous, the behavior at the contact area can be very different. Considering the special characteristics of cloth, boundary handling that ensures intersection-free coupling and allows modeling varying cloth thickness and different material properties at contact, is required. An additional challenge is the surface representation of fluids at the interaction area with cloth. The implicitly defined surface has to be intersection-free with the cloth mesh and the distance between both surfaces is important for ray tracing-based rendering of transparent liquids such as water.

When reconstructing fluid surface meshes from their implicit representations, surface quality can be viewed from different points. For pure visual quality, smoothness and the capability to preserve detailed features are certainly of major interest. However, the accuracy how well the implicit surface represents the simulated fluid is an important basis for the visual quality and is rarely considered. Especially for free surfaces, ensuring an accurate reconstruction of the surface with respect to the underlying fluid will help achieve the goal of increased perceived realism. A quantitative analysis of the final surface at interfaces and comparison to simulation-bound surface reconstruction techniques can provide valuable insights for visual quality.

In the development of simulation models, researchers normally set up a set of test cases to show the performance of their method. If there are previous approaches, they do a comparison using these tests. However, the set of tests is built from scratch and there are no unified benchmarks. Further, quantitative analysis and comparison are often not performed, which makes it difficult to judge if the proposed method is suitable for a given production scenario or as a basis for future research. An example of this situation are the different surface tension models for particle-based fluids. Several methods exist based on different underlying models, and it is difficult to estimate the specific properties of the respective methods. In particular, to evaluate these properties and compare the different approaches, there are no benchmark tests publicly available.

The goal of this thesis is to extend and improve the possibilities of creating animated surfaces with physically-based simulation in various ways. The creation of animations with a given target is an interesting topic and is addressed for cloth simulations. In addition, offering the user to influence the simulation models at contact of cloth and fluid surfaces is given attention. Further, evaluating existing methods that directly influence implicit surfaces to improve the understanding of their properties and, therefore, facilitate their application, is of particular interest.

1.3 OUTLINE AND CONTRIBUTIONS

This section provides a short abstract for each chapter and the author's contributions to the related publications. Chapter 2 summarizes fundamental principles and computational concepts of physics-based animation of fluids and cloth. It is based on previous work, and therefore, is not part of the technical contribution of this thesis. The publications of Chapters 3, 4, and 6 were co-authored with both of the author's advisors, Bernhard Eberhardt and Daniel Weiskopf. The publication of the contributions in Chapter 5 was co-authored with Daniel Weiskopf.

Chapter 3 presents a descriptor to capture the time-dependent shape of cloth surfaces and to describe the dynamic behavior of cloth animations. This *motion-shape signature* given by a feature vector includes geometrical features and physical properties of cloth, and therefore, encodes the changing shape of cloth over time. It enables the description of the specific characteristics of cloth witnessed in physics-based animations. With the signature, it is possible to calculate similarities of cloth animations and it is shown how to retrieve cloth animations from collections of simulations that feature the same characteristics as the input. As a result, an animation that exhibits similar appearance in terms of geometrical features and a parameter set that represents the properties of the input animation can be found. With this technique, the creation of goal-oriented animations can be facilitated, e.g., by matching animations to given references or reproducing animations after changes in initial conditions were made. The contribution of this work was published in a journal paper in IEEE Computer Graphics and Applications [3].

Chapter 4 introduces a robust and efficient method for the two-way coupling between particle-based fluid simulations and infinitesimally thin surfaces represented by triangular meshes. The interface handling is based on a hybrid method that combines a repulsion force approach with a continuous intersection handling to prevent penetration. Boundary conditions for the tangential component of the fluid's velocity are implemented to model the no-slip boundary condition. To account for the varying material properties of different fabrics, the model is extended to allow boundary conditions that exceed the standard slip or no-slip conditions. The method presented in this chapter is able to support textiles that can absorb fluid, which influences the boundary condition. With the proposed technique, elastic surfaces do not have to be sampled with particles and the interaction is handled separately from the respective simulation systems. In addition, it is shown how standard fluid surface reconstruction algorithms can be modified to prevent the extracted surface from intersecting surfaces of close objects. Results are shown for the bidirectional interaction between liquid simulations based on SPH and state-of-the-art mesh-based cloth simulation systems. The contributions of this chapter were published in a conference paper at the International Workshop on Vision, Modeling and Visualization [1] that received a Best Paper Award and in an extended journal paper in Computer Graphics Forum [2].

Chapter 5 presents an evaluation of surface reconstruction techniques in the context of multiphase SPH simulations. For a given simulation, the evolution of the area of reconstructed surfaces is measured and compared to an analytic power law for the given simulation. A quantitative analysis of a well-known computer graphics approach for calculating the surface of a simulation is performed, also in comparison to a model that is tailored to accurately represent the underlying simulation. The contributions of this chapter were published in a short paper at the Eurographics Conference on Visualization [7], co-authored with Michael Krone, Katrin Scharnowski, and Daniel Kauker, who provided the dedicated surface reconstruction method for the simulation method, and Manuel Hirschler, who provided the simulation data. Guido Reina, Ulrich Nieken, and Thomas Ertl helped writing the paper. The author of this thesis provided the surface reconstruction method and, together with Michael Krone, performed the evaluation.

In Chapter 6, an evaluation of surface tension models in particle-based fluid simulation systems using SPH is presented. Although visual quality is of major interest in computer graphics applications and is considered as well, quantification methods for the properties of these models using a benchmark test are proposed. The benchmark consists of three experiments and a set of analysis methods that allow for the comparison of surface tension models. The goal of the evaluation is to identify if a certain model is suitable for a given scenario and, by revealing the individual properties of a models, to support controlling results in the creation of animations. Three existing surface tension models from different classes in combination with different SPH techniques, state equation Smoothed Particle Hydrodynamics (SESPH), predictive-corrective incompressible Smoothed Particle Hydrodynamics (PCISPH), and implicit incompressible Smoothed Particle Hydrodynamics (IISPH), are evaluated and systematically tested to show the influence of different settings and parameter choices. The surface tension models are chosen from a pure inter-particle force model, a model based on surface curvature, and a model using a combination of these. The results of the evaluation are presented and an elaborate discussion of the findings is given. The contributions of this chapter were published in a paper at the Workshop on Virtual Reality Interaction and Physical Simulation [6]. The paper was co-authored with Stefan Reinhardt, who provided most of the implementations of the SPH and surface tension models and contributed a modification to the inter-particle force model to improve its visual results. The author of this dissertation developed the design of the benchmarks and performed the quantitative analysis of the results of the tests.

During the author's research period, further co-authored papers were published with the author's advisors that are related to this dissertation but are not part of its core topic. Together with Stefan Reinhardt, Otilia Dumitrescu, and Michael Krone, a visual debugging environment for SPH-based fluid simulations has been developed and was published in a conference paper at the International Conference on Information Visualization [8]. In addition, together with Stefan Reinhardt, a paper that introduces an asynchronous time integration scheme for SPH simulations was published at the ACM SIGGRAPH/Eurographics Symposium on Computer Animation [9].

Previous to the research period at the University of Stuttgart, co-authored work presenting the simulation of wet cloth was published [4], [5] and is is used in Chapter 4.

1.4 REUSED AND COPYRIGHTED MATERIAL

In this thesis, material from the following papers is partly reused where copyright is owned by the author according to the respective EUROGRAPHICS Exclusive License Forms:

- [1] ©2014. M. Huber, B.Eberhardt, and D. Weiskopf, Cloth-Fluid Contact, Vision, Modeling & Visualization, pp. 81–88, 2013.
- [7] ©2014. M. Krone, M. Huber, K. Scharnowski, M. Hirschler, D. Kauker, G. Reina, U. Nieken, D. Weiskopf, and T. Ertl, Evaluation of Visualizations for Interface Analysis of SPH, EuroVis - Short Papers, pp. 109–113, 2014.
- [6] ©2015. M. Huber, S. Reinhardt, D. Weiskopf, and B. Eberhardt, Evaluation of Surface Tension Models for SPH-Based Fluid Animations Using a Benchmark Test, Workshop on Virtual Reality Interaction and Physical Simulation, pp. 41–50, 2015.

In addition, material from a copyrighted paper is partly reused with the kind permission of John Wiley and Sons following the license agreement for Dissertation/Thesis use (License Number 4243210052431):

- [2] ©2014 John Wiley and Sons. M. Huber, B. Eberhardt, and D. Weiskopf, Boundary Handling at Cloth–Fluid Contact, Computer Graphics Forum, 34(1), 14-25, 2015.

Furthermore, material from an IEEE copyrighted paper is partly reused with kind permission of IEEE (Thesis / Dissertation Reuse):

- [3] ©2017 IEEE. Reprinted, with permission, from M. Huber, B. Eberhardt, and D. Weiskopf, Cloth Animation Retrieval Using a Motion-Shape Signature, IEEE Computer Graphics and Applications, 37(6), 54-64, 2017.

Chapter 2

Physically-based Simulation of Fluids and Cloth

This chapter familiarizes the reader with the fundamental concepts and methods used in the physically-based simulation of fluids and cloth. It presents the discretizations of the governing equations and illustrates techniques that allow an efficient implementation in computer graphics applications. Hence, this chapter forms the computational basis for the contributions of the subsequent chapters of this thesis. The following derivations and summarization introduce consistent terminology and provide theoretical background.

2.1 Basics of SPH-based Fluid Simulation

This first section presents basic principles of fluid dynamics and the methods for particle-based simulation of fluids using SPH. To begin with, the governing equations of fluid dynamics are given in Section 2.1.1, which lead to the equation of motion for fluids. The concepts of SPH that are needed for an application to fluid simulation are presented in Section 2.1.2. In Section 2.1.3, a typical spatial discretization of the governing equation of motion for fluids using SPH is given, followed by the temporal discretization used in numerical time integration (Section 2.1.4). An example of a basic SPH-based fluid solver for computer graphics applications is given in Section 2.1.5.

2.1.1 Fluid Dynamics

The dynamics of fluids are based on three fundamental physical principles, namely

- the conservation of mass,
- the conservation of momentum (Newton's second law),
- and the conservation of energy.

These principles give rise to the governing equations of fluid dynamics, which are the continuity equation, momentum equation, and energy equation. They form a coupled system of nonlinear PDEs describing viscous flow.

Continuum mechanics offers two approaches to express the equation of motion of fluids, the Lagrangian and the Eulerian formulation. In the Lagrangian viewpoint, the fluid volume is represented by partial fluid volumes, often called particles, that move through space and time. In the Eulerian formulation, a fixed coordinate system is used to measure fluid quantities at fixed spatial locations.

In this thesis, the focus is on simulating fluids using SPH, which is based on the Lagrangian formulation. However, the following derivation of the equation of motion for fluids is independent from the viewpoint and can be used for both approaches. In the following, only those equations for the dynamics of fluids are addressed that are relevant to this work. For example, thermodynamics is neglected and, therefore, there is no need to consider the energy equation. Comprehensive introductions to fluid dynamics can be found in dedicated textbooks, e.g., by Batchelor [21], Anderson [16], or Kundu and Cohen [65].

For the introduction of conservation laws, a control volume $\mathscr{V}$ in $\mathbb{R}^3$ of finite size and fixed mass that moves with the fluid flow is considered, as illustrated in Figure 2.1.

Conservation of Mass

The control volume $\mathscr{V}$ contains infinitesimally small fluid elements of volume dV of mass ρdV with mass density ρ. Hence, the overall mass of the control volume is

$$m_{\mathscr{V}} = \int_{\mathscr{V}} \rho dV. \tag{2.1}$$

Although the control volume itself can change as it moves with the flow, the principle of conservation of mass states that mass remains constant during motion. For the control volume, this means that all infinitesimally small fluid elements have constant mass and conservation of mass can be expressed by the continuity equation

$$\frac{D}{Dt} m_V = \frac{D}{Dt} \int_{\mathscr{V}} \rho dV = 0, \tag{2.2}$$

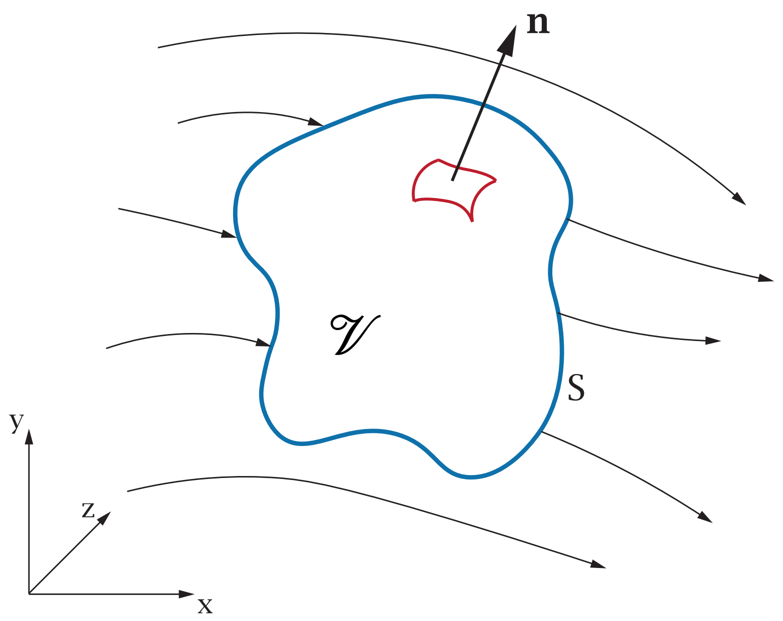


Figure 2.1: A finite control volume $\mathscr{V}$ and its surface S that moves with the fluid flow. The same material particles are always in the control volume. A surface patch dS with surface normal $\mathbf{n}$ is depicted in red.

where D/Dt is the substantial derivative calculated by the sum of the local and the convective derivative as

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla \qquad (2.3)$$

for the velocity $\mathbf{v} = (u,v,w)^T$ of a fluid element.

Conservation of Momentum

The principle of momentum conservation is based on Newton's second law

$$\mathbf{F} = m\mathbf{a}, \qquad (2.4)$$

which, applied to this context, states that the mass m times the acceleration $\mathbf{a}$ of an infinitesimal fluid element equals the net force $\mathbf{F}$ acting on the element. For the finite control volume $\mathscr{V}$ of a fluid, this statement requires that the time rate of change of its momentum $m\mathbf{v}$ equals the net force $\mathbf{F}$ acting on the control volume. This can be written as

$$\frac{\mathbf{F}}{\mathscr{V}} = \int_{\mathscr{V}} \mathbf{f} dV = \int_{\mathscr{V}} \frac{\rho D\mathbf{v}}{Dt} dV, \qquad (2.5)$$

where $\mathbf{v}$ is the fluids' velocity. The force on a volume element is considered as force per unit volume $\mathbf{f} = \mathbf{F}/V$ which is used in the following. At this point, it also has to be noted that the substantial derivative can be moved inside the integral in Equation 2.5, because the elements of the volume $\mathscr{V}$ do not change and mass is constant. The net force $\mathbf{f}$ consists of body forces $\mathbf{f}^b$ and surface forces $\mathbf{f}^s$. Body forces act on the entire control volume, usually without contact. They act at a distance such as gravitational, magnetic, or electric forces. Surface forces act directly and exclusively on the surface of the fluid volume and are given as forces per unit area. They can be separated in normal and tangential direction of the surface area. Examples of surface forces are forces due to pressure and forces due to shearing, resulting from friction. The surface force on a fluid element is given by

$$\mathbf{f}^s = \sigma \cdot \mathbf{n}, \tag{2.6}$$

with the 3×3 viscous stress tensor σ and the normal $\mathbf{n}$ of the surface area. Splitting the net force into body forces and surface force, Equation 2.5 reads

$$\begin{aligned} \int_{\mathscr{V}} \rho \frac{D\mathbf{v}}{Dt} dV &= \int_{\mathscr{V}} \mathbf{f}^b dV + \int_{\partial\mathscr{V}} \mathbf{f}^s dS \\ &= \int_{\mathscr{V}} \mathbf{f}^b dV + \int_{\partial\mathscr{V}} \sigma \cdot \mathbf{n} dS, \end{aligned} \tag{2.7}$$

where surface forces are written in terms of force per area $\mathbf{f}^s = \mathbf{F}^s / A$ and are zero inside of the volume. To transform the surface integral into a volume integral, the divergence theorem (also known as Gauss' theorem)

$$\int_{\mathscr{V}} \nabla \cdot \mathbf{U} dV = \int_{\partial\mathscr{V}} \mathbf{U} \cdot \mathbf{n} dS \tag{2.8}$$

can be applied, which relates the volume integral of the divergence of a vector field $\mathbf{U}$ to its surface integral. Hence, the formulation of change of momentum for a control volume of a fluid can be expressed as

$$\begin{aligned} \int_{\mathscr{V}} \rho \frac{D\mathbf{v}}{Dt} dV &= \int_{\mathscr{V}} \mathbf{f}^b dV + \int_{\mathscr{V}} \nabla \cdot \sigma dV \\ &= \int_{\mathscr{V}} \left(\mathbf{f}^b + \nabla \cdot \sigma \right) dV. \end{aligned} \tag{2.9}$$

Since the control volume $\mathscr{V}$ is arbitrary, the differential form of the above equation follows from the integral form as

$$\rho \frac{D\mathbf{v}}{Dt} = \mathbf{f}^b + \nabla \cdot \sigma, \tag{2.10}$$

which holds for an infinitesimally small fluid element. Equation 2.10 is also known as the Cauchy momentum equation.

In the above equation, the stress at a fluid element is specified by the symmetric tensor

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{pmatrix}, \tag{2.11}$$

with the individual components in terms of local coordinates $(x,y,z)^T$ for normal and shear stress. In this stress matrix, normal stresses are the diagonal elements, while off-diagonal elements represent tangential (shear) stresses. Since forces due to pressure p act along the surface normal (in positive or negative direction), $\boldsymbol{\sigma}$ can be split into two parts and be expressed as

$$\boldsymbol{\sigma} = \begin{pmatrix} -p & 0 & 0 \\ 0 & -p & 0 \\ 0 & 0 & -p \end{pmatrix} + \begin{pmatrix} \sigma_{xx} + p & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} + p & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} + p \end{pmatrix}. \tag{2.12}$$

The difference between the pressure stress tensor and the total stress tensor is called deviatoric stress and accounts for viscosity, which is the resistance of the fluid against deformation. Thus, denoting the deviatoric stress tensor as $\mathbf{T}$, $\boldsymbol{\sigma}$ can be expressed as

$$\boldsymbol{\sigma} = -p\mathbf{I} + \mathbf{T}, \tag{2.13}$$

using the identity matrix $\mathbf{I}$. Considering a Newtonian fluid, the stress $\mathbf{T}$ is proportional to the rate of deformation, which is the change in velocity in directions of stress. With the velocity vector $\mathbf{v} = (u,v,w)^T$, it is defined as

$$\mathbf{T} = \mu \begin{pmatrix} \frac{\partial u}{\partial x} + \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} + \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} & \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} + \frac{\partial w}{\partial z} \end{pmatrix}, \tag{2.14}$$

where μ is the dynamic viscosity constant. In order to use $\mathbf{T}$ in Equation 2.10, the divergence of the deviatoric stress tensor $\nabla \cdot \mathbf{T}$ is needed. For the first component of $\nabla \cdot \mathbf{T}$, the divergence is calculated by

$$\begin{aligned} (\nabla \cdot \mathbf{T})_1 &= \mu \left(\frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial z} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \right) \\ &= \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 v}{\partial y \partial x} + \frac{\partial^2 u}{\partial z^2} + \frac{\partial^2 w}{\partial z \partial x} \right) \\ &= \mu \left(\frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \\ &= \mu \frac{\partial}{\partial x} (\nabla \cdot \mathbf{v}) + \mu \nabla^2 u. \end{aligned}$$

For an incompressible fluid, the rate of change, and therefore, $\nabla \cdot \mathbf{v}$, is zero and the first term on the right hand side vanishes. Performing the above calculation for all components of the matrix, the divergence of the stress tensor results in the vector Laplacian

$$\nabla \cdot \mathbf{T} = \mu \nabla^2 \mathbf{v}. \tag{2.15}$$

Using $\nabla(p\mathbf{I}) = \nabla p$ and Equation 2.15, Equation 2.10 reads

$$\rho \frac{D\mathbf{v}}{Dt} = \mathbf{f}^b - \nabla p + \mu \nabla^2 \mathbf{v}, \tag{2.16}$$

also known as the Navier-Stokes equation for the motion of a fluid. By multipling both sides of the equation with $\frac{1}{\rho}$, Equation 2.16 leads to

$$\frac{D\mathbf{v}}{Dt} = \frac{\mathbf{F}^b}{m} - \frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v}, \tag{2.17}$$

introducing the kinematic viscosity coefficient $\nu = \frac{\mu}{\rho}$.

2.1.2 Smoothed Particle Hydrodynamics

The governing equations that describe the motion of a fluid have been derived assuming a continuous representation of the fluid. For a numerical solution of these equations, the resulting PDE have to be discretized to obtain a set of ODEs with respect to time. In this thesis, spatial discretization is performed by a Lagrangian method in which a finite set of interpolation points represent the fluid domain and move with the flow. The spatial derivatives of the governing PDEs are approximated based on the sample points using the SPH method. In this section, the derivation and the basic formulation of SPH are presented. The review article by Monaghan [80] and the text book by Liu and Liu [69] are recommended for further reading.

SPH was introduced independently by Gingold and Monaghan [43] and Lucy [72] for the solution of dynamic astrophysical problems. The method is based on kernel estimation techniques already known from statistics. The underlying principle of the method is that a quantity A, such as a scalar or a vector, at a position x can be represented by the integral formulation

$$A(x) = \int_\Omega A(x')\delta(||x - x'||)dx', \tag{2.18}$$

where x, x' are data points inside the domain Ω and δ is the Dirac or delta function. For practical purposes, the Dirac function is replaced with a kernel function $W(||x - x'||, h)$ with support h, which leads to the kernel approximation

$$\widetilde{A}(x) = \int_\Omega A(x')W(||x - x'||, h)dx', \tag{2.19}$$

where $\widetilde{A} = A$, if W equals the Dirac function. For a computer implementation, the choice of the kernel function has a considerable influence on accuracy and performance. A wide range of kernel functions can be found in literature for different applications, but all of them are modeled to fulfill the following conditions to produce valid approximations:

- for $h \to 0$, the kernel function converges to the Dirac function
- the kernel function is normalized

$$\int_{\Omega} W(||x - x'||, h) dx' = 1$$

- it has compact support

$$W(||x - x'||, h) = 0, \text{ if } ||x - x'|| > h$$

- the kernel function is non-negative

$$W(||x - x'||, h) \geq 0.$$

For numerical computation, the integral in the continuous setting (Equation 2.19) can be discretized using a sum over the finite set of sampling points of size N. With SPH, a material is represented with particles of volume V. A quantity A at the position $\mathbf{x}$ can be calculated with the so called particle approximation

$$A(\mathbf{x}) \approx \sum_{j=1}^{N} A_j W(||\mathbf{x}_j - \mathbf{x}||, h) V_j, \tag{2.20}$$

where $A_j = A(\mathbf{x}_j)$ for particle j is used. For the simulation of fluids, the particles represent a part of the fluid volume. Replacing V_j by a particles' mass m_j divided its density ρ_j and using the short notations $A_i = A(\mathbf{x}_i)$ and $W_{ij} = W(||\mathbf{x}_j - \mathbf{x}_i||, h)$, fluid quantities can be calculated with

$$A_i \approx \sum_{j=1}^{N} \frac{m_j}{\rho_j} A_j W_{ij}, \tag{2.21}$$

which is the basic form of SPH approximation for different quantities A_i.

To apply the SPH method for solving PDEs as the motion equation of a fluid (Equation 2.17), the first two spatial derivatives of the approximation A_i are needed. Therefore, $\widetilde{A}$ and A are replaced with their respective spatial derivatives in the integral formulation (Equation 2.19), which leads to

$$\nabla_x \widetilde{A}(x) = \int_{\Omega} \left(\nabla_{x'} A(x') \right) W(||x - x'||, h) dx'. \tag{2.22}$$

Applying the product rule to the integrand on the right hand side of the equation, the integral can also be written as

$$\int_\Omega \left(\nabla_{x'} A(x')\right) W(||x-x'||,h)dx' =$$

$$\int_\Omega \nabla_{x'} \left(A(x')W(||x'-x||,h)\right) dx' - \int_\Omega A(x') \left(\nabla_{x'} W(||x'-x||,h)\right) dx'.$$

Using this substitution, Equation 2.22 reads

$$\begin{aligned}\nabla_x \widetilde{A}(x) = &\int_\Omega \nabla_{x'} \left(A(x')W(||x'-x||,h)\right) dx' \\ &- \int_\Omega A(x') \left(\nabla_{x'} W(||x'-x||,h)\right) dx'.\end{aligned} \tag{2.23}$$

Gauss' theorem (Equation 2.8) is also applicable to the gradient scalar functions in the alternate form [17]

$$\int_V \nabla U dV = \int_{\partial V} U \cdot \mathbf{n} dS, \tag{2.24}$$

with the surface ∂V and its normal $\mathbf{n}$, and can be applied to the first integral in Equation 2.23:

$$\begin{aligned}\nabla_x \widetilde{A}(x) = &\int_{\partial V} A(x')W(||x'-x||,h) \cdot \mathbf{n} dS \\ &- \int_\Omega A(x') \left(\nabla_{x'} W(||x'-x||,h)\right) dx'.\end{aligned} \tag{2.25}$$

At this point, the kernel function condition of compact support can be utilized, which implies that the surface integral is zero. Applied to Equation 2.25, the first spatial derivative of $\widetilde{A}$ is

$$\nabla \widetilde{A}(x) = \int_\Omega A(x') \nabla W(||x'-x||,h) dx'. \tag{2.26}$$

Similarly, the Laplacian of Equation 2.19 can be obtained:

$$\nabla^2 \widetilde{A}(x) = \int_\Omega A(x') \nabla^2 W(||x-x'||,h) dx'. \tag{2.27}$$

It follows that the first and second derivative of a value A_i are easily calculated using the particle approximation (Equation 2.21) with

$$\nabla A_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} A_j \nabla W_{ij}, \tag{2.28}$$

$$\nabla^2 A_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} A_j \nabla^2 W_{ij}, \tag{2.29}$$

where N is the number of particles.

2.1.3 Spatial Discretization Using SPH

The equations of the previous section can be applied for the spatial discretization of the governing fluid equations. This section summarizes the methods for calculating the most basic quantities needed for simulating fluids. A more elaborate overview of varying methods and numerous extensions for computer graphics can be found in the survey of Ihmsen et al. [56].

If a fluid is discretized with Lagrangian particles that move with the fluid, the equation of motion (Equation 2.17) in the form of a PDE derived in Section 2.1.1 for a particle i can be written as

$$\frac{D\mathbf{v}_i}{Dt} = \frac{\mathbf{F}_i^b}{m_i} - \frac{1}{\rho_i}\nabla p_i + \nu\nabla^2\mathbf{v_i}. \tag{2.30}$$

Multiplying both sides of the above equation with m_i results in the net force

$$\mathbf{F}_i = \mathbf{F}_i^b - \underbrace{\frac{m_i}{\rho_i}\nabla p_i}_{\mathbf{F}_i^p} + \underbrace{m_i\nu\nabla^2\mathbf{v_i}}_{\mathbf{F}_i^v} \tag{2.31}$$

acting on the particle. As it can be seen, the net force is the sum of body forces $\mathbf{F}^b$, pressure force $\mathbf{F}^p$, and viscosity force $\mathbf{F}^v$. To calculate the motion of a fluid, the three forces have to be evaluated for each particle and integrated in time. In the basic setting, only gravity is considered as a body force and can be computed for each particle by $\mathbf{F}_i^b = m_i\mathbf{g}$ using standard gravity.

Pressure Force

The pressure force term in Equation 2.31 can be calculated using the first derivative of the particle approximation of SPH (Equation 2.28) as

$$\mathbf{F}_i^p = -\frac{m_i}{\rho_i}\sum_{j=1}^{N}\frac{m_j}{\rho_j}p_j\nabla W(||\mathbf{x}_i - \mathbf{x}_j||, h). \tag{2.32}$$

However, the force in this notation is not symmetric. In addition, linear and angular momentum are not preserved since only the pressure of particle j is considered for the pressure force acting on particle i. Monaghan [79] proposes symmetrizing the pressure gradient, which leads to

$$\mathbf{F}_i^p = -m_i\sum_{j=1}^{N} m_j\left(\frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2}\right)\nabla W(||\mathbf{x}_i - \mathbf{x}_j||, h) \tag{2.33}$$

for the pressure force term.

Density and Pressure

For the calculation of pressure forces (Equation 2.33), density and pressure values for all particles have to be known. The density can be evaluated for each particle by applying the SPH approximation of Equation 2.21

$$\begin{aligned} \rho_i &= \sum_{j=1}^{N} \frac{m_j}{\rho_j} \rho_j W(||\mathbf{x}_i - \mathbf{x}_j||, h) \\ &= \sum_{j=1}^{N} m_j W(||\mathbf{x}_i - \mathbf{x}_j||, h). \end{aligned} \tag{2.34}$$

Hence, density is determined by the weighted mass of all particles in the influence radius.

One of the commonly used methods to evaluate pressure at a particle location is by using an equation of state (EOS). With this approach, the pressure p_i for a particle i is calculated by

$$p_i = \frac{\kappa \rho_0}{\gamma} \left(\left(\frac{\rho_i}{\rho_0} \right)^{\gamma} - 1 \right), \tag{2.35}$$

where κ is a stiffness constant, γ a pressure constant, and ρ_0 is the rest or reference density. In computer graphics literature, e.g. [38][83], γ is often chosen to be 1, which leads to the simplified expression

$$p_i = \kappa(\rho_i - \rho_0). \tag{2.36}$$

Viscosity Force

For the viscosity force, the Laplacian of the particle approximation (Equation 2.29) can be applied, resulting in

$$\mathbf{F}_i^v = \nu m_i \sum_{j=1}^{N} \frac{m_j}{\rho_j} \mathbf{v}_j \nabla^2 W(||\mathbf{x}_i - \mathbf{x}_j||, h). \tag{2.37}$$

Similar to the pressure force evaluation, this standard formulation is non-symmetric and leads to instabilities at surfaces and sparsely sampled regions. These problems can be avoided using the artificial viscosity force evaluation by Morris et al. [82]:

$$\mathbf{F}_i^v = 2\nu m_i \sum_{j=1}^{N} \frac{m_j}{\rho_j} (\mathbf{v}_i - \mathbf{v}_j) \left(\frac{(\mathbf{x}_i - \mathbf{x}_j) \cdot \nabla W(||\mathbf{x}_i - \mathbf{x}_j||, h)}{(\mathbf{x}_i - \mathbf{x}_j)^2 + 0.01h^2} \right). \tag{2.38}$$

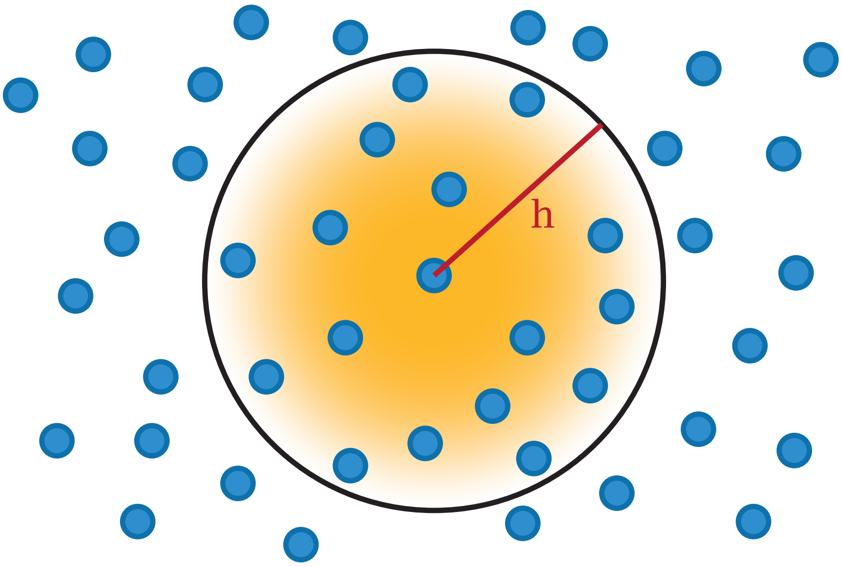


Figure 2.2: The fluid quantities of particles are approximated by their compact neighborhood with a smoothing kernel of influence radius h. The influence of a neighboring particle decreases with its distance.

Kernel Functions

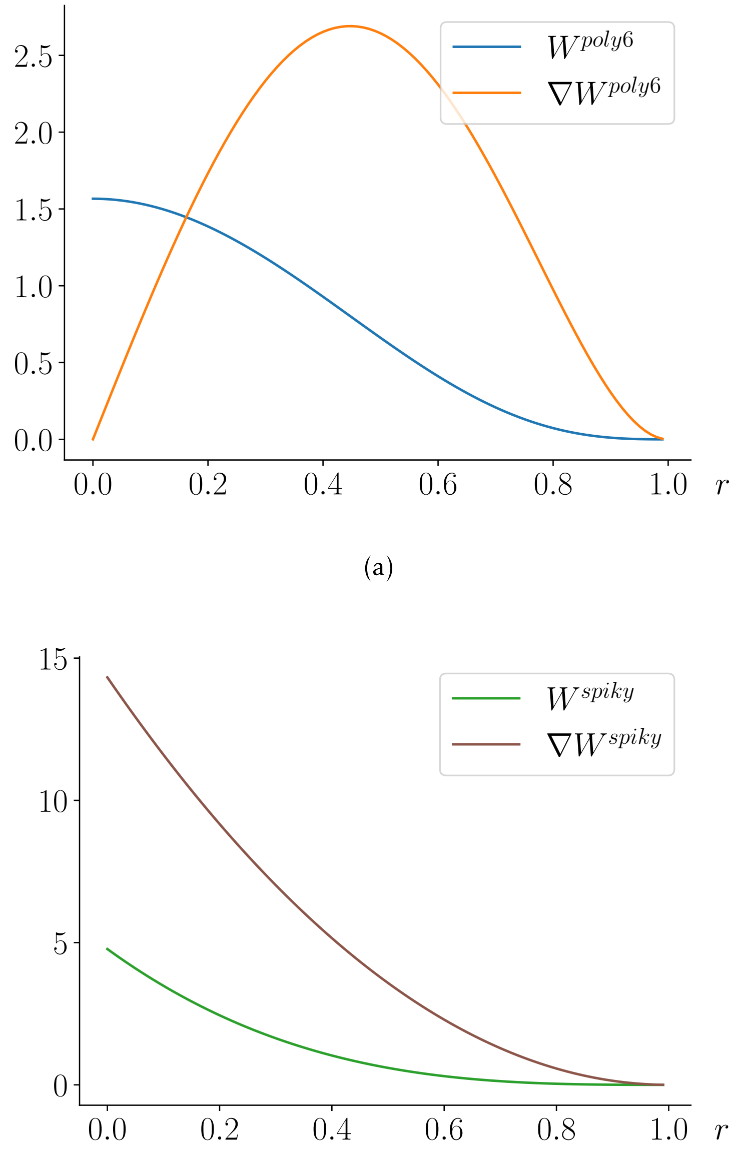
In Section 2.1.2, the desired properties for SPH kernel functions have been discussed. The influence of the kernel with smoothing length h is illustrated in Figure 2.2, where the influence of the kernel is color-coded. For computer graphics applications, different kernels have been proposed, e.g., by Desbrun and Gascuel [38] and Müller et al. [83]. Key factors for the choice of kernel function are computational cost and accuracy. A popular choice for most applications is the polynomial kernel

$$W^{poly6}(r,h) = \frac{315}{64\pi h^9} \begin{cases} \left(h^2 - r^2\right)^3 & \text{if } r \leq h \\ 0 & \text{else,} \end{cases} \tag{2.39}$$

presented by Müller et al. [83], where $r = ||\mathbf{x} - \mathbf{x}'||$. For one dimension, a plot of the kernel and its gradient is shown in Figure 2.3(a). However, the gradient of this kernel, which is needed for pressure force and viscosity force evaluations (Equations 2.33 and 2.38), has a major drawback: when the distance between two particles tends toward zero, the gradient gets smaller, and close-by particles are underrepresented. Therefore, Müller et al. propose using a *spiky* kernel for calculations that involve the kernel gradient which reads

$$W^{spiky}(r,h) = \frac{15}{\pi h^6} \begin{cases} (h - r)^3 & \text{if } r \leq h \\ 0 & \text{else.} \end{cases} \tag{2.40}$$

The kernel and its gradient are depicted in Figure 2.3(b).



(a)

(b)

Figure 2.3: Plots of the SPH kernel functions proposed by Müller et al. [83]: (a) The polynomial kernel W^{poly6} and its derivative are shown. (b) The spiky kernel W^{spiky} that is used for SPH calculations involving the kernel gradient.

2.1.4 Numerical Time Integration

At this point, methods to solve each term in the spatially discrete equation of motion (Equation 2.30) in the static equilibrium setting are known and can be transferred to the dynamics of the fluid particle system. The equation of motion for each particle i with location $\mathbf{x}_i(t)$ at time t reads

$$m_i \mathbf{a}_i(t) = \mathbf{F}_i^b(t) - \mathbf{F}_i^p(\mathbf{x}_i(t)) + \mathbf{F}_i^v(\mathbf{v}_i(t)), \tag{2.41}$$

where $\mathbf{a}_i$ is a particle's acceleration. Equation 2.41 is an ODE of second order with respect to time. Together with given initial conditions of starting position and velocity for each particle, the equation is an initial value problem (IVP) that describes the particles' trajectories. A discrete approximation of particle trajectories is obtained using numerical time integration methods. To facilitate time integration, the second order ODE of Equation 2.41 is reduced to the coupled system of first order ODEs, expressed as

$$\frac{d}{dt}\mathbf{x}_i(t) = \mathbf{v}_i(t), \tag{2.42}$$

$$m_i \frac{d}{dt}\mathbf{v}_i(t) = \mathbf{F}_i(t), \tag{2.43}$$

where the forces on the right side of Equation 2.41 are combined in $\mathbf{F}_i$. The simplest representative for an integration scheme is the explicit Euler method. Applied to the coupled system of ODEs, the state of the simulation is calculated for each particle by

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \Delta t \frac{\mathbf{F}_i(t)}{m_i}, \tag{2.44}$$

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \Delta t \mathbf{v}_i(t), \tag{2.45}$$

with a given time step size Δt. As it can be seen, the state of the system at a future time $t + \Delta t$ is evaluated based on the state at the current time t. In many computer graphics applications, the first order explicit Euler scheme is replaced with the semi-implicit Euler expressed as

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \Delta t \frac{\mathbf{F}_i(t)}{m_i}, \tag{2.46}$$

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \Delta t \mathbf{v}_i(t + \Delta t). \tag{2.47}$$

In contrast to the explicit Euler step, the velocity $\mathbf{v}_i(t + \Delta t)$ of the next time step is used to evaluate the particle positions, which leads to second order accuracy. Therefore, accuracy is increased with similar performance.

The time step Δt is usually determined such that it satisfies the Courant–Friedrichs–Lewy (CFL) condition

$$\Delta t \leq \vartheta \frac{h}{||\mathbf{v}^{max}||}, \tag{2.48}$$

where $||\mathbf{v}^{max}||$ is the maximum velocity of all particles and ϑ is usually set to 0.4 [56], [79].

2.1.5 Fluid Animation Loop

With the techniques of the previous sections, all necessary building blocks are available to build a basic SPH-based fluid solver as illustrated in Figure 2.4. Using the EOS (Equation 2.35) for pressure evaluation, the algorithm is often called SESPH. In each simulation step, four loops over all input particles are performed: First, the neighborhood for each particle is calculated and stored to ensure an efficient look-up of particles within the influence radius h in the following kernel evaluations. Subsequently, density and pressure are computed for all particles. In the third loop, all forces are calculated and added together that are then used to numerically integrate the velocities and positions of all particles in the last loop.

For realistic animations, surface tension has an major impact on the behavior of a liquid. Surface tension is usually modeled as an additional force and contributes to the net force. This thesis presents an evaluation of surface tension models for SPH-based fluid simulations and the different approaches are given in Chapter 6.

If there are other objects in the scene, interaction has to be modeled that may include two-way coupling with special boundary conditions. Whereas the interaction of rigid bodies and fluids was studied earlier [14], [24], [54], [97], modeling the interaction between particle-based fluids and thin deformable objects faces additional challenges. An approach on this is presented in Chapter 4.

The fluid dynamics loop and interaction is usually based on the particle representation of the fluid. As discussed in Section 1.1, the surface is given as an implicit representation by a density or distance field based on the particle densities and positions. To obtain an explicit surface, the marching cubes (MC) algorithm is applied to extract a surface mesh. Chapter 4 introduces a method to prevent possible intersection of the fluids' surface with interacting objects at contact.

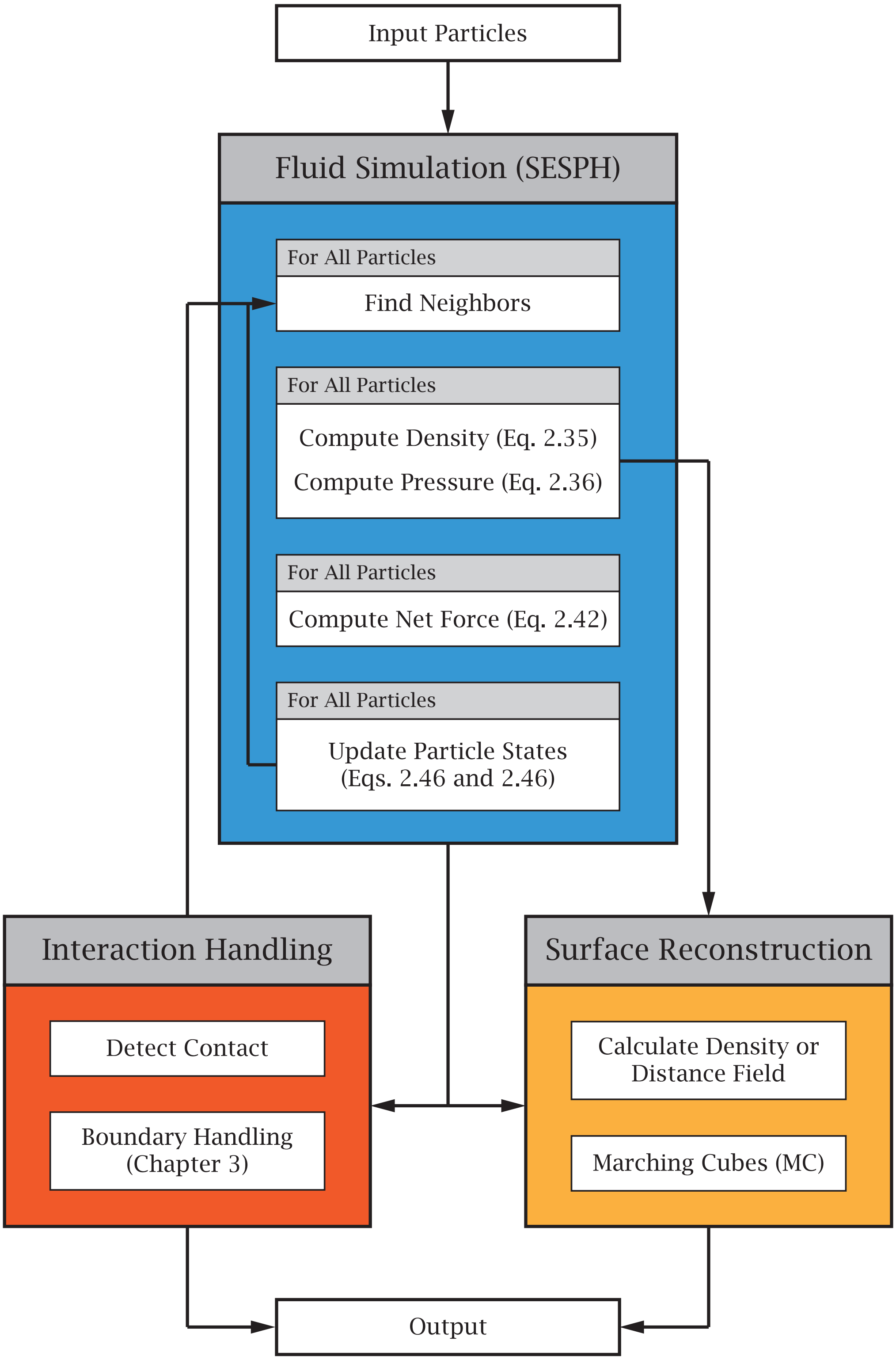


Figure 2.4: The fluid animation loop.

2.2 BASICS OF CLOTH SIMULATION

Cloth is usually considered as a thin deformable object and planar deformations can be modeled based on continuum mechanics. Section 2.2.1 starts with the description of a suitable measure for deformation, followed by introducing the concept of stress and mechanical equilibrium. Then, the relevant material laws of linear elastics for thin deformable objects are given. Spatial discretization of the equilibrium equations with the finite element method (FEM) that lead to force computations for planar deformation is demonstrated in Section 2.2.2. Bending deformation is handled separately from planar deformation and discrete models are described in Section 2.2.3. In Section 2.2.4, numerical time integration is discussed and finally, the building blocks are assembled to a basic cloth solver (Section 2.2.5).

2.2.1 Continuous Representation of Deformable Objects

The foundations of deformable objects based on a continuous representation are covered well in textbooks and for further reading, the extensive introduction to continuum mechanics by Bonet and Wood [28] is recommended. For modeling cloth in computer graphics, a summary and additional related topics can be found in the cloth simulation tutorial notes by Thomaszewski et al. [109].

Deformation and Strain

A deformable body $\mathcal{B}$ that covers a region in space $\bar{\Omega} \subset \mathbb{R}^3$ in its initial configuration and $\Omega \subset \mathbb{R}^3$ in its current, deformed configuration, can be described by a time-dependent mapping

$$\boldsymbol{\varphi} : \bar{\Omega} \times [0,\infty) \to \Omega \subset \mathbb{R}^3 \tag{2.49}$$

between initial configuration and current configuration, known as configuration mapping (see Figure 2.5). The function $\boldsymbol{\varphi}$ can be applied to the positions $\mathbf{x}_i$ of material particles of $\mathcal{B}$ in initial configuration, which leads to

$$\mathbf{x}_i(t) = \boldsymbol{\varphi}(\mathbf{x}_i(t_0), t) \tag{2.50}$$

and maps the positions from initial configuration at time t_0 to current configuration at time t. In the following, the notation for positions $\bar{\mathbf{x}}_i = \mathbf{x}_i(t_0)$ and $\mathbf{x}_i = \mathbf{x}_i(t)$ is used.

Given the above description of the deformable object, the first step is to obtain a measure for deformation. To this end, deformation is first described based on the relative change of positions of material particles. Considering three

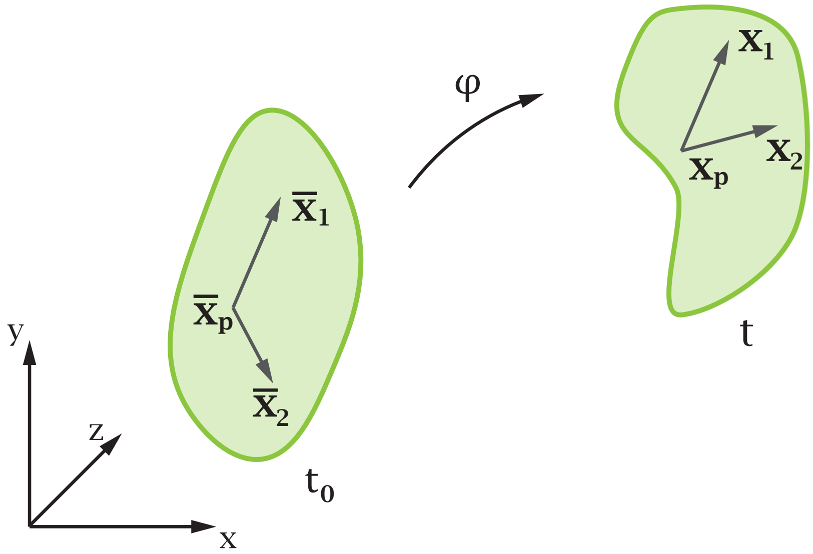


Figure 2.5: Motion and change of configuration at deformation of a body. $\boldsymbol{\varphi}$ maps a deformable body from initial to current, deformed configuration.

material particles in rest configuration with positions $\bar{\mathbf{x}}_p$, $\bar{\mathbf{x}}_1$, and $\bar{\mathbf{x}}_2$ (Figure 2.5), connecting vectors with respect to $\bar{\mathbf{x}}_p$ are given by

$$d\bar{\mathbf{x}}_{p1} = \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_p, \tag{2.51}$$

$$d\bar{\mathbf{x}}_{p2} = \bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_p, \tag{2.52}$$

where $d\bar{\mathbf{x}}_{p1}$ and $d\bar{\mathbf{x}}_{p2}$ are infinitesimal vectors. The positions in deformed state are expressed by the configuration mapping (Equation 2.50) as

$$\mathbf{x}_p = \boldsymbol{\varphi}(\bar{\mathbf{x}}_p), \quad \mathbf{x}_1 = \boldsymbol{\varphi}(\bar{\mathbf{x}}_1), \quad \mathbf{x}_2 = \boldsymbol{\varphi}(\bar{\mathbf{x}}_2) \tag{2.53}$$

and the corresponding vectors result as

$$d\mathbf{x}_{p1} = \mathbf{x}_1 - \mathbf{x}_p = \boldsymbol{\varphi}(\bar{\mathbf{x}}_p + d\bar{\mathbf{x}}_1) - \boldsymbol{\varphi}(\bar{\mathbf{x}}_p), \tag{2.54}$$

$$d\mathbf{x}_{p2} = \mathbf{x}_2 - \mathbf{x}_p = \boldsymbol{\varphi}(\bar{\mathbf{x}}_p + d\bar{\mathbf{x}}_2) - \boldsymbol{\varphi}(\bar{\mathbf{x}}_p). \tag{2.55}$$

To facilitate mapping vectors in initial configuration to their deformed configuration, the deformation gradient tensor $\mathbf{F}$ is introduced by

$$\mathbf{F} = \frac{\partial \boldsymbol{\varphi}}{\partial \bar{\mathbf{x}}} = \nabla \boldsymbol{\varphi}. \tag{2.56}$$

The vectors $d\mathbf{x}_{p1}$ and $d\mathbf{x}_{p2}$ can then be expressed in terms of $\mathbf{F}$ as

$$d\mathbf{x}_{p1} = \mathbf{F} d\bar{\mathbf{x}}_{p1}, \tag{2.57}$$

$$d\mathbf{x}_{p2} = \mathbf{F} d\bar{\mathbf{x}}_{p2}. \tag{2.58}$$

To actually measure deformation, the change of the dot product between the pair of vectors in initial and current configuration can be used. It encodes changes in length as well as angular changes of the vectors. The dot product of the vectors $d\mathbf{x}_{p1}$ and $d\mathbf{x}_{p2}$ using deformation gradient notation of Equations 2.57 and 2.58 leads to

$$\begin{aligned} d\mathbf{x}_{p1} \cdot d\mathbf{x}_{p2} &= (\mathbf{F}d\bar{\mathbf{x}}_{p1}) \cdot (\mathbf{F}d\bar{\mathbf{x}}_{p2}) \\ &= d\bar{\mathbf{x}}_{p1}(\mathbf{F}^T\mathbf{F})d\bar{\mathbf{x}}_{p2} \\ &= d\bar{\mathbf{x}}_{p1}\mathbf{C}d\bar{\mathbf{x}}_{p2}, \end{aligned} \tag{2.59}$$

where $\mathbf{C} = \mathbf{F}^T\mathbf{F}$ denotes the right Cauchy-Green deformation tensor. Alternatively, the same procedure can be applied to the difference in dot products of initial and current configuration that can be written as

$$\begin{aligned} d\mathbf{x}_{p1} \cdot d\mathbf{x}_{p2} - d\bar{\mathbf{x}}_{p1} \cdot d\bar{\mathbf{x}}_{p2} &= (\mathbf{F}d\bar{\mathbf{x}}_{p1}) \cdot (\mathbf{F}d\bar{\mathbf{x}}_{p2}) - d\bar{\mathbf{x}}_{p1} \cdot d\bar{\mathbf{x}}_{p2} \\ &= d\bar{\mathbf{x}}_{p1}(\mathbf{F}^T\mathbf{F})d\bar{\mathbf{x}}_{p2} - d\bar{\mathbf{x}}_{p1} \cdot d\bar{\mathbf{x}}_{p2} \\ &= d\bar{\mathbf{x}}_{p1}(\mathbf{F}^T\mathbf{F} - \mathbf{I})d\bar{\mathbf{x}}_{p2}, \end{aligned} \tag{2.60}$$

with the identity matrix $\mathbf{I} \in \mathbb{R}^{3\times3}$. In this case, the nonlinear and symmetric Green strain tensor $\mathbf{E}$ can be identified, which is defined as

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I}) = \frac{1}{2}(\mathbf{C} - \mathbf{I}). \tag{2.61}$$

A fundamental property of these deformation measures becomes evident when a polar decomposition of $\mathbf{F}$ is performed: the deformation gradient can be decomposed in a product of a rotation tensor and a pure stretch tensor as

$$\mathbf{F} = \mathbf{R}\mathbf{U}. \tag{2.62}$$

Applied to the Cauchy-Green tensor $\mathbf{C}$, this leads to

$$\mathbf{F}^T\mathbf{F} = \mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U} = \mathbf{U}^2. \tag{2.63}$$

As $\mathbf{R}$ is a pure rotation tensor, $\mathbf{R}^T\mathbf{R}$ equals the identity matrix $\mathbf{I}$, hence, rotational invariance of the deformation measures is given.

The Green strain tensor $\mathbf{E}$ can also be expressed in terms of the displacement field $\mathbf{u}$ defined as $\mathbf{u} = \mathbf{x} - \bar{\mathbf{x}}$. With the deformation gradient given in terms of $\mathbf{u}$ as

$$\mathbf{F} = \mathbf{I} + \nabla\mathbf{u}, \tag{2.64}$$

it can be written as

$$\mathbf{E} = \frac{1}{2}(\nabla \mathbf{u}^T + \nabla \mathbf{u} + \nabla \mathbf{u}^T \nabla \mathbf{u}). \quad (2.65)$$

With this expression, the Green strain can be linearized by omitting the nonlinear term $\nabla \mathbf{u}^T \nabla \mathbf{u}$, which is a valid approximation if only small displacements are assumed. This leads to the definition of Cauchy strain

$$\varepsilon_C = \frac{1}{2}(\nabla \mathbf{u}^T + \nabla \mathbf{u}), \quad (2.66)$$

which has a widespread use in computer graphics literature. However, this linearization has the drawback that it is no longer invariant regarding rotations, which can lead to erroneous results. Therefore, several researchers propose using the polar decomposition of the deformation gradient (Equation 2.62) and extract the rotational part of the deformation before the actual strain computation [41], [76], [85], [108]. The rotated linear strain tensor then reads

$$\varepsilon_{CR}(\mathbf{Ru}) = \varepsilon(\mathbf{R}^T \mathbf{Ru}) = \varepsilon(\mathbf{u}) \quad (2.67)$$

and is known as co-rotational strain ε_{CR}.

Stress and Equilibrium

Strain is usually accompanied by forces in an equilibrium state. To derive the differential static equilibrium equations, a volume element V of finite size of a deformable body with boundary area $\partial V = A$ is considered. The net force acting on the volume element consists of body forces $\mathbf{f}^b$ acting on the entire volume and traction forces $\mathbf{t}$ acting on the boundary of the element. The traction vector $\mathbf{t}$ is defined as the force vector on a differential surface area dA and the symmetric Cauchy stress tensor

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{pmatrix}, \quad (2.68)$$

relates the normal vector $\mathbf{n}$ to $\mathbf{t}$ as

$$\mathbf{t}(\mathbf{x}, \mathbf{n}) = \boldsymbol{\sigma}(\mathbf{x}) \cdot \mathbf{n}. \quad (2.69)$$

In translational equilibrium, the sum of body forces and traction forces is zero and can be expressed as

$$\int_V \mathbf{f}^b dV + \int_{\partial V} \mathbf{t} dA = 0. \quad (2.70)$$

The traction force vector can be replaced by its notation in terms of the stress tensor (Equation 2.69), which leads to

$$\int_V \mathbf{f}^b dV + \int_{\partial V} \boldsymbol{\sigma} \cdot \mathbf{n} dA = 0. \quad (2.71)$$

The surface integral can be transformed into a volume integral using the divergence theorem (Equation 2.8) and the equation can be written as

$$\int_V (\mathbf{f}^b + \nabla \cdot \boldsymbol{\sigma}) dV = 0. \tag{2.72}$$

This statement must hold for any infinitesimally small element of the volume V and, therefore, can be expressed in differential form as

$$\mathbf{f}^b + \nabla \cdot \boldsymbol{\sigma} = 0, \tag{2.73}$$

which is known as the pointwise spatial equilibrium equation for deformable bodies in its strong form.

Material Laws

In the previous section, the concepts of strain and stress have been shown separate from each other. With material laws, stress is linked to the measure of deformation. A simple constitutive relation between stress and strain is given by the linear law

$$\boldsymbol{\sigma} = \boldsymbol{\mathcal{C}} : \boldsymbol{\varepsilon}, \tag{2.74}$$

where the relation is described by the elasticity tensor $\boldsymbol{\mathcal{C}}$. The above equation uses double dot product notation for tensors and the entries of the stress tensor can be calculated using the *Einstein convention* as

$$\sigma_{ij} = \mathcal{C}_{ijkl}\varepsilon_{kl}. \tag{2.75}$$

Equation 2.74 is a generalization of Hooke's law, which is the one-dimensional case of this equation, i.e., a linear spring.

Given the linear Cauchy strain tensor (Equation 2.66) as presented in Section 2.2.1 for a 3D material in matrix notation

$$\begin{aligned}\boldsymbol{\varepsilon}_C &= \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{1}{2}\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) & \frac{1}{2}\left(\frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y}\right) & \frac{\partial u_y}{\partial y} & \frac{1}{2}\left(\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y}\right) \\ \frac{1}{2}\left(\frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z}\right) & \frac{1}{2}\left(\frac{\partial u_z}{\partial y} + \frac{\partial u_y}{\partial z}\right) & \frac{\partial u_z}{\partial z} \end{bmatrix} \\ &= \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix},\end{aligned} \tag{2.76}$$

normal strains are located on the diagonal, and off-diagonal elements are shear strains. Due to its symmetry, the three-dimensional second-order tensor can be written as six-dimensional vector using Voigt's notation as

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{yy} & \varepsilon_{zz} & \varepsilon_{xy} & \varepsilon_{yz} & \varepsilon_{xz} \end{bmatrix}^T. \tag{2.77}$$

The same holds for the Cauchy stress tensor and leads to

$$\boldsymbol{\sigma} = \begin{bmatrix}\sigma_{xx} & \sigma_{yy} & \sigma_{zz} & \sigma_{xy} & \sigma_{yz} & \sigma_{xz}\end{bmatrix}^T. \tag{2.78}$$

Hence, the linear material law in Equation 2.74 relates stress to strain by the symmetric 6×6 matrix $\boldsymbol{\mathcal{C}}$ with 36 components.

As cloth is modeled as a thin elastic surface, stress and strain are reduced to three entries in Voigt's notation

$$\boldsymbol{\varepsilon} = \begin{bmatrix}\varepsilon_{xx} & \varepsilon_{yy} & \varepsilon_{xy}\end{bmatrix}^T \tag{2.79}$$

and

$$\boldsymbol{\sigma} = \begin{bmatrix}\sigma_{xx} & \sigma_{yy} & \sigma_{xy}\end{bmatrix}^T. \tag{2.80}$$

As it can be seen in Equation 2.74, the elasticity tensor $\boldsymbol{\mathcal{C}}$ then has 16 components. Due to symmetries, it can be reduced to 6 independent entries for a linear and anisotropic, two-dimensional material. Similar to stress and strain notation, an elasticity matrix

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix}\mathcal{C}_{1111} & \mathcal{C}_{1122} & \mathcal{C}_{1112}\\ \mathcal{C}_{1122} & \mathcal{C}_{2222} & \mathcal{C}_{1222}\\ \mathcal{C}_{1112} & \mathcal{C}_{1222} & \mathcal{C}_{1112}\end{bmatrix} \tag{2.81}$$

can be constructed. For cloth simulation, the entries of this matrix depend on Young's moduli E for the individual material directions weft (u) and warp (v), and on the shear modulus G. The relation can then be expressed as

$$\begin{bmatrix}\sigma_{uu}\\ \sigma_{vv}\\ \sigma_{uv}\end{bmatrix} = \frac{1}{1-\nu_u\nu_v}\begin{bmatrix}E_u & \nu_u E_v & 0\\ \nu_v E_u & E_v & 0\\ 0 & 0 & G(1-\nu_u\nu_v)\end{bmatrix}\begin{bmatrix}\varepsilon_{uu}\\ \varepsilon_{vv}\\ \varepsilon_{uv}\end{bmatrix}, \tag{2.82}$$

where ν is Poisson's ratio. For an isotropic material, the relation further simplifies, because $E = E_u = E_v$ and $\nu = \nu_u = \nu_v$.

2.2.2 Spatial Discretization Using Finite Elements

In practice, the strong form of the equilibrium equation (2.73) is not easy to evaluate since it has high continuity requirements. Before performing the actual finite element discretization, a weak form is needed. In continuum mechanics, this transformation is achieved by the principle of virtual work. To begin with, the equilibrium equation (2.72) is multiplied with by an arbitrary, but smooth, test function $\delta\mathbf{v} \in \mathbb{R}^3$, which leads to

$$\int_V (\mathbf{f}^b + \nabla \cdot \boldsymbol{\sigma}) \cdot \delta\mathbf{v}^T dV = \int_V (\delta\mathbf{v}^T \cdot \mathbf{f}^b + \delta\mathbf{v}^T \cdot \nabla \cdot \boldsymbol{\sigma}) dV = 0. \tag{2.83}$$

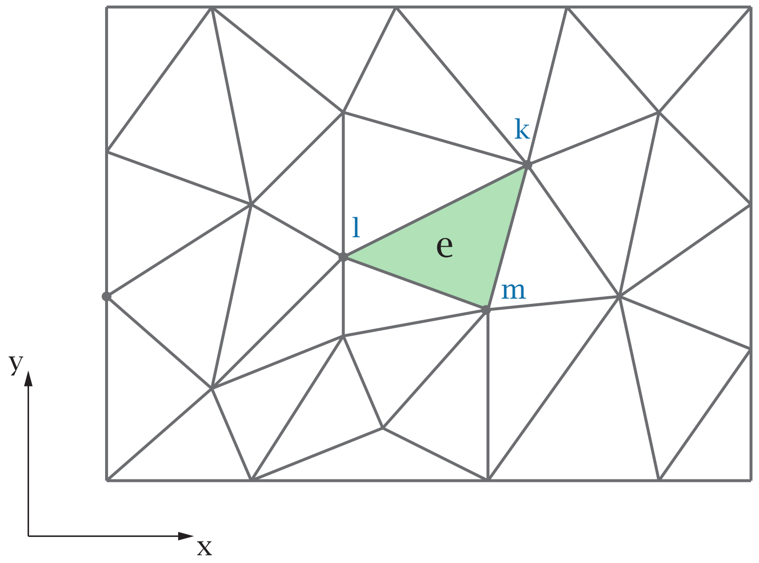


Figure 2.6: A typical 2D cloth discretization with linear finite elements. e denotes a planar element with nodes k, l, m.

The test function can also be replaced by a displacement variation $\delta\mathbf{u}$. Applying integration by parts, exploiting the symmetry of $\boldsymbol{\sigma}$, and the divergence theorem, Equation 2.83 can be expressed as [126]

$$\int_V \delta\varepsilon^T \boldsymbol{\sigma} dV - \int_V \delta\mathbf{u}^T \mathbf{f}^b dV - \int_{\partial V} \delta\mathbf{u}^T \mathbf{t} dA = 0, \tag{2.84}$$

which is the weak form of the Equation 2.73, also known as the virtual work equation. At this point, the equilibrium equation is still valid for both nonlinear and linear stress-strain relation.

Based on Equation 2.84, the finite element discretization is performed. In the following, the discretization will be given for a deformable surface to model cloth. An in detail introduction to the FEM is given in the textbook by Zienkiewicz and Taylor [126]. The deformable surface of cloth will be discretized by triangular (linear) elements as illustrated in Figure 2.6. The continuous displacement field inside an element can be approximated by

$$\mathbf{u} = \sum_{i=1}^{3} N_i \tilde{\mathbf{u}}_i, \tag{2.85}$$

where $i \in \{1,2,3\}$ denotes the node of the element, the nodal in-plane displacement vector is given by $\tilde{\mathbf{u}}_i$, and N_i are linear shape functions (see Figure 2.7).

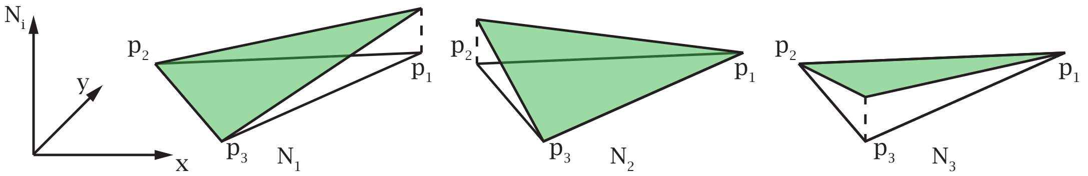


Figure 2.7: Linear finite element shape functions N_i.

For a linear elastic material, Equation 2.84 reads

$$\delta\left(\frac{1}{2}\int_V \boldsymbol{\varepsilon}^T\mathcal{C}\boldsymbol{\varepsilon} - \int_V \mathbf{u}^T\mathbf{f}^b dV - \int_{\partial V}\mathbf{u}^T\mathbf{t}dA\right) = 0. \tag{2.86}$$

Using the discrete approximation for the displacement and the linear strain approximation

$$\boldsymbol{\varepsilon} = \mathbf{B}\tilde{\mathbf{u}}, \tag{2.87}$$

Equation 2.86 can be written as

$$\delta\tilde{\mathbf{u}}\left(\int_V \mathbf{B}^T\mathcal{C}\mathbf{B}dV - \int_V \mathbf{N}^T\mathbf{f}^b dV - \int_{\partial V}\mathbf{N}^T\mathbf{t}dA\right) = 0, \tag{2.88}$$

where $\mathbf{N} = (N_1, N_2, N_3)^T$. Since any variation for $\delta\tilde{\mathbf{u}}^T$ is possible, the equation can be simplified as

$$\int_V \mathbf{B}^T\mathcal{C}\mathbf{B}dV - \int_V \mathbf{N}^T\mathbf{f}^b dV - \int_{\partial V}\mathbf{N}^T\mathbf{t}dA = 0. \tag{2.89}$$

Defining the stiffness matrix $\mathbf{K} = \int_V \mathbf{B}^T\mathcal{C}\mathbf{B}dV$ and expressing the elastic force as $\mathbf{f} = -\int_V \mathbf{N}^T\mathbf{f}^b dV - \int_{\partial V}\mathbf{N}^T\mathbf{t}dA$, the equation can be written as a linear system of (3×3) matrix block entries

$$\begin{bmatrix} \int_V \mathbf{B}_1^T\mathbf{C}\mathbf{B}_1 dV & \cdots & \int_V \mathbf{B}_1^T\mathbf{C}\mathbf{B}_n dV \\ \vdots & \ddots & \vdots \\ \int_V \mathbf{B}_n^T\mathbf{C}\mathbf{B}_1 dV & \ddots & \int_V \mathbf{B}_n^T\mathbf{C}\mathbf{B}_n dV \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_1 \\ \vdots \\ \tilde{\mathbf{u}}_n \end{bmatrix} + \begin{bmatrix} \int_V \mathbf{N}_1^T\mathbf{f}^b dV + \int_{\partial V}\mathbf{N}_1^T\mathbf{t}dA \\ \vdots \\ \int_V \mathbf{N}_n^T\mathbf{f}^b dV + \int_{\partial V}\mathbf{N}_n^T\mathbf{t}dA \end{bmatrix} = 0, \tag{2.90}$$

where n is the number of nodes, or in short notation,

$$\mathbf{K}\tilde{\mathbf{u}} + \mathbf{f} = \mathbf{0}. \tag{2.91}$$

In this linear approach, displacement and strain are constant, and therefore, all $\mathbf{B}_i$ are constant as well. Additionally, a linear material law is used and the evaluation of $\mathbf{K}$ can further be reduced to

$$\mathbf{K} = \sum_e \int_{A_e} \mathbf{B}_e^T\mathcal{C}\mathbf{B}_e dA = \sum_e \mathbf{B}_e^T\mathcal{C}\mathbf{B}_e t A_e, \tag{2.92}$$

where t is the material thickness and A_e the area of a triangular element e. The global stiffness matrix $\mathbf{K}$ can be assembled by evaluating its entries per triangle element e as

$$\mathbf{K}_{ij} = \sum_e \mathbf{K}^e_{ij}. \tag{2.93}$$

For the complete dynamics using linear deformations, damping forces $\mathbf{f}^d = \mathbf{D}\dot{\mathbf{x}}$ can be added and the second-order ODE

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{D}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) + \mathbf{f} = \mathbf{0} \tag{2.94}$$

with respect to time, where $\mathbf{M}$ is a diagonal matrix with the point masses m_i of each node. For the damping matrix, *Rayleigh damping* given as $\mathbf{D} = \alpha\mathbf{M} + \beta\mathbf{K}$ can be employed, where the parameters α and β are user-defined and control the damping. By setting β to zero, simple mass damping is achieved.

2.2.3 Discrete Bending Models

Modeling bending stiffness is an important aspect for textiles as it has an major influence on the folds and wrinkles so typical for the appearance of cloth. Deriving a finite element approach for thin shells, however, is a complex task and requires high computational effort. Although being less accurate, direct discrete bending models are therefore preferred in computer graphics applications. Typical requirements for realistic modeling of bending are consistent behavior for different mesh resolutions and discretizations, anisotropic bending with respect to the material directions, and the possibility of non-flat rest angles. It is important to note that bending is usually modeled completely decoupled from stretch deformation. There have been several approaches to bending in cloth simulations, e.g., simple linear models that are suitable for small deformations such as presented by Bergou et al. [26] and Volino and Magnenat-Thalmann [114], and nonlinear models such as proposed by Grinspun et al. [45] and Bridson et al. [32]. In the context of this thesis, the bending models by Volino and Magnenat-Thalmann and Bridson et al. have been used and are briefly presented in the following.

Given a triangle-based cloth mesh discretization, a bending element consists of two adjacent triangles that share an edge $\mathbf{e} = \mathbf{x}_4 - \mathbf{x}_3$ as shown in Figure 2.8(a). With discrete bending models, a bending force is evaluated based on the local curvature given by the adjacent triangles and distributed to the individual nodes of the bending element. The linear approach by Volino and Magnenat-Thalmann [114] is based on a bending vector $\mathbf{R}^{B_l}$ that is calculated by a linear combination of the nodes $\mathbf{x}_i$ of the bending element expressed as

$$\mathbf{f}_i^{B_l} = -\lambda a_i \mathbf{R}^{B_l}, \tag{2.95}$$

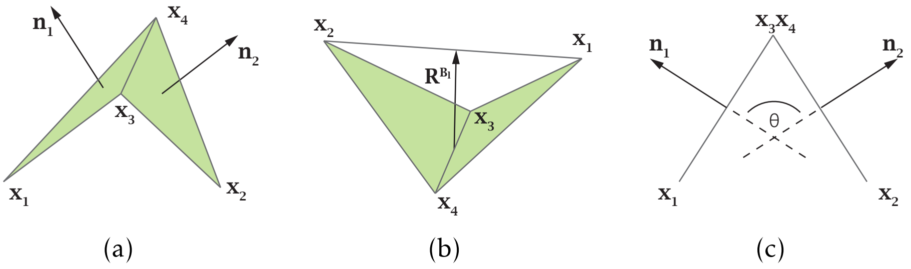


Figure 2.8: Illustrations of a bending element. (a) A bending element consists of two adjacent triangles that share an edge. (b) The bending vector $\mathbf{R}^{B_l}$ as proposed by Volino and Magnenat-Thalmann [114]. (c) The dihedral angle θ between the two triangle normals $\mathbf{n}_1$ and $\mathbf{n}_2$.

where λ is a stiffness coefficient. The bending vector $\mathbf{R}^{B_l}$ is in direction of the edge normal of $\mathbf{e}$ and has the length of the height difference of the triangles (see Figure 2.8(b)). The coefficient a_i is calculated for each node so that translational and rotational momentum is conserved, as shown in the original paper [114]. Non-flat rest angles can be modeled by adding the correspondent rest angle $\mathbf{R}_0^{B_l}$ to $\mathbf{R}^{B_l}$.

While the previous approach may be sufficiently accurate for small bending deformations, a nonlinear approach is advantageous for applications with many details such as fine wrinkles, especially for soft materials. The bending model of Bridson et al. [32] is based on the dihedral angle θ between the face normals $\mathbf{n}_1$ and $\mathbf{n}_2$ as shown in Figure 2.8(c). The bending force is calculated for each node by

$$\mathbf{f}_i^{B_n} = k^B \mathbf{b}_i \left(\sin\left(\frac{\theta}{2}\right) - \sin\left(\frac{\theta_0}{2}\right)\right), \tag{2.96}$$

where k^B is the bending stiffness coefficient and the model-specific vector $\mathbf{b}_i$ is also calculated to account for triangulation anisotropy and conservation of momentum.

2.2.4 Numerical Time Integration

Given the spatially discrete equation of motion for elastic surfaces (Equation 2.94), the temporal evolution of the system has to be solved by numerical time integration. To this end, the dynamic setting using Equation 2.94 and external forces $\mathbf{f}$ can be expressed by

$$\mathbf{Ma}(t) = \mathbf{f}^b - \mathbf{f}^e(\mathbf{x}(t)) - \mathbf{f}^v(\mathbf{v}(t)), \tag{2.97}$$

with the unknown acceleration $\mathbf{a}$. Just as in the simulation of fluids, an IVP with given start conditions (positions and velocities) has to be solved numerically in order to obtain trajectories of the nodes of the discretized surface. Again, Equation 2.97 is a second-order ODE with respect to time that is transformed into a coupled system of first-order ODEs to apply numerical time integration schemes. The reduced system reads as

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{v}(t), \tag{2.98}$$

$$\mathbf{M}\frac{d}{dt}\mathbf{v}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{v}(t), t), \tag{2.99}$$

where the net force $\mathbf{f}$ combines the evaluation of body, elastic, and viscous forces. As for particle-based fluids, the first-order explicit Euler integration scheme

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \Delta t\mathbf{M}^{-1}\mathbf{f}(\mathbf{x}(t), \mathbf{v}(t), t), \tag{2.100}$$

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \Delta t\mathbf{v}(t) \tag{2.101}$$

with time step Δt, seems to be an obvious choice due to its easy implementation. However, for the simulation of elastic materials, the explicit Euler method proves to be disadvantageous because of its limited stability and requirement for very small time steps, and therefore, low computational efficiency in most cases. The reason is that the ODEs of elastodynamic problems contain a stiff[1] part and require advanced numerical methods. With higher-order explicit methods, such as the Verlet method or Runge-Kutta methods, stability can be improved, but in general, implicit integration methods are favorable. Stability, accuracy, and suitability of numerical time integration schemes in computer graphics have been intensively studied [49], [50], [113].

A popular candidate for an implicit time integration scheme for cloth simulation is the implicit or backward Euler method, which in general can be written as

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \Delta t\mathbf{M}^{-1}\mathbf{f}(\mathbf{x}(t+\Delta t), \mathbf{v}(t+\Delta t), t+\Delta t), \tag{2.102}$$

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + h\mathbf{v}(t+\Delta t). \tag{2.103}$$

Just as the explicit Euler scheme, it is a first-order approximation scheme, however, it is unconditionally stable [49]. As it can be seen in Equation 2.102, the state of the system of the future time $t + \Delta t$ appears at both sides of the

[1] Although there is a common understanding of the term *stiff equations*, there is no mathematical definition. A basic property of practical relevance is described by Hairer and Wanner [47]: *"The most pragmatically opinion is also historically the first one (Curtiss & Hirschfelder 1952): stiff equations are equations where certain implicit methods, in particular BDF, perform better, usually tremendously better, than explicit ones."*

equation and a nonlinear system of equations has to be solved. To avoid solving this equation, Baraff and Witkin [19] propose linearizing the forces at the current state by applying a Taylor series expansion and using the first-order approximations

$$\mathbf{f}^e(\mathbf{x}(t+\Delta t)) \approx \mathbf{f}^e(\mathbf{x}(t)) + \frac{\partial \mathbf{f}^e(\mathbf{x}(t))}{\partial \mathbf{x}}\Delta \mathbf{x}, \tag{2.104}$$

$$\mathbf{f}^v(\mathbf{v}(t+\Delta t)) \approx \mathbf{f}^v(\mathbf{v}(t)) + \frac{\partial \mathbf{f}^v(\mathbf{v}(t))}{\partial \mathbf{v}}\Delta \mathbf{v}, \tag{2.105}$$

where

$$\Delta \mathbf{x} = \mathbf{x}(t+\Delta t) - \mathbf{x}(t), \tag{2.106}$$

$$\Delta \mathbf{v} = \mathbf{v}(t+\Delta t) - \mathbf{v}(t). \tag{2.107}$$

Before using the approximation, Equation 2.102 is first multiplied by the mass matrix $\mathbf{M}$ and rewritten as

$$\mathbf{M}\mathbf{v}(t+\Delta t) = \mathbf{M}\mathbf{v}(t) + \Delta t(\mathbf{f}^b + \mathbf{f}^e(\mathbf{x}(t+\Delta t)) + \mathbf{f}^v(\mathbf{v}(t+\Delta t)), \tag{2.108}$$

where body forces $\mathbf{f}^b$ are assumed to be constant over time. Applying the first-order Taylor series approximations (Equations 2.104 and 2.105) and using $\Delta\mathbf{x} = \Delta t \Delta \mathbf{v}$, the linear system of equations

$$\left[\mathbf{M} + \Delta t \frac{\partial \mathbf{f}^v(\mathbf{v}(t))}{\partial \mathbf{v}} + \Delta t^2 \frac{\partial \mathbf{f}^e(\mathbf{x}(t))}{\partial \mathbf{x}}\right] \Delta \mathbf{v} = \Delta t(\mathbf{f}^b - \mathbf{f}^v(\mathbf{v}(t)) - \mathbf{f}^e(\mathbf{x}(t))) \tag{2.109}$$

has to be solved for $\Delta\mathbf{v}$, e.g., using the conjugate gradient (CG) method. Using a linear deformation model, the system can be solved in one step and new positions can be calculated using Equation 2.103. However, the problem can also be nonlinear, which requires an iterative solution such as the Newton-Rhapson method.

With the proposed modification by Baraff and Witkin [19], the integration scheme is no longer unconditionally stable, but large time steps are still possible. Compared to most explicit methods, the large time steps compensate the additional computational effort of solving a system of equations and the implicit method leads to considerably faster run times. However, the integration scheme is still only of first-order accuracy and using large time steps, numerical damping occurs that could lead to reduced detail in the final animation. These problems can be approached by using higher-order, preferably multi-step, implicit methods, such as backward differential formula (BDF). Also, exponential integrators, introduced to computer graphics by Michels et al. [77], are a promising approach for cloth, combining stability with accuracy and energy conservation.

2.2.5 Animation Loop

The structure of a basic cloth simulation solver is illustrated in Figure 2.9. With a given triangulated cloth mesh in rest state, the forces per node are calculated based on the material law and the discretized equilibrium equation (Equation 2.91). For time integration using the implicit Euler method, the Jacobian of the forces has to be calculated to build the system of linear equations. This system can be solved, e.g., using the CG method, and the velocities and positions of the mesh nodes can be updated.

Typically, cloth interacts with virtual characters, with its environment, and, due to its low resistance to bending, with itself. To prevent intersections, a robust collision handling is necessary. Especially because of the highly flexible nature of cloth in combination with many contact areas for dressed characters, collision handling is a complex task and, therefore, beyond the scope of this thesis. In the subsequent chapter, collision detection based on k-DOP hierarchies and collision response based on impulses are used. Details of the techniques can be found in the survey article of Teschner et al. [107] and the original papers [31], [61], [92].

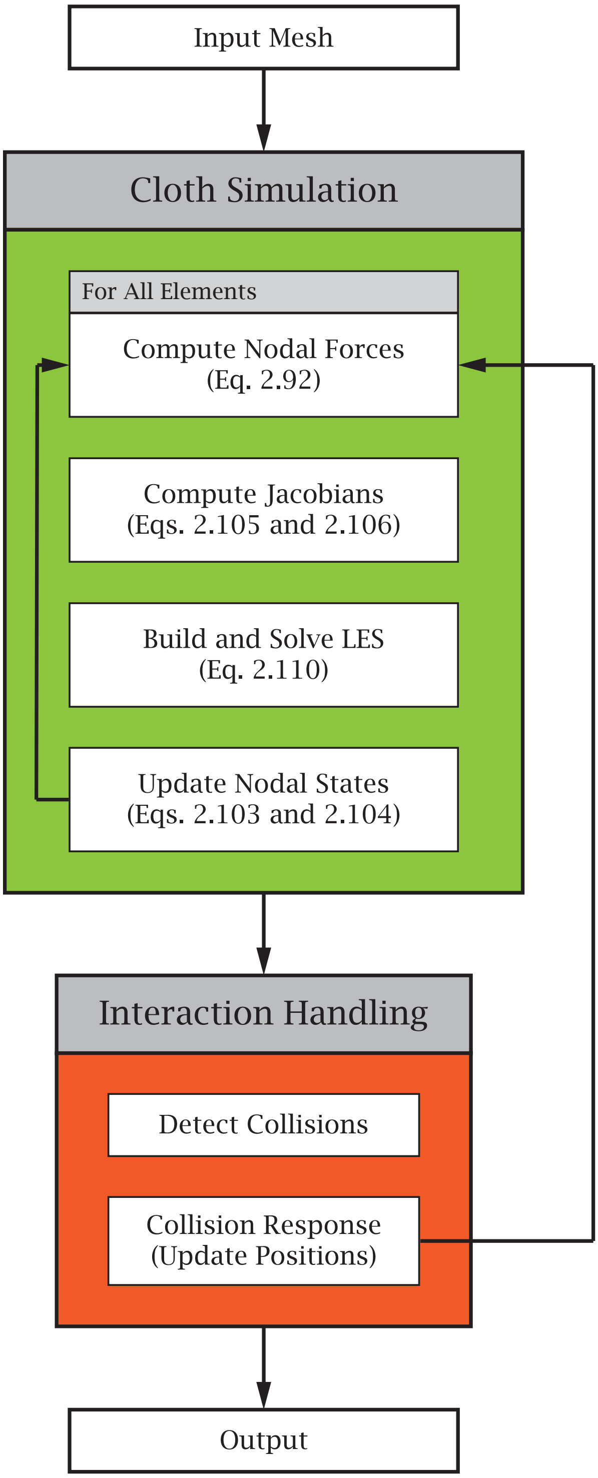


Figure 2.9: Cloth animation loop.

CHAPTER 3

Cloth Animation Retrieval Using a Motion-Shape Signature

3.1 Introduction

State-of-the-art cloth simulation systems allow impressive animations of textiles with a wide range of material parameters that determine the overall behavior of cloth. In particular, the shape of the deformable surface and its change over time depend on many aspects: deformation models, numerical methods and parameters, material properties, and scene setup are just some of the components that influence the simulation result. When creating cloth animations, several ways of achieving a result based on these components are possible. A direct approach is to perform multiple simulations consecutively and in the process, sample parameter spaces manually until a satisfying result is obtained.

In practice, however, there is often a certain goal specified, for instance, by sketches or photographs of key frames, by a 3D scan of the textiles motion, or by a reference animation. The task at hand is then to reproduce the reference by a new cloth simulation aiming to preserve the specified features and transferring the simulation to other environments. This requires a procedure that is able to capture features and determine if the input is matched. The challenge of this task is not only matching the shape of cloth at a specific frame, e.g., an equilibrium state, but also incorporating its temporal evolution. For example, this problem arises in the work by Aliaga et al. [15]: to analyze the perception of cloth with experiments, a video reference has to be matched to generate stimuli. In their case, parameters are mainly set manually to match features, such as size, number, and shape of folds.

The aim of the approach in this chapter is to capture spatio-temporal features of an input example and identify the closest match in a collection of simulations and, thus, retrieve a simulation that exhibits similar characteristics as the input. In doing so, it is possible to find a cloth animation that exhibits the same characteristics as an example. Identifying a close match, however, demands a suitable similarity measure for cloth simulations incorporating spatio-temporal properties. Determining similarity of meshes by visual inspection could suffice for certain applications, but in general, it will result in a time-consuming and error-prone task. For static shapes, there exist many approaches for determining similarity, e.g., using registration or shape descriptors. Extending these techniques to temporal changing shapes, however, is not straightforward, especially if the specific needs of cloth animations have to be met.

This chapter introduces a feature vector that is used as a motion-shape signature to capture the spatio-temporal shape characteristics of cloth and can be used as a space-time similarity measure for physics-based cloth animations. The feature vector not only contains the underlying geometry of the cloth as a mesh, but also measured quantities from the simulation that describe the deformation state and therefore, its shape. The motion-shape signature can be used to retrieve a cloth simulation from a collection of simulations that matches a given input animation. This approach enables a way of goal-oriented creation of cloth animations and directability.

3.2 RELATED WORK

The approach introduced in this chapter does not alter existing simulation models but builds on available or newly generated cloth animations. To create animations, a cloth simulation system based on the models described in Section 2.2 is used.

The presented technique of animation retrieval can be considered as a form of creating goal-oriented animations, which is an active area of research in computer graphics. However, this work is not only positioned in this field, but also overlaps with research in other fields.

Control of Features in Cloth Simulations

Creating goal-oriented cloth animations only by manually adjusting simulation and material parameters is often not feasible in practical scenarios. For this purpose, several methods for directing cloth toward a desired result have been proposed. One approach uses global control forces to direct the motion of cloth by minimizing the distance to a sparse set of key frames, such as in the work of Wojtan et al. [120]. Similarly, Jeon and Choi [58] define control points and use control forces to direct the cloth mesh. A second approach for directing cloth is presented by Schumacher et al. [99]: they extend the work of Martin et al. [75] and use a set of example poses as input for the simulation that represents different rest shapes and, thus, deformation is guided toward these preferred shapes. Opposed to control forces in combination with key frames or control points, the shape of the textile is directed independently from time. An interesting approach that facilitates finding desired simulation parameters is shown by Sigal et al. [100], who present a learned mapping from familiar descriptions for fabric to simulation parameters. All of these approaches present methods that allow control by altering the simulation itself. In this work, similarities are calculated on sets of existing simulations. The aforementioned techniques can be used for the generation of simulations and, therefore, are complementary to this work.

Another category to influence time-dependent features regarding the shape of cloth is augmenting coarse simulations meshes with details, such as fine-scale wrinkles. Cutler et al. [37] synthesize predefined wrinkle patterns on cloth based on stress maps. Rohmer et al. [96] present a method to generate wrinkles along curves determined by the stretch tensor. With the approach of Müller and Chentanez [84], coarse simulation meshes are enhanced with fine wrinkles generated by a secondary parallel simulation. Popa et al. [91] show how folds can be reintroduced in garment capture by extracting them from the video input. Further, example-based methods for synthesizing wrinkles have been proposed [116], [127]. These methods are mainly used to introduce details on existing coarse simulations. Both coarse and enhanced simulations could be used as input for the presented approach. Additionally, the motion-shape signature could be used in the process of creating fine details by example.

Umetani et al. [112] present a system for the simultaneous editing of 2D garment patterns and clothing in 3D that enables interactive preview of drape. Their approach allows artists to interactively design clothing on static characters or objects. In the same context, Bartle et al. [20] present an approach to generate 2D cloth patterns from 3D garment editing. This work, however, targets capturing cloth behavior in dynamic scenarios.

Matching of Temporal Data

In fluid animation, Raveendran et al. [93] present an approach to create 4D space-time surfaces for grid-based fluids and perform registration by a non-rigid iterated closest-point algorithm.

Within motion-capture, similar problem settings exist. One deals with complicated multidimensional temporal data and it is desirable to find or match a certain given motion. This is important for retrieval and analysis of the motion. Usually, the underlying data is skeleton-based, but similar to the proposed approach, a feature vector can be used to for matching sequences as shown in the work of Müller et al. [87]. In contrast to this work, their feature vector contains geometric features connected to body parts of the skeleton.

Temporal data in general appears in many domains, animation is just one of many examples. For data analysis, matching of this kind of data is an important task. From a visualization point of view, techniques and examples can be found in the textbook by Aigner et al. [11]. There are similarities in respect of matching time series of higher dimensional data, but most of these techniques address the comparison of time series in general, mostly measured data, and target determining correlations and gaining insights for predictions.

Identification and Measurement of Surface Features

Describing the shape of an object and calculating similarities between 3D shapes has been widely studied in the fields of object recognition, retrieval, clustering, and classification. The most popular techniques are summarized in surveys, e.g., for content-based shape retrieval [104], non-rigid shape retrieval [68], or shape registration [103]. These methods are often tailored to specific applications, such as object retrieval, and are usually based on static shapes. Temporal datasets, in particular animations, are rarely considered in this context. Similarity measures for temporal data, in particular of deformable objects, have not gained much attention.

One approach that attempts to define shape descriptors for temporal surface sequences is presented by Huang et al. [53]. In their work, classic histogram-based shape descriptors are aligned for time series to a 4D "shape-flow" descriptor with temporal filtering. Their technique is based on existing shape descriptors and targets shape matching of human motion data.

Closely related to the technique of this chapter is the work of Luo et al. [73], who propose doing spatio-temporal segmentation of deforming meshes based on strain. In their approach, strain is used as a binary indicator to identify deformation on a spatial and temporal basis, but is not used as a shape descriptor or similarity measure. Strain alone would not be sufficient as a shape description for cloth, because in-plane deformation is typically far less distinctive than bending deformation.

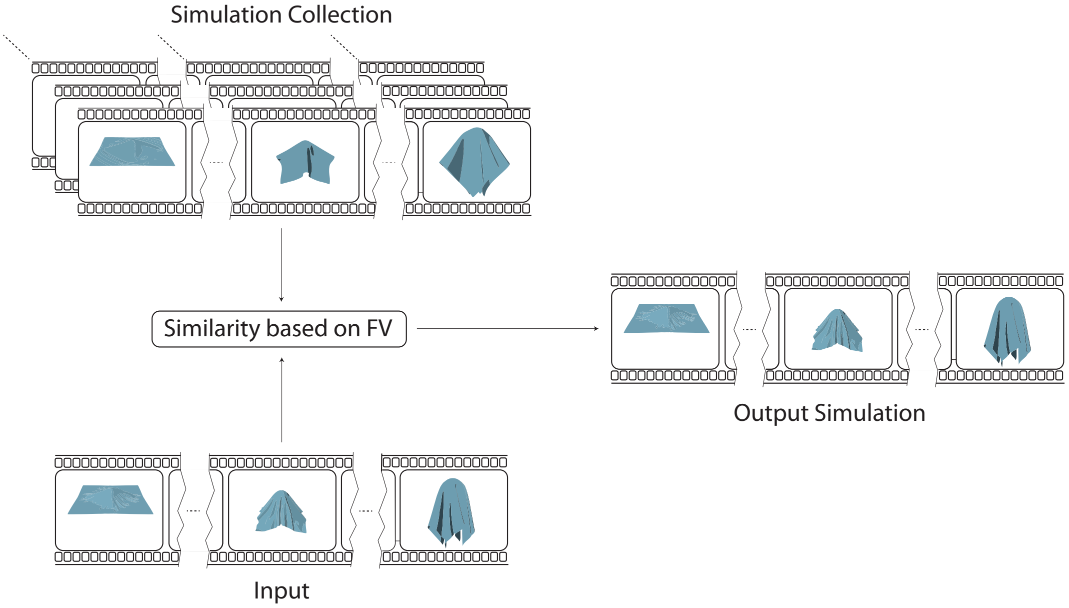


Figure 3.1: Overview of the example-based cloth retrieval method: an input simulation or animation is compared against a collection of other simulations. The comparison is based on a feature vector (FV) that allows one to retrieve a simulation that matches closest to the input and, therefore, exhibits similar spatio-temporal features.

3.3 ANIMATION RETRIEVAL BY EXAMPLE

This section describes the concept of retrieving cloth simulations using an example input simulation aiming to preserve characteristic features. In particular, incorporating the time dependency of those features is of major interest. The basic outline of the approach is depicted in Figure 3.1. The goal is to retrieve an output simulation that exhibits the desired features of an input example from a collection of simulations. This collection can be obtained from different origins, e.g., a simulation database, a simulation ensemble, or by creating (guided) simulations in place. The input for this retrieval process can generally take different forms: For instance, it can be an example simulation that consists of the underlying mesh together with deformation measures such as strain or bending energy. Likewise, a single mesh, key frames, or an animation (sequence of meshes) are other options of example inputs. The focus of this work is on input simulations and animations. In the retrieval process, the similarity of the input to all the elements of the simulation collection is calculated. To describe a simulation and calculate similarities, a feature vector is introduced that incorporates important measures representing the time-dependent shape of cloth.

Below, the data model is described, followed by the proposed concept to calculate similarity of simulations.

3.3.1 Data Model

A cloth simulation is given by a sequence of consecutive simulation states ς_i with T time steps as

$$\mathcal{S}(\mathcal{X}) = \{\varsigma_1, ..., \varsigma_T\} \tag{3.1}$$

and depends on a parameter set for the material and simulation methods, and initial conditions, combined in $\mathcal{X}$. As cloth is modeled as a thin deformable surface, each state ς_i is represented by a discretization of the surface, usually a regular or irregular triangle mesh. Depending on the underlying deformation model, attributes of the cloth simulation are connected to different elements of the mesh, e.g., to vertices, faces, or edges. Such attributes are quantities that represent the deformation state of the cloth such as strain and bending deformation, and their energies respectively.

As the simulation sequence $\mathcal{S}$ can be of any length, it can be split into multiple segments $\mathcal{S}' \subseteq \mathcal{S}$ of varying length. To begin with, it is desirable to have segments showing behavior that is continuous in time. Similar to techniques in video processing where segments are divided by shot boundaries, sequences are split at points where there are large differences in similarity at consecutive frames. This not only helps compare sequences, but it can also be beneficial to process subsequent calculations on smaller segments for computational efficiency.

3.3.2 Similarity Calculation

As described above, the focus is on the comparison of segments $\mathcal{S}'$ and the similarity calculation is processed pairwise. The segments are assumed to be of equal size, i.e., they consist of the same number of simulation states ς. If this is not the case, registration of time curves of attributes and cropping, or dynamic time warping (DTW) of the sequences has to be performed in a preprocessing step.

The similarity of two simulation segments $\mathcal{S}'_a$ and $\mathcal{S}'_b$ is calculated by

$$d(\mathcal{S}'_a, \mathcal{S}'_b) = \sum_{i=0}^{N} ||d(s^i_a, s^i_b)||, \tag{3.2}$$

where s^i_a and s^i_b are single or aggregated subsegments of simulation states, d is a distance metric, and N is the number of simulations states or subsegments.

To represent simulation states, a feature vector is used that holds the simulation states at each simulation step:

$$\mathbf{A} = [a_1^{t_1} a_1^{t_2} \dots a_1^{t_m}, a_2^{t_1} a_2^{t_2} \dots a_2^{t_m}, \dots, a_n^{t_1} a_n^{t_2} \dots a_n^{t_m}]. \tag{3.3}$$

The feature vector contains n attributes for every simulation step t_j in the vector $[t_1 \dots t_m]$. Hence, each segment is represented by an $(m \times n)$-dimensional feature vector. The feature vector is referred to as *motion-shape signature* for a segment or a simulation. Using this signature, the similarity of two segments can be calculated by the distance between their feature vectors $d(\mathbf{A}_{\mathcal{S}'_a}, \mathbf{A}_{\mathcal{S}'_b})$.

3.4 ATTRIBUTES OF THE MOTION-SHAPE SIGNATURE

With the feature vector $\mathbf{A}$ (Equation 3.3), the temporal behavior of cloth should be captured as accurately as possible. Therefore, it is crucial that the attributes a that form the feature vector are tailored to represent characteristic features of cloth shapes, such as planar deformation and wrinkling. Additionally, the attributes or metrics have to fulfill several requirements. It is assumed that the underlying cloth mesh of a simulation is a triangulated surface and does not change in composition, topology, or resolution. The metrics should be independent of translation and rotation of the simulation. Regarding scaling, the value of the metric should scale with the simulation. Another important property of the attributes is that they are not bound to a specific triangulation strategy and are robust against changes in resolution. To meet these demands, measures that account for the two main deformation modes used in modeling cloth, namely planar and bending deformation, are included in the feature vector.

In particular, strain, strain energy, bend deformation, and bending energy are incorporated in the feature vector $\mathbf{A}$. In the following, a brief overview of the used attributes and their calculation is given. In typical mesh-based cloth simulation systems, these features are computed on elements of the mesh: in-plane deformations are computed on triangles, and bending deformations are determined on pairs of triangles sharing an edge. As a combined description of the respective attributes is desirable, the element-wise features are aggregated. Using this spatial aggregation, the shape or deformation description is independent from the mesh triangulation and resolution, and registration of meshes can be avoided. The way of aggregation is also given for the subsequent descriptions of the respective attributes.

At this point, it has to be mentioned that the attributes above can be exported per element directly during simulation, as the respective values are computed anyway in the process. Hence, the attributes can be integrated immediately into the feature vector and do not have to be calculated again in the retrieval process.

3.4.1 Strain

For planar deformations of cloth, strain is used as a deformation measure. Here, linear Cauchy strain ε_C (Equation 2.66) is used and can be computed by the cloth simulation system described in Section 2.2. Of course, it could also be exchanged by the rotationally invariant Green strain (Equation 2.65).

In the discrete setting, it can be calculated on the triangles elements of the given mesh by the linear approximation of Equation 2.87. The strain measure is composed of three scalar values, one value for deformation in weft and warp direction respectively, and one value for shearing.

For each time step in a simulation, strain of the entire cloth mesh is aggregated over all T triangular elements e_i with $i = 1,..,T$. Therefore, at time t_j the strain attribute

$$a_{\varepsilon}^{t_j} = \sum_{e_i} \varepsilon \tag{3.4}$$

is added to the feature vector $\mathbf{A}$. The strain attribute can be calculated separately for the weft, warp, and shear components, or by the norm of the strain vector in Equation 2.79. Here and in the following, the short notation to calculate the sum over triangle elements $e_i \in e_T$ and edges $b_i \in b_M$ of bending elements is used. The terms of the sum depend on the area of the respective element.

3.4.2 Strain Energy

In addition to strain, strain energy is also included as an attribute in the feature vector. For an element e, it is calculated by

$$W_s = \frac{1}{2} A \sigma \varepsilon, \tag{3.5}$$

where A denotes the area of a triangle, and σ is stress. Stress is calculated by the strain–stress relationship given in Equation 2.82, using the elasticity matrix $\boldsymbol{C}$ that describes the material properties.

Similar to the strain attribute, an aggregated value for the strain energy

$$a_{W_s}^{t_j} = \sum_{e_i} W_s \tag{3.6}$$

is stored to the feature vector. The strain energy attribute can also be calculated separately for the weft, warp, and shear components, or by their sum.

3.4.3 Bending Deformation

Bending deformation is generally measured by curvature, respectively its deviation from the rest state. With the discrete models described in Section 2.2.3, the angle θ between the normals of two adjacent triangles sharing an edge of the cloth mesh is measured.

Again, the bending attribute at time t_j for all M bending elements b_i for $i = 1, .., M$ with common edge $\mathbf{e}_i$ is calculated by aggregation:

$$a_\theta^{t_j} = \sum_{b_i} \theta_i ||\mathbf{e}_i||. \tag{3.7}$$

In this case, the edge length $||\mathbf{e}||$ is included for proper scaling.

3.4.4 Bending Energy

As for the planar deformation, bending energy for out-of-plane deformation is also included in the feature vector. As shown for example by Wardetzky et al. [117], bending energy can be expressed based on mean curvatures H with

$$W_b' = \frac{1}{2} \int_S H^2 dA \tag{3.8}$$

over a surface S.

In the discrete case, bending energy for a bending element can be written as

$$W_b = \theta^2 \frac{||\mathbf{e}||}{h_\mathbf{e}}, \tag{3.9}$$

where $||\mathbf{e}||$ is the length of the shared edge of two adjacent triangles, and $h_\mathbf{e}$ denotes the sum of the heights of the two triangles.

For bending energy, the aggregated value that is added to the feature vector is calculated by

$$a_{W_b}^{t_j} = \sum_{b_i} W_b. \tag{3.10}$$

3.4.5 Composition of the Motion-Shape Signature and Similarity

Having stored the aforementioned attributes for a simulation segment $\mathcal{S}'$, the motion-shape signature can be set up. First, the aggregated attributes per simulation step t_j are grouped into sets

$$A_\varepsilon = [a_\varepsilon^{t_1} \dots a_\varepsilon^{t_m}],$$

$$A_{W_s} = [a_{W_s}^{t_1} \dots a_{W_s}^{t_m}],$$

$$A_\theta = [a_\theta^{t_1} \dots a_\theta^{t_m}],$$

$$A_{W_b} = [a_{W_b}^{t_1} \dots a_{W_b}^{t_m}],$$

for strain, strain energy, bend angle, and bend energy, respectively. At this point, it is assumed that there are no discontinuities of the temporal evolution of the features in the considered simulation segment $\mathcal{S}'$—otherwise the segment should be split. With this assumption and the observation that the change of features between time steps is relatively small, temporal aggregation of a small number of subsequent frames is also possible. In this case, consecutive entries of the attribute sets are grouped in equally sized subsets and averaged, which then results in the new attribute sets A'_ε, A'_{W_s}, A'_θ, and A'_{W_b}. As a result, the length of the feature vector can be reduced. Then, the attribute sets are grouped together in the actual feature vector $\mathbf{A}$.

In a final step, it is possible to assign user-defined weights to each attribute set in the feature vector to be able to accentuate on specific attributes if needed. For example, there are many cloth materials that are almost inextensible. In this case, the shape is dominated by the bending deformation mode. If an elastic fabric is attached to a collision object, stretch properties should be considered at least equally in the feature vector. It follows that the specific motion-shape signature for cloth animations results in the final feature vector (introduced in Equation 3.3) as

$$\mathbf{A} = [w_\varepsilon A'_\varepsilon, w_{W_s} A'_{W_s}, w_\theta A'_\theta, w_{W_b} A'_{W_b}], \tag{3.11}$$

with weights w_j. The proposed attributes in the feature vector $\mathbf{F}$ are used paired: strain and strain energy are features to describe planar shapes, bending and bending energy measure curvature of shapes. Hence, weights are usually set pairwise with $w_\varepsilon = w_{W_s}$ and $w_\theta = w_{W_b}$.

After the motion-shape signature has been put together for existing cloth animations, it is possible to determine the similarity of simulation segments (Equation 3.2). Given two simulation sequences S_1 and S_2 of the same length m, the similarity is calculated by their distance $d(\mathbf{A}_{S_1}, \mathbf{A}_{S_2})$. In doing so, the distances are normalized for each attribute individually to be independent of units and scale. For a time step t_j from the vector $[t_1 \dots t_m]$ of the two sequences, the similarity calculation is illustrated in Figure 3.2.

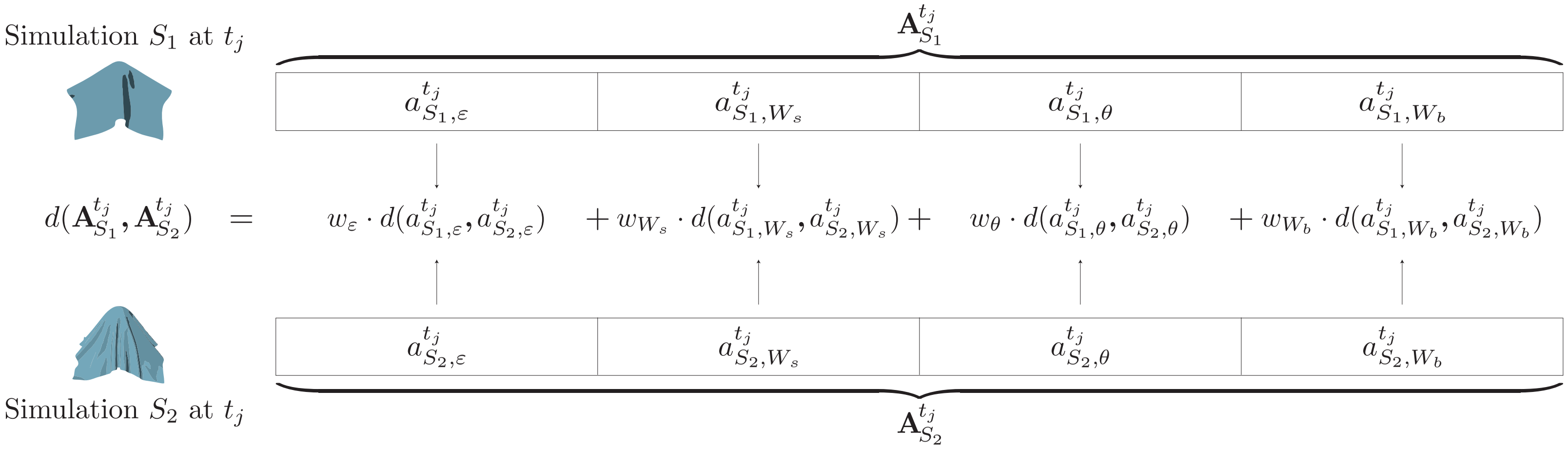

Simulation S_1 at t_j
$\mathbf{A}^{t_j}_{S_1}$
$a^{t_j}_{S_1,\varepsilon}$
$a^{t_j}_{S_1,W_s}$
$a^{t_j}_{S_1,\theta}$
$a^{t_j}_{S_1,W_b}$
$d(\mathbf{A}^{t_j}_{S_1}, \mathbf{A}^{t_j}_{S_2}) = w_\varepsilon \cdot d(a^{t_j}_{S_1,\varepsilon}, a^{t_j}_{S_2,\varepsilon}) + w_{W_s} \cdot d(a^{t_j}_{S_1,W_s}, a^{t_j}_{S_2,W_s}) + w_\theta \cdot d(a^{t_j}_{S_1,\theta}, a^{t_j}_{S_2,\theta}) + w_{W_b} \cdot d(a^{t_j}_{S_1,W_b}, a^{t_j}_{S_2,W_b})$
$a^{t_j}_{S_2,\varepsilon}$
$a^{t_j}_{S_2,W_s}$
$a^{t_j}_{S_2,\theta}$
$a^{t_j}_{S_2,W_b}$
Simulation S_2 at t_j
$\mathbf{A}^{t_j}_{S_2}$

Figure 3.2: Illustration of the similarity calculation of two given simulation sequences S_1 and S_2 at time t_j. The distance between the individual attributes of the respective feature vectors is calculated and weighted according the user-defined weights w.

3.5 EXPERIMENTAL RESULTS

In order to analyze and test the proposed methods of the preceding sections, a series of experiments for different purposes is conducted. In the following, all cloth simulations were generated using a co-rotated finite element as described in Section 2.2 for planar deformations and the model of Bridson et al. [32] (see Section 2.2.3) for bending deformation. For time integration, a first-order implicit Euler method as described in Section 2.2.4 is used with a fixed time step of $\Delta t = 0.001\,\mathrm{s}$. Whilst performing simulations, the attributes of the feature vector described in Section 3.4.5 are calculated during runtime.

First, some basic examples of the motion-shape signature for simple cloth shapes are given to present the properties of the shape metrics. Then, results using the signature for calculating similarity of static shapes of cloth are shown. After that, an example of the retrieval of a simulation from a collection by an input example is given where the shape of a collision object is changed. Also, the presented method can be used to retrieve a simulation by the captured geometry of a real-world cloth. In a last scenario, cloth animation retrieval of a dressed character is performed using an input simulation with considerably different material parameters than the simulation collection.

3.5.1 Properties of the Shape Metrics

In Figure 3.3, examples of the motion-shape signature for three simple cloth animations are given to illustrate its basic properties. For the three examples, a square regular mesh with 2048 faces is used. In the figures, fixed edges and vertices are depicted in black, animated edges vertices are depicted in yellow. First, the flat piece of fabric is stretched with a relatively high Poisson coefficient without the influence of gravity. As expected, in-plane deformation is the dominating effect, which can be seen in the strain component of the motion-shape signature in the first diagram column of row one in Figure 3.3. An almost linear behavior can be observed with increasing stretch. For the second example shown in Figure 3.3, the cloth is bent by a cylindrical collision object to observe bending properties. In this case, out-of-plane deformation is much stronger than in-plane deformation, which is captured by the bending metrics, illustrated with the aggregated crease angle component. Bending deformation increases as the fabric drapes over the object and after some oscillation, an equilibrium state is established. In the third example, the piece of fabric is pinned at three corners, one moving outward, stretching the cloth. Strong in-plane and bending deformation can be observed that is represented in the strain and crease angle components of the feature vector.

As explained in Section 3.4, spatial aggregation is performed for each time step for the respective attributes in the feature vector. In the second and

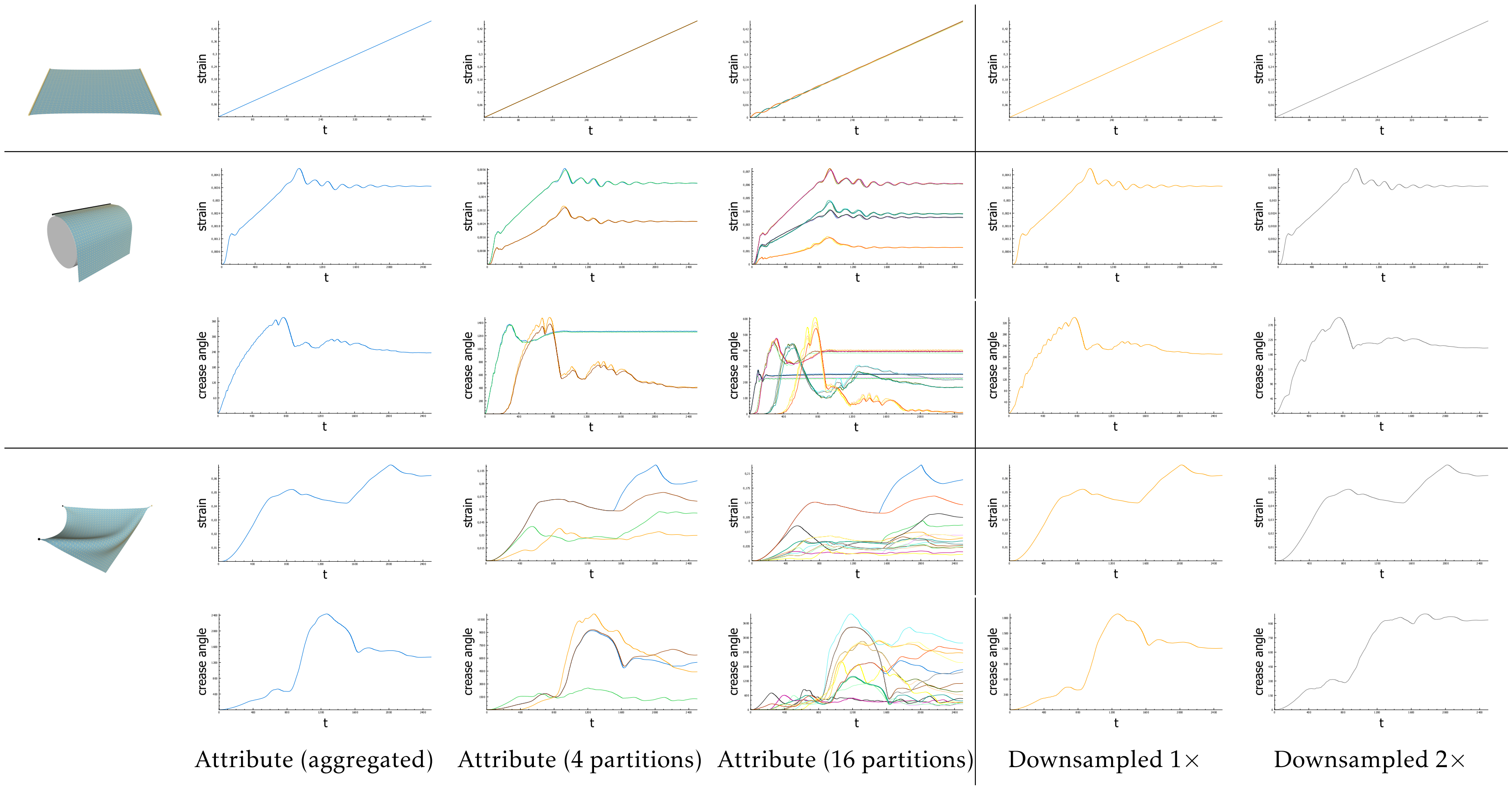


Figure 3.3: Three basic cloth simulations (2048 triangles), illustrated by a representative frame on the left. In the first diagram column, aggregated attributes of the respective feature vectors are shown. In the next column, spatial partitioning is performed on the cloth mesh with four equally sized partitions, followed by 16 partitions in the third column. The two columns on the right represent the attributes for a downsampled animation with 512 triangles and 128 triangles, respectively.

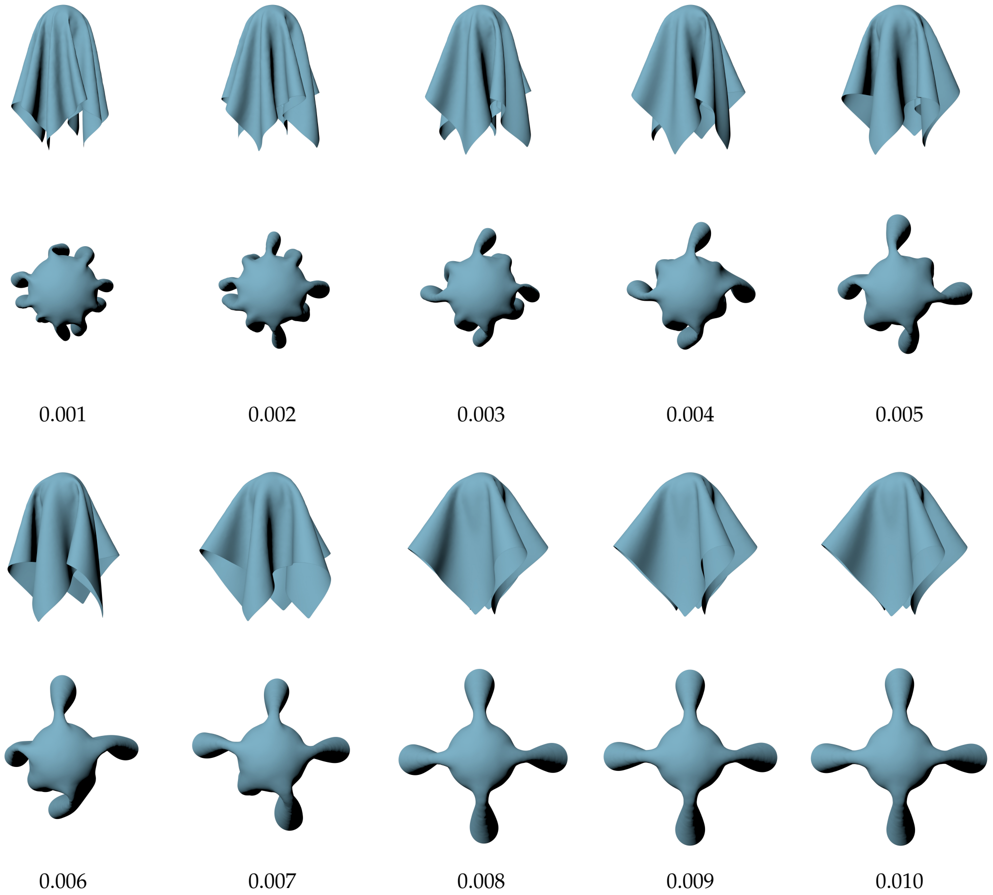
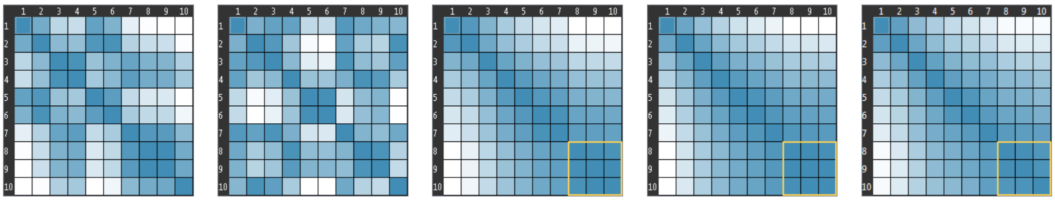


Figure 3.4: Final equilibrium states of a piece of cloth draped over a sphere from side view and top view with the respective bending stiffness parameters k_b used for calculating similarities of static shapes. The three last shapes are considered as most similar ($k_b = 0.008 - 0.010$).

third column of Figure 3.3, it is demonstrated how partitioning of the mesh affects the components of the feature vector. While there is no difference if the deformation is homogeneous across the mesh as in the first example, varying behavior of the individual partitions can be observed for the other two examples. Hence, it would be difficult to calculate similarity between cloth simulations without spatial aggregation as, e.g., spatial matching would be a requirement.

Further, mesh resolution has only minor influence on the attributes. The components for in-plane deformation behave almost identical for downsampled simulations, as it can be observed in the two columns on the right of Figure 3.3. Especially for bending deformation, only slight differences are noticeable in 1× decreased resolution. Reduced to 128 faces, the differences increase, but



strain strain energy bending angle bending energy combined

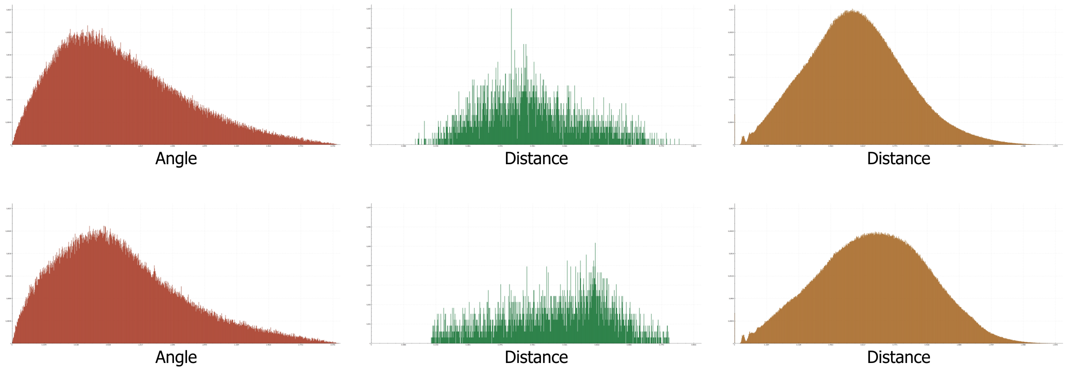
Figure 3.5: Similarity matrices for the static shapes as shown in Figure 3.4 using the motion-shape signature. The first four matrices show the individual components of the feature vector. The rightmost matrix combines the four attributes with weights $(0.05, 0.05, 0.45, 0.45)$. The highlighted blocks correspond to the three last shapes in Figure 3.4 ($k_b = 0.008 - 0.010$).

this resolution is no longer usable in practical scenarios. Still, as long as the same overall shape of the fabric is equal, the components of the feature vector exhibit comparable properties and animation retrieval can be performed independent from resolution or triangulation.

3.5.2 Similarity of Static Shapes

In this experiment, similarity measurements for a set of static cloth shapes are examined. A set of ten simulations of a cloth draped over a sphere is executed with a fixed parameter set except for a varying bending stiffness parameter k_b. Young's modulus is set to $E_u = E_v = 500\,N/m$ for both material directions u and v, shear modulus is set to $G = 50\,N/m$, and the cloths' mass is $0.23\,kg/m^2$. The simulation is carried out until the cloth reaches an equilibrium state and the last simulation state as shown in Figure 3.4 is used for further analysis.

The similarity of the ten different shapes is calculated using the Euclidean distance between the respective feature vectors presented in Section 3.4. Similarity matrices for the motion-shape signature with the attributes described in Section 3.4 are shown in Figure 3.5. The first four images represent similarity matrices for the individual attributes strain, strain energy, bending angle, and bending energy. In the last image, similarity is calculated with the combined feature vector given in Equation 3.11. As this scenario is an example of cloth, where the out-of-plane deformation is much higher than in-plane deformation, it can be seen that the attributes representing the planar features provide only little insight opposed to the attributes representing bending deformation. Therefore, the weights $(w_\varepsilon, w_{W_s}, w_\theta, w_{W_b})$ are set to $(0.05, 0.05, 0.45, 0.45)$ to account for the deformation modes.



A3 distribution D1 distribution D2 distribution

Figure 3.6: Histograms for three different classic shape descriptors and two parameter sets of the shapes shown in Figure 3.4. A3 distribution is shown in red, D1 distribution in green, and D2 distribution in brown. In the top row, the shape for the smallest bending parameter ($k_b = 0.001$) is analyzed, in the bottom row the largest ($k_b = 0.010$).

3.5.3 Comparison to Classic Shape Descriptors

In comparison to the proposed signature, classic shape descriptors are also used for calculating similarities of static cloth shapes as shown in Figure 3.4. The purpose of this comparison is not about whether the proposed descriptor is able to outperform classic methods, but if it leads to at least similar results for a static shape, but with the advantage to be easily extensible to simulation sequences and being particularly suited for cloth.

Three shape descriptors based on distributions are used as described in the article by Osada et al. [89]:

- A3 distribution: the angles between three nodes of the cloth mesh.
- D1 distribution: distances between the centroid and the nodes of the cloth mesh.
- D2 distribution: distances between 2 random nodes of the cloth mesh.

Examples of histograms of the respective shape distributions applied to the draping scenario are given in Figure 3.6. In the top row, histograms for the lowest value for bending stiffness are shown and in the bottom row, histograms for the highest value which correspond to the most left and most right shape in Figure 3.4.

To calculate similarity between the ten equilibrium states, the distance between their histograms is calculated. Three common distance measures for comparing histograms are used: Euclidean distance, Chi-Square, and Bhattacharyya distance. Results are shown as similarity matrices in Figure 3.7 with the given distance measures from left to right. In this illustration, the lighter the color

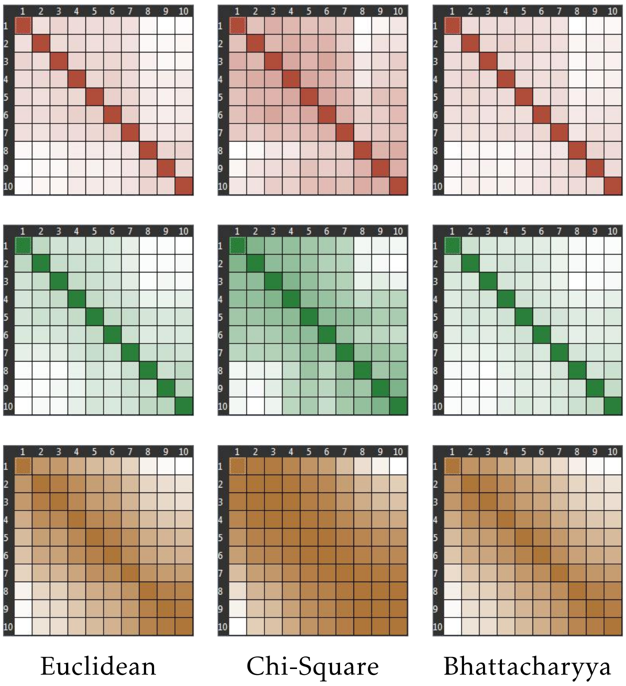


Figure 3.7: Similarity matrices for three different shape descriptors and three distance metrics for the shapes shown in Figure 3.4. The color is set for the shape descriptors as described in Figure 3.6, and distance metrics are shown in columns.

of a cell, the higher the dissimilarity. Most noticeable is that a group of the three last shapes can be identified in the similarity matrices, particularly in case of the D3 distribution using Euclidean and Bhattacharyya distance. For both cases, this conforms to Figure 3.4 by visual inspection and to the results of the motion-shape signature (Section 3.5.2).

3.5.4 Simulation Retrieval with Changing Collision Object

In the second example, animation retrieval using an input simulation and a collection of simulations is performed as illustrated in the introductory overview in Figure 3.1. The input example is a piece of cloth consisting of 6326 faces draped over a rectangular collision object with a given parameter set. The simulation sequence lasts $2.5s$ and the parameters for Young's and shear moduli are $(400, 400, 50)\,N/m$ and $k_b = 0.003$ for bending stiffness.

Now, the collision object is changed to a cylindrical shape, and the goal is to find a simulation that exhibits a similar spatio-temporal behavior as with the rectangular shaped collision object. To this end, a simulation collection of about 40 simulations is generated by systematically sampling the parameter

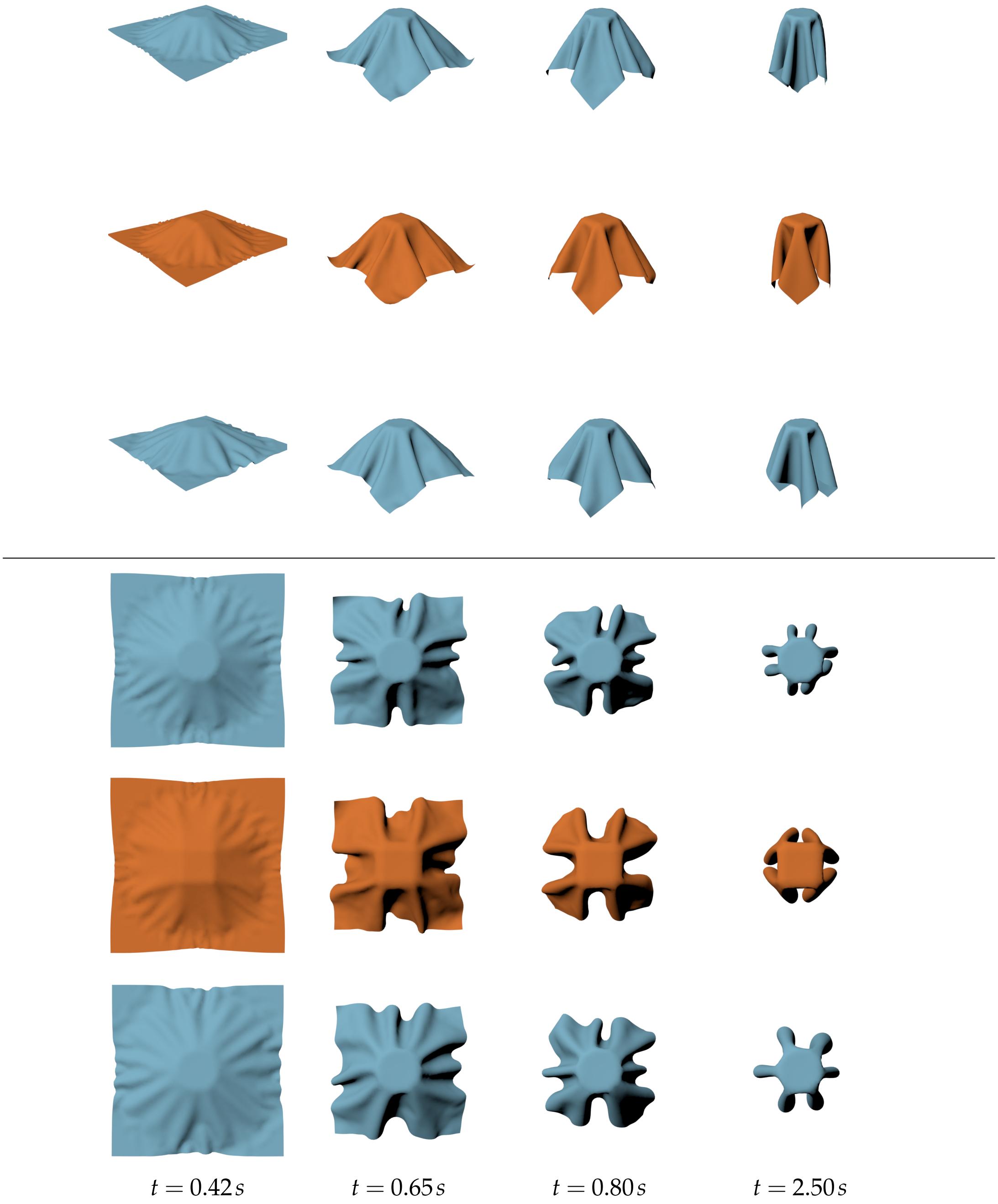


Figure 3.8: A typical retrieval scenario: the input simulation involving a rectangular collision object is shown in orange color. A collection of simulations is created with a cylindrical collision object using varying parameters. The result above the input has the same parameters as the input, but the original features, such as the shape of the wrinkles, are not preserved. In the row below the input, the closest matching simulation of the collection is shown.

Table 3.1: Similarities of the input simulation to a chosen subset of the simulation collection and the respective parameters for simulation retrieval with changing collision object described in Section 3.5.4 using the set of weights $(0.0, 0.0, 0.5, 0.5)$. The lower the value, the closer is the match. The highlighted row is the best match and is shown in Figure 3.8.

E (N/m)	G (N/m)	k_b	similarity
400	50	0.003	0.719
400	50	0.005	0.347
400	50	0.007	0.172
400	100	0.003	0.630
400	100	0.005	0.360
400	100	0.007	0.222
400	150	0.003	1.000
400	150	0.005	0.264
400	150	0.007	0.225
400	200	0.003	0.963
400	200	0.005	0.162
400	200	0.007	0.153
400	250	0.003	0.828
400	250	0.005	0.441
400	250	0.007	0.161

space of the cloth. At this point, parameters are varied only for isotropic stretch resistance, shear resistance, and bending resistance. Young's modulus varies from $300\,N/m$ to $500\,N/m$ for both material directions, shear modulus is in a range between $50\,N/m$ and $250\,N/m$, and the bending stiffness parameter k_b is between 0.001 and 0.010. A subset of the used parameter combinations is shown in Table 3.1. The input is compared with all entries of the simulation collection using the motion-shape signature using the weights $w_\varepsilon = w_{W_s} = 0.0$ and $w_\theta = w_{W_b} = 0.5$ and subsegments with the length of 100 frames to retrieve the closest match.

In Figure 3.8, selected frames of the input simulation (orange colored) are shown from different perspectives. The top row shows the same frames of the simulation with the different collision object, but the same parameter set as the input example. In this case, a different impression of the material is observable. Mainly, this is due the higher amount of wrinkles evolving with the same parameters, especially noticeable in the second and third given frame.

The bottom rows of the different perspectives in Figure 3.8 show frames of the simulations of the closest match, having the parameters $(400, 400, 200)\,N/m$ for The shapes at the given time frames are more similar to the input simulation with this altered material parameter. In particular, the coarser features are preserved better. Another important aspect is that the temporal behavior over the different time frames is maintained.

A subset of the parameter combinations of the simulation collection along with the similarity calculations for this example is shown in Table 3.1. The highlighted row in the table corresponds to the bottom row for each perspective in Figure 3.8, respectively.

3.5.5 Retrieval of Simulations for Captured Input

In this example, the input for the animation retrieval technique is not a simulation as before, but a captured real-world cloth. The captured geometry is provided by White et al. using their method for capturing garment [119]. The garment mesh consists of approx. 2600 faces and 26 frames of the sequence are used. In Figure 3.9, the rendered input geometry is shown in the first row (orange) from two different camera positions. The scene is manually reverse-engineered and the starting position of a cloth estimated using a mesh with similar resolution (approx. 2500 faces). 10 simulations are performed with varying bending parameters, and the parameter set $(800, 800, 200)\,N/m$ for Young's and shear moduli, respectively. The input is a sequence of meshes, hence comparison can only be done on the bend angle attribute set A'_θ in the feature vector of Equation 3.11. As the remaining attribute sets cannot be calculated for captured input, they are omitted. In addition, the time curves of the bending angle and the time frame of the animations are registered manually to ensure temporal coherence. Using a single segment, the closest match is retrieved with a bending stiffness parameter $k_b = 0.007$. This is shown in the lower row in Figure 3.9.

This example shows that it is also possible to achieve a satisfying matching even with a sparse input and using only a subset of the attributes in the feature vector. Using advanced garment capturing techniques and sampling a large space of material parameters together with varying initial conditions could also lead to very promising results for determining cloth parameters for captured input.

3.5.6 Simulation Retrieval with Animated Character

In the last example, simulation retrieval for an animated character dressed with a simulated skirt is performed (Figure 3.10). The cloth mesh consists of 2926 faces and $2s$ of a simulation are examined, which roughly corresponds to two steps of the animated character. A collection of simulations consisting of 8 simulation runs is created using parameters for Young's and shear moduli $(200, 200, 50)\,N/m$ and varying bending stiffness from $k_b = 0.0002$ to $k_b = 0.01$. As input for the retrieval procedure, a simulation with considerable different

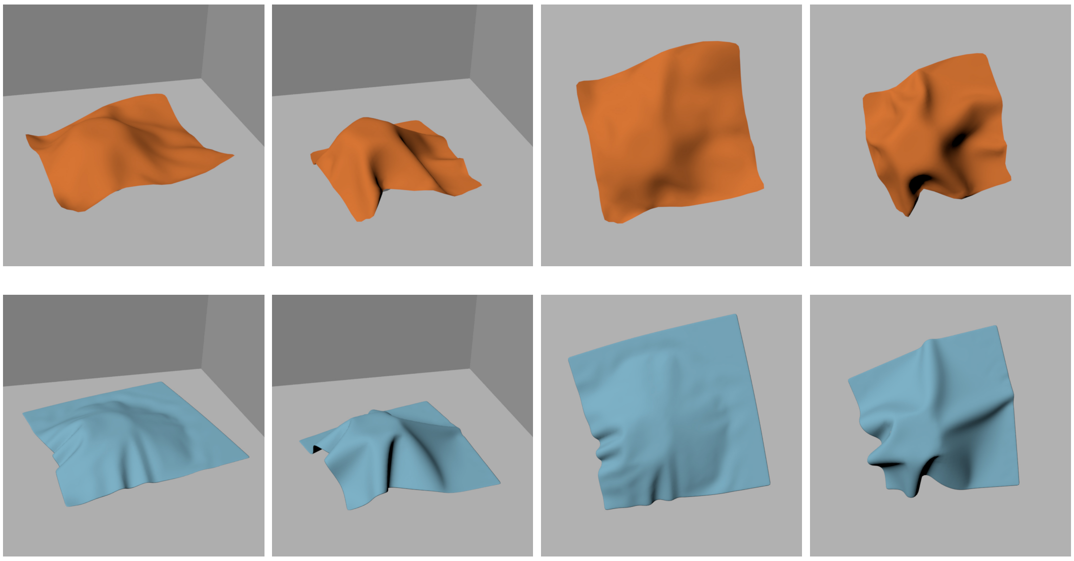


Figure 3.9: Top row: a captured garment established by White et al. [119] is used as an input example for the matching procedure. The most similar result of a set of 10 simulations is shown in the second row.

material parameters is used: stretch stiffness is set to $(400,400,250)\,N/m$, bending stiffness is set to $k_b = 0.0007$, and the mass of the cloth is almost doubled. Further, the weights of the motion-shape signature are set equally to $(0.25,0.25,0.25,0.25)$.

In the original collection, the simulation with $k_b = 0.0004$ is determined as the closest match. In Figure 3.10, several viewpoints at four different time steps are shown for representative simulation runs: for each viewpoint, the input simulation is shown in orange, on its left is the closest match, on the outer left is the simulation with the smallest bending stiffness, and on the outer right the simulation with the largest bending stiffness of the collection. In this example as well, the retrieval match exhibits similar characteristics throughout the whole animation. Important features such as wrinkles are similar to the input. At the beginning of the animation (Figure 3.10, first row), only small differences in the shape of the skirt are noticeable with the softer materials. As expected, a high bending coefficient leads to a considerable different shape. During the first step of the character (Figure 3.10, second row), folds and wrinkles arise and it can be seen that the input and the corresponding match exhibit similar patterns, e.g., located at the waist of the character and the lower part of the skirt. In the subsequent frames, characteristic shapes on the back and front of the skirt (Figure 3.10, lower two row) match the input, too.

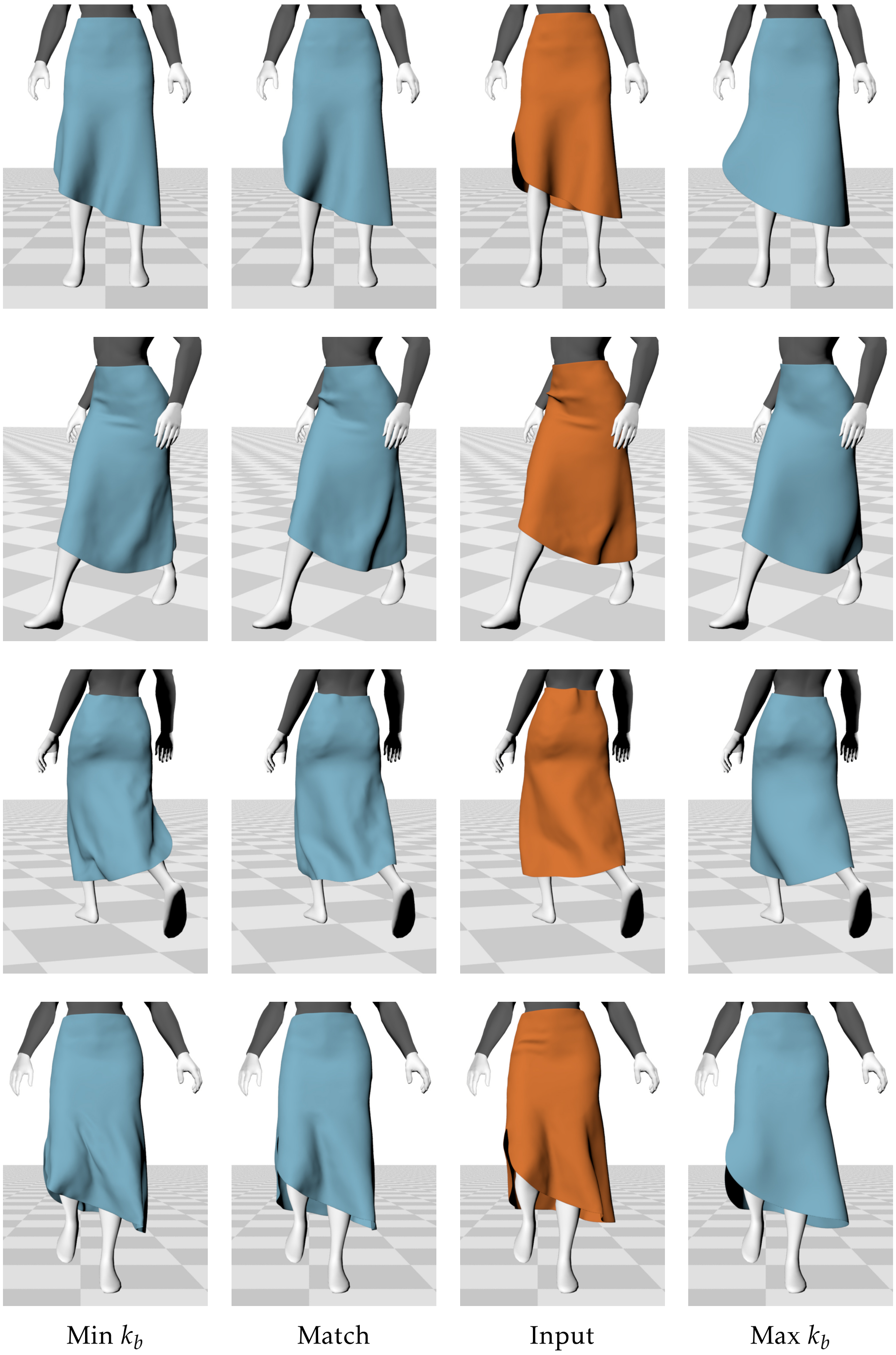


Figure 3.10: Four frames of an example of cloth animation retrieval presented in Section 3.5.6. For each frame, the input simulation is represented in orange in the image set. The simulation of the collection with the lowest bending stiffness is shown at the outer left, followed by the closest match of the signature. At the outer right, the simulation with the largest bending stiffness of the simulation collection is shown. Especially during walking, the matching simulation exhibits similar patterns as the input simulation.

3.6 DISCUSSION

The presented motion-shape signature and its application to animation retrieval is one step in the development of spatio-temporal descriptors, which has been given little attention. Although good results for the given animation retrieval scenarios could be achieved, there are several issues regarding limitations of the approach and relating to future research directions that have to be addressed.

With the presented signature, multiple metrics are incorporated to make sure that the different deformation modes of cloth are captured. Therefore, the signature is a hybrid classifier. To achieve the best possible results, suitable weights are assigned to the individual attributes. In scenarios where the cloth is able to move freely as in the draping examples, the characteristics of most cloth materials exhibiting comparatively low bending resistance is most prominent, and, therefore, it is obvious that the weight for bending measures has to be increased. This situation, however, changes if the cloth is partially pinned or strongly interacting with (animated) objects as it is the case for dressed characters. A possible drawback of this approach is that the weights w for the respective attributes in the feature vector (Equation 3.11) are determined manually. This often requires fine tuning, which can be a time-consuming task and, in some cases, might not be feasible. A possible solution for this could be to analyze the complete animations in a preprocessing step to find out the predominant deformation modes and set the weights automatically and even temporally adaptively. Moreover, a machine learning approach to determine the attribute weights, possibly in a visual analytics environment to allow for user interaction, is considered a very promising research direction as an improvement and extension to the presented approach.

Another point to be discussed are the possible attributes that are incorporated in the feature vector. As measuring similarity of cloth animations is the main focus, the signature is designed specifically for this application. The four attributes presented in Section 3.4 are chosen for several reasons: the attributes are considered as intuitive, as they directly represent geometric and physical properties. However, a validation of the signature is not provided. For this, either a benchmark with an animation database or a known ground truth would be required. The topic of time-dependent signatures in the context of deformable objects, however, has only gained little attention and there is no community-driven data source yet. Also, a generally valid ground truth for this kind of animations might not exist. In contrast to the presented approach, it could also be possible to concatenate classic shape descriptors in the feature vector. One major advantage of the presented approach is that the attributes used in the motion-shape signature can be directly exported from the cloth simulations and, therefore, do not have to be calculated separately, which, especially for the dynamic case, saves a significant amount of computational cost. Computing shape distributions or similar techniques for animations of several hundred frames containing thousands of data points each, and calculating their distances, involves much time and additional effort. In summary, there could be attributes that lead to similar or even better results, but with the proposed metrics, good results can be achieved in combination with an intuitive usage and computational efficiency. Of course, replacing or extending the metrics is an interesting and challenging topic for future work.

As described in Section 3.4, spatial aggregation for the deformation measures is performed to obtain a global measure throughout the cloth mesh. While this is convenient for relatively small pieces of cloth with a fairly homogeneous spatial distribution of the deformations, large pieces of fabric or deformations that occur only at a small area of a garment could require a more sophisticated solution.

An aspect that is closely connected to validation and weighting of the metrics is the question how to verify the similarity. At this point, this is done by visual inspection, but a quantitative analysis would be preferable. However, given that cloth involves highly deformed surfaces, geometric matching of static shapes alone is a challenging task, e.g., by registration. Finding a balance between the similarity of the overall global shape and preserving fine local details will require a sophisticated method. Including dynamics will make the task even more difficult and is an important research direction.

A separate, but closely related topic to this work is the generation of cloth simulations. If there is no collection of animations given, simulations have to be performed to sample the parameter space or phase space of simulations. Sampling the phase space of cloth [60] or its parameter space requires a huge amount of computational resources and still, could be an unfeasible task. A more elaborate solution could be an optimization approach that uses the proposed similarity measure to create the desired collection of animations.

3.7 SUMMARY

In this chapter, a novel approach for animation retrieval for cloth animations was presented. A motion-shape signature represented by a feature vector has been introduced that allows for the calculation spatio-temporal similarity of cloth simulations. Using the motion-shape signature, it is possible to retrieve simulations from collections that exhibit similar behavior as the input example. The method can capture temporal cloth behavior and find suitable matches for exchanged collision objects and modified material parameters. With the proposed approach, a novel way of creating goal-oriented cloth animations has been shown that has not been considered before. Additionally, it is also possible to retrieve a simulation using captured cloth as an input. Even with purely geometric input data, matching is possible and useful results can be achieved.

CHAPTER 4

Boundary Handling at Cloth-Fluid Contact

4.1 INTRODUCTION

In high-quality digital productions, physically-based simulated cloth has become common practice, e.g., to enhance the appearance of characters. It occurs in many scenarios that there is interaction with fluids in the environment or even wetting of the cloth. In this chapter, the interaction and contact of infinitesimally thin deformable objects, like cloth or thin shells, with fluids is addressed. The focus is on state-of-the-art cloth simulation systems that use mesh representations of the cloth surfaces and on particle-based methods, such as SPH, to simulate fluid dynamics.

This chapter presents an approach to overcome the difficulties of modeling the interaction and contact of the respective simulation approaches. Regarding the coupling of cloth and particle-based fluids, the special characteristic of cloth being represented as two-manifold surfaces that have no inside and outside, requires particular attention. During the interaction process, it is necessary to prevent the fluid particles to move from one side of the cloth surface to the other, often referred to as leaking. Once a particle has moved to the wrong side of the cloth mesh, it is very difficult to correct the situation even in postprocessing. Additionally, boundary conditions that characterize the fluid's movement at the contact area have to be modeled to represent the specific properties of the fluid and cloth. In this context, the varying structure of textiles characterized by the material of threads and the weaving pattern is an important aspect for the coupling of cloth with fluids. Depending on the specific properties of the fabric and the fluid, different degrees of wetting can be observed and the cloth can absorb the fluid that spreads inside the fabric. If these processes occur, noticeable differences at contact will appear.

Having dealt with the interaction, another problem arises in the simulation pipeline: In many particle-based fluid systems, an MC [71] based surface reconstruction pipeline is used to create a polygon mesh of the fluid surface. In the reconstruction process, the calculated surface mesh might intersect the surface mesh of close objects. While this is easily fixed with volumetric objects in postprocessing, 2D manifolds are more challenging. Especially in case of highly flexible materials like cloth that result in folds and wrinkles during the simulation, extensive intersections with the generated fluid surface occur. These material intersections do not only produce implausible visual artifacts but can also cause difficulties in rendering liquid materials, like water, when generating images with a raytracing-based renderer.

This chapter introduces an approach to treat two main difficulties at contact between cloth and particle-based fluids. The first part demonstrates how particle-based fluids can be coupled with thin deformable surfaces using repulsion forces and a continuous intersection test that robustly prevents the fluid leaking through the surface. With continuous intersection handling, two-way coupling is achieved that prevents leaking and is not bound to small time steps. In addition to the separation in normal direction, a boundary condition is employed that affects the tangential component of the velocities, thus modeling the no-slip boundary condition. Further, tangential boundary conditions at the cloth-fluid interface for different material properties that exceed the standard slip or no-slip conditions are discussed.

In this context, it is also considered that cloth can absorb fluid, which can lead to changing boundary conditions depending on how much fluid is inside the fabric, for example if it is dry, damp, or wet. To this end, the absorption and diffusion model of a previously presented approach is incorporated [5]. The presented methods are designed to be independent of the internal specifics of the simulation systems so that they can be integrated in common simulation environments. Furthermore, differences to previous approaches using boundary particles are discussed.

In the second part, a method to reconstruct the surface of the particle-based fluid at complex boundaries in an MC-based pipeline is presented. With the proposed method, it is possible to extract an intersection-free surface mesh at cloth-fluid contact. Also, details are given about the reconstruction of surfaces at contact considering that the fluid can wet the cloth.

4.2 RELATED WORK

The methods in the following section are designed to be applicable to the combination of any mesh-based cloth and particle-based fluid simulation method. In this thesis, the interaction builds on the well-established models for cloth and SPH-based fluids that are described in Section 2.2 and Section 2.1, respectively.

Rigid-fluid coupling for particle-based fluids is often achieved by transforming the solid to a particle representation. Monaghan [80] replaces boundaries with particles and avoids penetrations by applying a repulsion force between the particles. Becker et al. [24] use direct forcing to achieve two-way rigid-fluid coupling with particle-sampled rigid bodies. This approach is extended by Ihmsen et al. [54] by including boundary particles in the fluid's SPH computation. Akinci et al. [14] present a method for the two-way coupling of SPH-based fluids with rigid bodies represented with particles. They include the rigid body particles in the SPH computation to handle density discontinuities and their approach also considers one-layered particle objects. To account for errors in the density estimation at free surfaces and to obtain a realistic cohesion behavior of SPH-based liquids at solid boundaries, Schechter and Bridson [97] introduce a ghost particle method for static solids. In addition to ghost particles in the solid, boundary particles in the air located at the free surfaces of the fluid are added that contribute to the SPH computations. Sampling boundaries with particles can also lead to fluctuations of density in planar regions addressed by Band et al. [18], who propose a boundary representation based on moving least squares (MLS).

To simplify the interaction of fluids and solids, Keiser et al. [59] present a unified particle model for the interaction and phase change of objects. Elastic forces are calculated using MLS. Solenthaler et al. [102] extend this approach by developing a purely SPH-based particle model that also does not need an interface for coupling. This model is modified by Lenaerts and Dutré [67] to cope with thin objects like cloth. However, this approach is limited because it does not allow material stretching for cloth, just bending. Their work adds explicit collision handling to avoid leaking.

For deformable objects, Müller et al. [86] place virtual boundary particles on the surface of the deformable object and use a Lennard-Jones-like force to model repulsion and adhesion. However, their approach requires small time steps to ensure that particles do not penetrate the solid. To overcome leaking issues with boundary particles, Akinci et al. [13] extend their rigid-fluid coupling approach [14] for elastic objects and introduce an adaptive boundary sampling scheme to avoid over- and undersampling of meshes at deformation. Yang et al. [121] generate proxy-particles during runtime at contact points and apply coupling forces with the direct forcing approach of

Becker et al. [24]. Like all methods that rely on inter-particle interaction, the contact distance depends on the particle radii.

A different approach for rigid body interaction is presented by Koschier and Bender [63], who use a discretized density map based on a distance field to represent boundaries. In contrast to particle sampling, this results in smooth boundaries and the density field is also be incorporated in the SPH computation.

Du et al. [39] present an approach for the bidirectional interaction of cloth and fluids without boundary particles, similar to the approach in this chapter. They also use a continuous intersection test to ensure that fluid particles do not penetrate the cloth, however, they use elastic impulses to separate the objects and only consider pure slip boundary conditions.

Collision handling methods are related to the presented boundary handling technique. The state-of-the-art report by Teschner et al. [107] is recommended for a comprehensive overview.

There are several approaches for the surface reconstruction for particle-based fluids. The main challenge in the development of these techniques is to overcome the bumps of classical blobbies and to generate a smooth surface that represents the fluid in a visually plausible and appealing manner, despite irregular particle distributions. Müller et al. [83] represent the surface as the isosurface of a density field of the particles. Although this approach results in fewer bumps on the surface, they are still clearly visible. The approach of Yu and Turk [123] is also based on a density field. They overcome the problem of irregular particle distribution and use an anisotropic kernel to capture the smooth shape of the free surface more accurately. Zhu and Bridson [125] achieve smooth surfaces based on a distance field over the particles. This approach is improved by Adams et al. [10] by tracking the particle to surface distance over time. Both of these methods suffer from artifacts in concavities and between particles with a certain distance. Solutions to this problem are proposed by Solenthaler et al. [102] and Onderik et al. [88]. None of these methods considers areas of contact with other, not to mention thin, deformable objects in a scene.

The interaction of cloth and fluids is not only limited to boundary handling, as most fabrics absorb fluid upon contact. Lenaerts et al. [66] apply their porous flow model to a simple particle-based cloth model. Wet cloth simulation that supports absorption and liquid transport in cloth based on a diffusion model is introduced in a previous approach [5]. A similar approach by Parker and Chaudhuri [90] additionally supports dripping and liquid diffusion in volumetric deformable objects. Based on the previous work on wet cloth [5], Um et al. [111] model changing material properties of wet thin shells, such as paper, including material weakening and tearing in this context. However, none of those methods that allow absorption consider different boundary conditions of the tangential component for dry or wet materials. This gap is filled with the presented method in Section 4.3.5. Relating to contact of wet cloth with virtual humans, Chen et al. [35] present a model with measured data for friction and wrinkling based on the imperfection sensitivity theory. Their work does not consider coupling cloth with fluid simulations.

For the simulation of fluids with the Eulerian approach, coupling to thin deformable objects is covered by Guendelman et al. [46]. Fully two-way coupling is achieved by Robinson-Mosher et al. [95] and later extended with the accurate handling of tangential velocities [94].

4.3 TWO-WAY COUPLING

To achieve a plausible motion upon contact, it is necessary that the fluid affects the cloth object and vice versa. In the presented approach, the bidirectional interaction between cloth (as a thin deformable object) and fluid is modeled with repulsions. For the interaction, the normal component and the tangential component at contact are treated separately from each other and boundary conditions are modeled for both parts. In this context, cloth is considered as impenetrable, which means that it is crucial to prevent particles from intersecting the cloth mesh. Later in this section, the ability of fluid to wet the textile is incorporated and the presented method is adapted to model appropriate boundary conditions.

4.3.1 Interaction Detection

Before handling the coupling between the particles of the fluid and the triangles of the cloth mesh, close particle-triangle pairs are identified to improve performance. To this end, a spatial hashing technique based on Teschner et al. [106] is employed. The particles are inserted in the grid and an extended bounding box of each triangle is tested for intersection with the grid cells. All particles contained in the intersected cells are considered as close. Subsequently, the proximity detection by Bridson et al. [31] is performed for the resulting particle-triangle pairs to test if a particle is located in the user-defined interaction distance. Similar to their approach, cloth is modeled with thickness τ. Following their notation, a mesh triangle represented by its corner vertices $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ and a particle's center is defined as $\mathbf{x}_p$.

The proximity test consists of two parts. First, the distance between the particles' center and the plane containing the triangle is calculated and checked against the thickness with $|\mathbf{x}_{p3} \cdot \mathbf{n}| < h$, where $\mathbf{n}$ is the normal of the triangle and $\mathbf{x}_{p3}$ the vector from the particle to $\mathbf{x}_3$. If the particle is close, the particle's position is projected onto the plane in order to evaluate if the position lies within the edges of the triangle. This is done by calculating the barycentric coordinates of the projected position on the triangle. This process results in a set of particles and triangles that are possibly in contact.

4.3.2 Normal Component of Boundary Conditions

To prevent a particle from intersecting the cloth surface and adjust the velocities, a repulsion to particle-triangle pairs is applied to keep the particle and triangle apart. To this end, a stopping impulse is applied if the particle and triangle approach each other. This is the case if the scalar product between the relative velocity $\mathbf{v}_r = \mathbf{v}_p - \mathbf{v}_t$ of a particle and a triangle with the normal $\mathbf{n}$ is negative, where $\mathbf{v}_p$ and $\mathbf{v}_t$ are the particle's and triangle's velocities. The impending intersection is stopped by applying an inelastic impulse to keep the particles and the cloth mesh separated. Thus, adding stiffness to the system at this point is avoided and the fluid can move smoothly along the cloth surface. Following the approach of Bridson et al. [31], the impulse is calculated with

$$\mathbf{J}_{c,N} = m_s \mathbf{v}_{r,N} \tag{4.1}$$

using the sum of the masses of the particle and the triangle nodes, and $\mathbf{v}_{r,N}$ is the normal component of the relative velocity. The impulse is split between particle and triangle according to their respective masses. The impulse of the triangle is distributed to the triangle nodes weighted with the barycentric coordinates to ensure proper torsional moments.

As stated in Section 4.3.1, a thin region around the cloth mesh is defined to model the thickness of the cloth. To ensure that this particle-triangle distance is kept, an elastic impulse is applied over multiple time steps once an intersection is prevented. The impulse is proportional to the the particle touches the mesh triangle. The impulse is then calculated by

$$\mathbf{J}_s = -\kappa m_s d \Delta t, \tag{4.2}$$

where κ is a material constant. The penetration depth d is given by the difference between the cloth thickness τ and the distance between particle and triangle at the current time step. The impulse is applied sequentially in a Gauss-Seidel style iteration.

Using the above mentioned repulsions, most of the possible intersections can be successfully handled in an efficient manner and the fluid moves smoothly along the cloth surface. In contrast to coupling techniques that use particle-sampled solids or introduce proxy-particles (e.g. [121]), the proposed approach provides full control over the distance between particles and the cloth mesh.

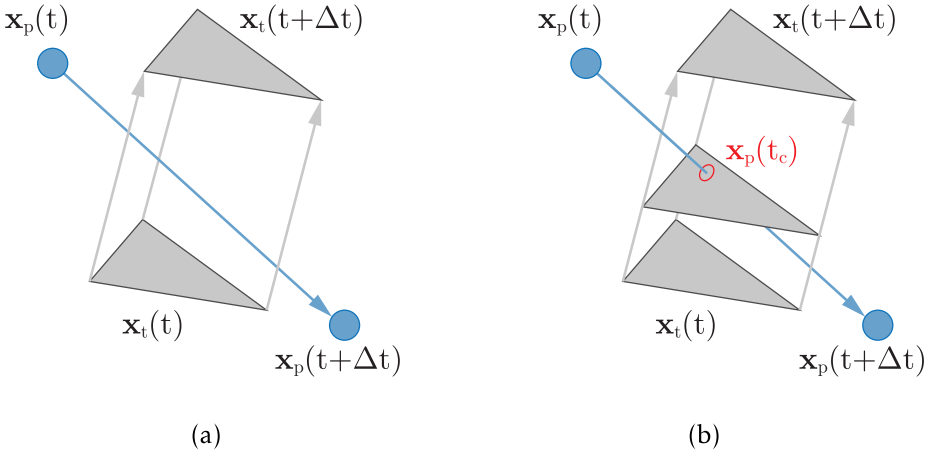

$\mathbf{x}_p(t)$
$\mathbf{x}_t(t+\Delta t)$
$\mathbf{x}_t(t)$
$\mathbf{x}_p(t+\Delta t)$
(a)
$\mathbf{x}_p(t)$
$\mathbf{x}_t(t+\Delta t)$
$\mathbf{x}_p(t_c)$
$\mathbf{x}_t(t)$
$\mathbf{x}_p(t+\Delta t)$
(b)

Figure 4.1: Intersection in between a time step: (a) A particle is intersecting a moving triangle without being detected. (b) With the continuous intersection test, the particle's position at contact $\mathbf{x}_p(t_c)$ is detected.

Although most of the intersections can be handled up to this point, it is not possible to prevent all fluid particles from penetrating the cloth mesh because intersections are not detected continuously with the approach in Section 4.3.1, but at discrete instants of time. Especially in the case of large time steps, particles can change sides of the surface unnoticed if particles or triangles cover a distance larger than τ. With the relative velocity $\mathbf{v}_r$, the maximum possible time step size to avoid intersections can be determined for a given τ. In case a larger time step is used, intersections have to be detected continuously over the trajectories.

4.3.3 Continuous Intersection Detection

The approach is extended by considering the path of the moving particle-triangle pairs in between a time step. Since in the first phase the intersection detection can fail to detect penetrations, this second phase with continuous intersection handling is employed. The goal of this step is to detect the intersection between time steps and calculate positions of the features at impact to prevent the intersection.

This step is based on the work of Provot [92]. As shown in Figure 4.1, a particle can move through a triangle of the cloth mesh in between a time step Δt, especially when using large time steps. To prevent the intersection, the time t_c at which the three nodes of the moving triangle and the moving particle are coplanar has to be determined. Similar to the approach by Provot, the unknown time t_c is calculated assuming constant velocity for both triangle and particle over the time step. As proposed by Bridson et al. [31], a rounding error tolerance in the root finding process is used and the test for collisions is performed at the end of the time step.

4.3.4 Cloth Wetting

Although the cloth object is considered as impenetrable in modeling the normal component of the two-way coupling, the most common types of fabric can be considered as a porous medium. Since the focus is on liquids, it is possible that the fluid can wet the cloth at contact. The textile can be regarded as a network of capillaries that is able to absorb fluid that will then spread inside the textile.

To this end, the wetting model presented in previous work [5] is incorporated. With this method, absorption and liquid diffusion are dependent on the specific properties of the fluid, as well as the fabric. To represent the liquid distribution in the cloth, a 2D grid is mapped on the cloth mesh where each cell can hold a certain amount of fluid. The spreading of the fluid inside the cloth is calculated using Fick's second law that is discretized on the 2D grid. In the absorption process, the change of volume due to the fluid transferred from the fluid simulation into the cells on the cloth is modeled with the simplified Lucas-Washburn equation. The corresponding amount of fluid is removed from the SPH particle and transferred to the grid cells at the contact area. Thereby, an additional boundary condition in the normal direction is modeled that allows the fluid to flow into the textile.

Spreading of liquid inside the textile is modeled as diffusion on the grid cells based on Fick's law. The cells can exchange fluid according to the diffusion equation and the process depends on the specified material properties.

Allowing for liquid transport, the behavior of moist fabric changes considerably regarding stretch and bend resistance [111], frictional contact with solids [35], and by its gained mass [5]. Beyond that, the contact characteristics between cloth and fluid with regard to the tangential component of the velocity may also differ for dry, damp, or wet cloth. To this end, the standard boundary model for the tangential component presented in the following section needs to be extended to support varying conditions of the textile.

4.3.5 Tangential Component of Boundary Conditions

With the method in Section 4.3.2, the fluid is prevented from intersecting a surface by correcting the components of the velocities in the normal direction with repulsions. However, there are additional boundary conditions regarding the tangential component of the velocity that have to be considered. As stated in fluid simulation literature, e.g. [33], letting the tangential component of the fluid's velocity untouched is only valid for an inviscid fluid. In case of a viscous fluid, the no-slip boundary condition where the fluid has zero velocity relative to the boundary is incorporated. The change of velocity can be set instantaneously.

Furthermore, tangential boundary conditions are modeled that exceed the standard slip and no-slip conditions so that boundary conditions in between are possible. Just as in the case of the normal component of the boundary conditions in Section 4.3.2, an impulse is applied on the particle and the triangle in contact. To this end, the approach of Bridson et al. [31] is adopted that was designed for modeling the friction between clothes. A friction force $\mathbf{F}_F$ acts in the opposite of the tangential component of the relative velocity. The magnitude of the friction force depends on the normal force $\mathbf{F}_N$:

$$||\mathbf{F}_F|| = \mu||\mathbf{F}_N||, \tag{4.3}$$

where μ is a unit-free control parameter. This translates to an impulse of magnitude $\mu||\mathbf{F}_N||\Delta t$. Thereby, the resulting change of the velocity is $\mu(||\mathbf{F}_N||/m)\Delta t$ that is calculated with the impulse of Section 4.3.2. The impulse for the tangential component can thus be written as

$$\mathbf{J}_{c,T} = \mu\, m_s||\Delta \mathbf{v}_{r,N}||\frac{\mathbf{v}_{r,T}}{||\mathbf{v}_{r,T}||}. \tag{4.4}$$

Considering that the change of the velocity should not exceed the tangential velocity, the change of the tangential component of the velocity can be clamped according to

$$\mathbf{v}_{r,T}^{new} = \max\left(1-\mu\frac{||\Delta \mathbf{v}_{r,N}||}{||\mathbf{v}_{r,T}^{pre}||}, 0\right)\mathbf{v}_{r,T}^{pre}, \tag{4.5}$$

where $\mathbf{v}_{r,T}^{pre}$ is the tangential component of the relative velocity at the beginning of the time step. The velocity correction is applied to both particle and triangle in the same way as the treatment for the normal component of the velocities.

As explained in the previous section, varying tangential boundary conditions dependent on the condition of the textile should also be considered. Thus, it is convenient to control the resulting tangential component of the velocity with the parameter μ similarly to the approach by Schechter and Bridson [97], to make it possible to blend between pure slip ($\mu = 0$) and the no-slip boundary condition. The parameter μ is dynamically set for a particle-triangle pair dependent on the saturation of the grid cells (see Section 4.3.4) located at the contact position on the textile with

$$\mu = 1 - \mathcal{F}\left(\frac{\text{saturation of cells}}{\text{maximum saturation of cells}}\right), \tag{4.6}$$

where $\mathcal{F}$ is a monotonically increasing function. Typically $\mathcal{F}$ is chosen as a linear function with $\mathcal{F}(1) = 1$. If the cell is fully saturated, the tangential component of the velocity is untouched and thereby, the slip boundary condition is modeled. The idea is that with a dry textile, the tangential velocity is slowed down due to the rough structure of the fabric and fluid entering the textile in the absorption process. With a fully saturated textile, a thin liquid layer on the surface forms where the fluid particles move along with low viscosity. The proposed model is inspired by similar observations in fluid mechanics [22]: the change of the tangential component of the fluid's velocity at the interface with a porous medium depends on the permeability of the medium. The method does not accurately model the physics of this process, but is used to control the tangential component of the boundary condition for animation purposes. Therefore, the function $\mathcal{F}$ can freely be changed by the user.

As an alternative approach, the impulse can also be calculated considering that the friction force that acts opposite to the tangential velocity depends on the tangential component of the relative velocity

$$\mathbf{F}_F = -\delta \mathbf{v}_{r,T}^{pre}. \tag{4.7}$$

It has to be noted that the scaling factor δ used here is not unit-free. The resulting change of velocity is then

$$\mathbf{v}_{r,T}^{new} = \mathbf{v}_{r,T}^{pre} - \delta \frac{\mathbf{v}_{r,T}^{pre}}{m} \Delta t. \tag{4.8}$$

Again, the velocity correction is clamped to avoid a negative direction of movement. Similar to Equation 4.6, a dynamically varying δ is supported.

Using the model of Equation 4.5 is recommended because the tests presented in Section 4.6 show that this model leads to better "linearity" between the control parameter μ and its effect on the animation; therefore, the mapping of the saturation of the cloth on the parameter μ (Equation 4.6) is more predictable and it is easier to control by the animator.

4.3.6 Differences to Boundary Sampling Approaches

With the approach of coupling fluids and solids using boundary particles, it is also possible to model the coupling effects and different boundary conditions by the interaction of pairwise particles. The generated boundary particles on the interacting object can be incorporated into the SPH computation of the fluid. The main challenge there is to sample the deformable mesh consistently, adjusted to the size of the fluid particles. In this chapter, a different approach is used: Coupling is achieved with the exchange of impulses between the fluid particles and the simulation mesh directly. Thus, boundary conditions can be calculated in the quality given by the simulation mesh. Coupled surfaces are represented in an arbitrary resolution, independent of the resolution of the fluid simulation. With the increasing particle count in simulation methods (e.g.

[36], [55]), decoupling the boundary representation from the fluid resolution allows a flexible scalability in particle number.

By avoiding any sampling of the cloth and the consequential need for resampling if deformations occur, temporal coherence of the treatment of the boundary is guaranteed and all cloth details, such as folds and wrinkles, are considered. In contrast, with boundary sampling, the resolution of the boundary conditions calculation is bound to the particle size of the SPH simulation. This also does not allow one to model an arbitrary thickness of the cloth that can be used to control the distance between the fluid particles and the cloth mesh.

With the proposed approach, interaction and calculation of boundary conditions are implemented as a module that couples the simulation engines in a separate process. No additional particles have to be integrated in the SPH computation loop and different simulation engines can be coupled without altering their internal systems.

4.4 SURFACE RECONSTRUCTION AT CONTACT

As mentioned in Section 4.2, there are several methods for the reconstruction of surfaces for particle-based fluids. With most of the popular MC-based surface reconstruction methods, the resulting surface mesh has a certain offset to the fluid particles. Therefore, when using small contact distances between particles and moving objects, the reconstructed surface mesh of the fluid will intersect the mesh of objects in contact. To resolve these intersections, snapping the vertices of the surface mesh to the boundary, as proposed for example by Harada et al. [48], is one possible approach. Snapping surface vertices, however, can result in (self-) intersections of the meshes. While this could work satisfactorily in case of large area boundaries, e.g. container walls, it could lead to defective meshes with complex boundaries. Another possibility is to solve these intersections manually in a postprocessing step. This however, is a time-consuming task and in some cases, it is very hard or even impossible to eliminate all the intersections.

Therefore, the surface reconstruction algorithm is modified to eliminate possible intersections. In common surface reconstruction algorithms, scalar values ϕ on the MC grid are calculated based on contributions of nearby particles. This holds for both density field approaches as well as distance field approaches (Section 4.2). When iterating over the neighboring particles, a visibility criterion is used to determine if the particle contributes to the current scalar value. A point in space is visible if the line segment that points to the reference

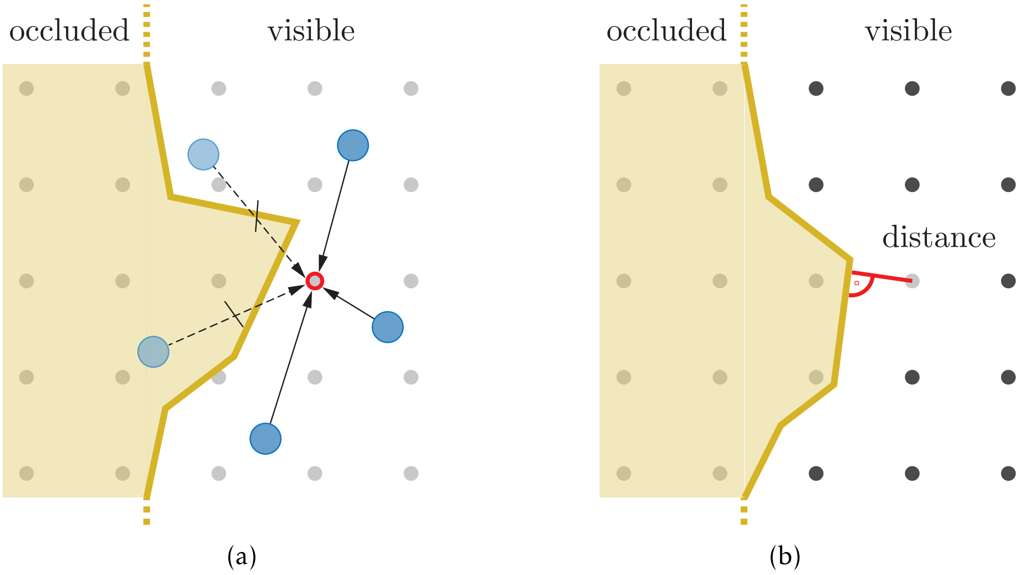


Figure 4.2: 2D view of the MC grid at a contact area: (a) For the calculation of the scalar values on grid nodes, particles that are not visible from the current grid node do not contribute. (b) Grid nodes are skipped if the distance to the closest face of the cloth mesh is smaller than a certain threshold.

point does not intersect the cloth mesh (see Figure 4.2(a)). Hence, particles that are located on the opposite side of the cloth mesh, as well as particles that are located behind folds or wrinkles, do not contribute to the scalar value computation.

At this point, it is still possible that the resulting surface mesh intersects the cloth mesh. Therefore, an additional iteration over the grid nodes is performed after the scalar value computation to identify close faces of the cloth mesh. If the distance of a triangle to a grid node is smaller than a threshold, the scalar value is reset and does not contribute to the surface generation in the MC algorithm (see Figure 4.2(b)).

With the proposed techniques, costly and error-prone repositioning of surface mesh vertices is prevented. In combination with the cloth thickness value τ as introduced in Section 4.3.2 that models the interaction distance in the cloth-fluid coupling process, it is possible to control the location of the fluid surface at contact.

Different material properties of the interaction objects also play a role for the fluid surface at the interface, especially when it comes to rendering of transparent liquids. The fact that the material appearance changes for wet objects has already been discussed in the rendering literature [57]. With the most common materials, the liquid and the solid object form a single interface at contact, which means that there is only a liquid-solid interface and no additional air layer in between. Of course, this depends on the specific material properties; for example, there are materials that have a very high liquid repellence (hydrophobic) that leads to the lotus effect at which the liquids surface is separated from the object at contact. This effect can be achieved by specifying the desired distance between the fluid particles and the surface mesh so that the render engine will produce a reflection and refraction event.

However, especially with a porous medium such as cloth, that can absorb the fluid, the fluid surface at contact is not visible. In this case, it is desirable that the fluid's surface at contact does intersect the cloth. With the proposed method it is possible to move the reconstructed fluid mesh into the cloth object that has a very small thickness without intersecting the backside, given that it has the same shape. Examples will be shown in the results section.

4.5 IMPLEMENTATION

The approach for the interaction detection and intersection handling from Section 4.3 is independent of the particular choice of the simulation engine. Any of the well-established particle-based fluid simulation systems can be coupled with state-of-the-art cloth models as long as the cloth is based on a triangle mesh. As input to the system, only the positions of the current and the previous time step of the particles and the mesh nodes are needed. This is an important property since existing simulation systems can be extended with the proposed technique without altering the systems.

The experiments are conducted with cloth simulators based on continuum mechanics (Section 2.2) of Volino et al. [115] and Etzmuß et al. [41], combined with the simple linear bending approach by Volino et al. [114] (Section 2.2.3). However, a mass-spring model for cloth is also possible. For the simulation of fluids, a standard SESPH-based method similar to the approach proposed by Müller et al [83] is employed. Since the proposed method is able to prevent leaking even at large time steps, methods such as PCISPH [101], IISPH [55], or the fluid simulation method with position-based dynamics proposed by Macklin and Müller [74] should be particularly suitable. For a detailed review of up to date SPH methods, the survey paper of Ihmsen et al. [56] is recommended.

For the surface reconstruction, the scalar value computation is based on the method presented by Zhu and Bridson [125]. Modifications by Onderik et al. [88] are added to avoid the artifacts resulting from uneven particle distributions. The method to reconstruct the fluid surface at complex boundaries, however, is not limited to a certain method to calculate the scalar values. It

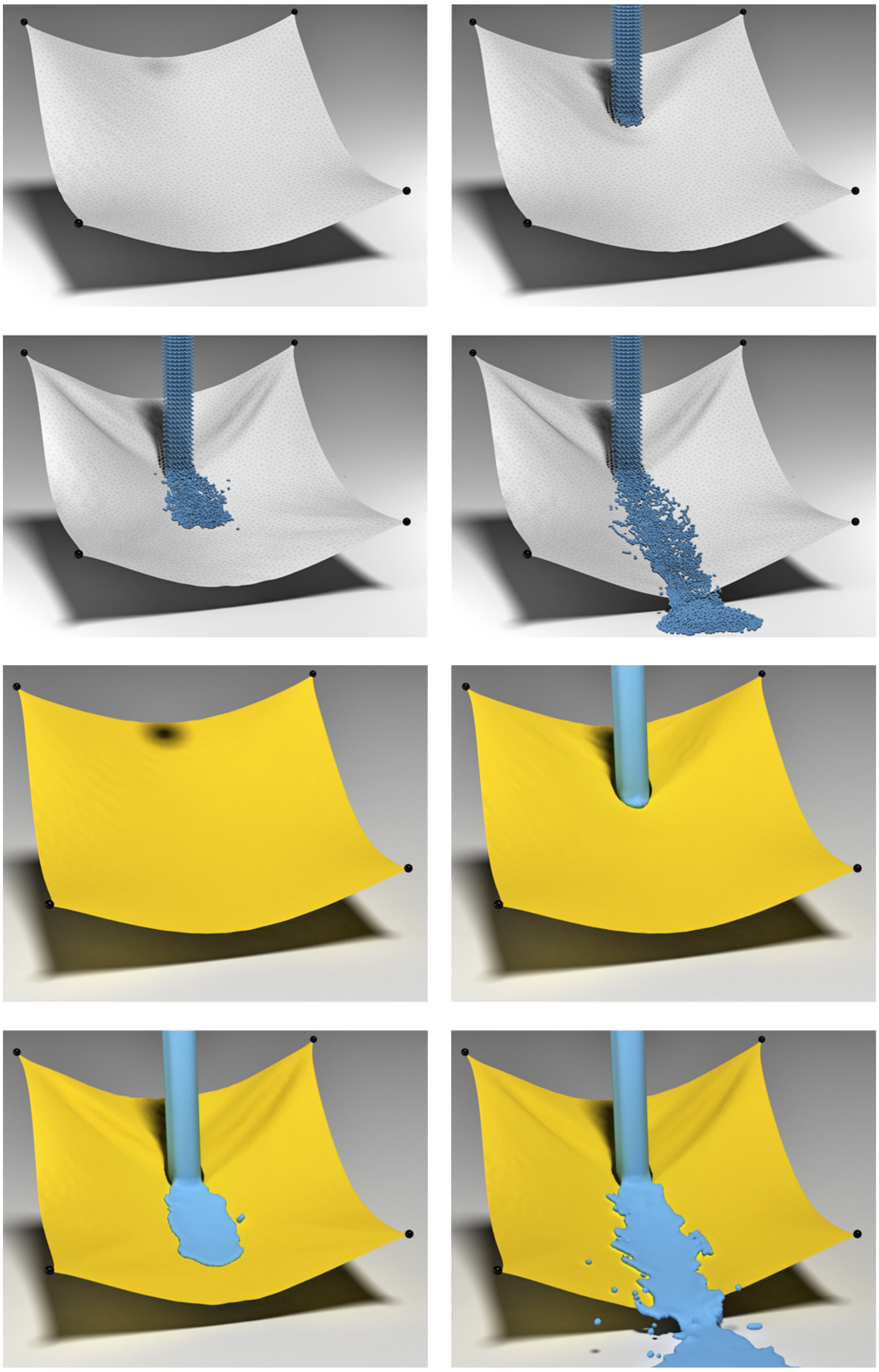
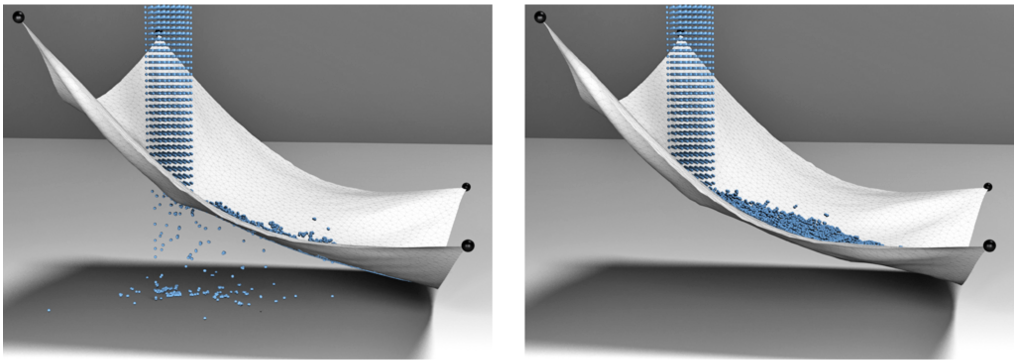


Figure 4.3: Particle view and opaque surface renderings of a jet of water hitting a piece of cloth that is pinned at its corners without leaking.



(a) (b)

Figure 4.4: Side view of the first scene. (a) With repulsions only, fluid particles can intersect the cloth surface. (b) With continuous intersection handling, leaking is prevented.

can be employed in any MC-based surface reconstruction pipeline that uses one of the surface reconstruction methods mentioned in Section 4.2.

4.6 RESULTS

The proposed methods are implemented in C++ and all the following test scenes were performed on an Intel Core i7-2600 processor at 3.4 GHz. For all scenes, the given length unit is m, the time unit is s, and the unit of mass is kg. The simulations and the intersection handling are calculated with a constant time step of $\Delta t = 0.001$ if not stated otherwise. The initial smoothing length of the SPH particles is 0.005 and the resting distance is about 0.0015 with a particle mass of approximately $3.5 \cdot 10^{-5}$ if not stated otherwise.

In the first experiment, a jet of water hits a pinned piece of fabric. As shown in Figure 4.3, the water flows off the cloth that stretches under the weight of the jet, producing wrinkles. The cloth mesh consists of about 6300 faces and there are up to $15k$ fluid particles in the scene. The two-way coupling system reliably prevents fluid particles from leaking through the cloth mesh. In Figure 4.4(a), it can be seen that penalty forces are not sufficient to prevent fluid particles from intersecting the object in contact using a small τ (0.001). With the proposed continuous intersection handling, robust two-way coupling without leaks even with large time steps (Figure 4.4(b)) is achieved. The simulation shown in Figures 4.3 and 4.4 is based on the slip boundary condition.

In the second scenario, different boundary conditions for the tangential component of the velocity at cloth-fluid contact are used, as explained in Section 4.3.5. Figure 4.5 demonstrates the effects of different values of the parameter μ in

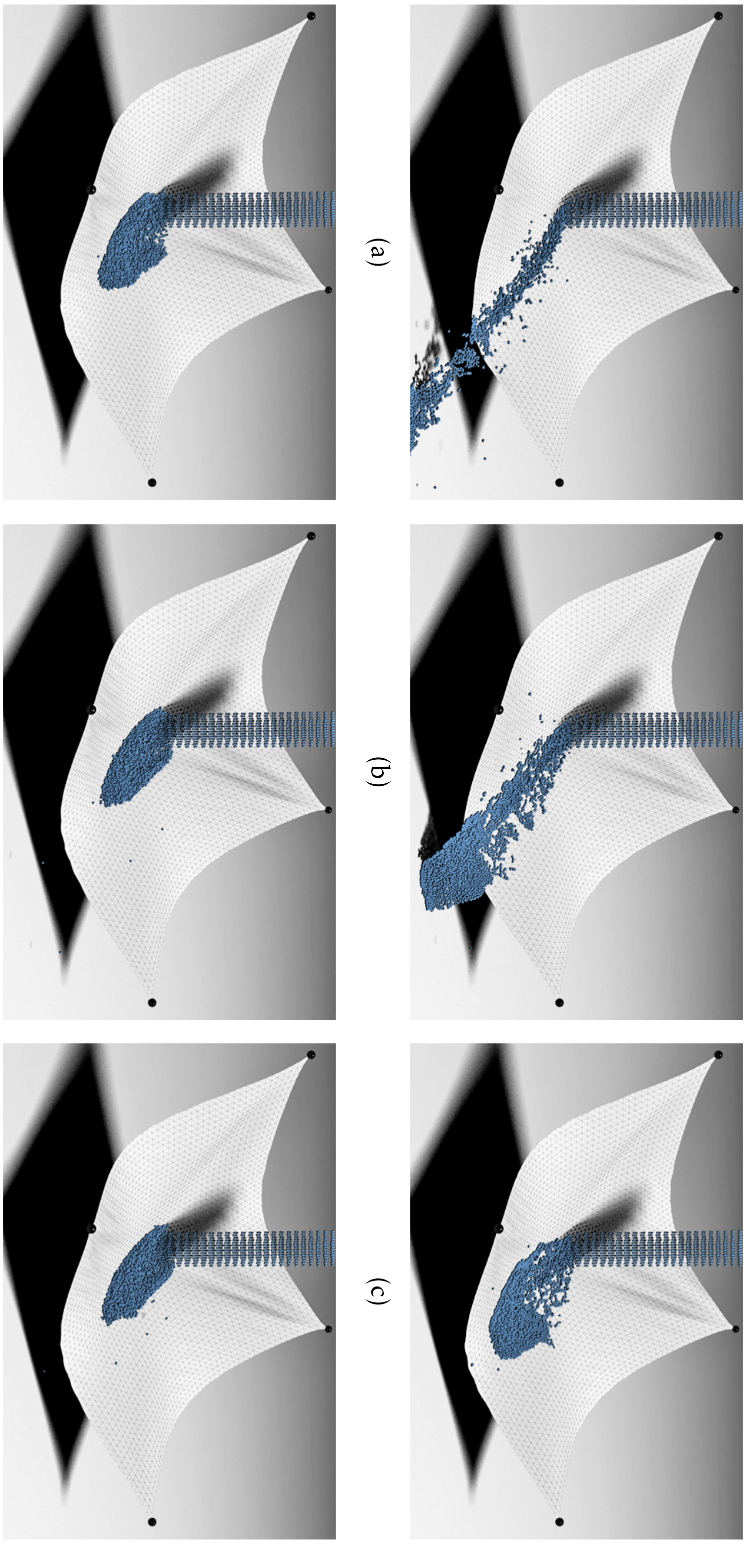
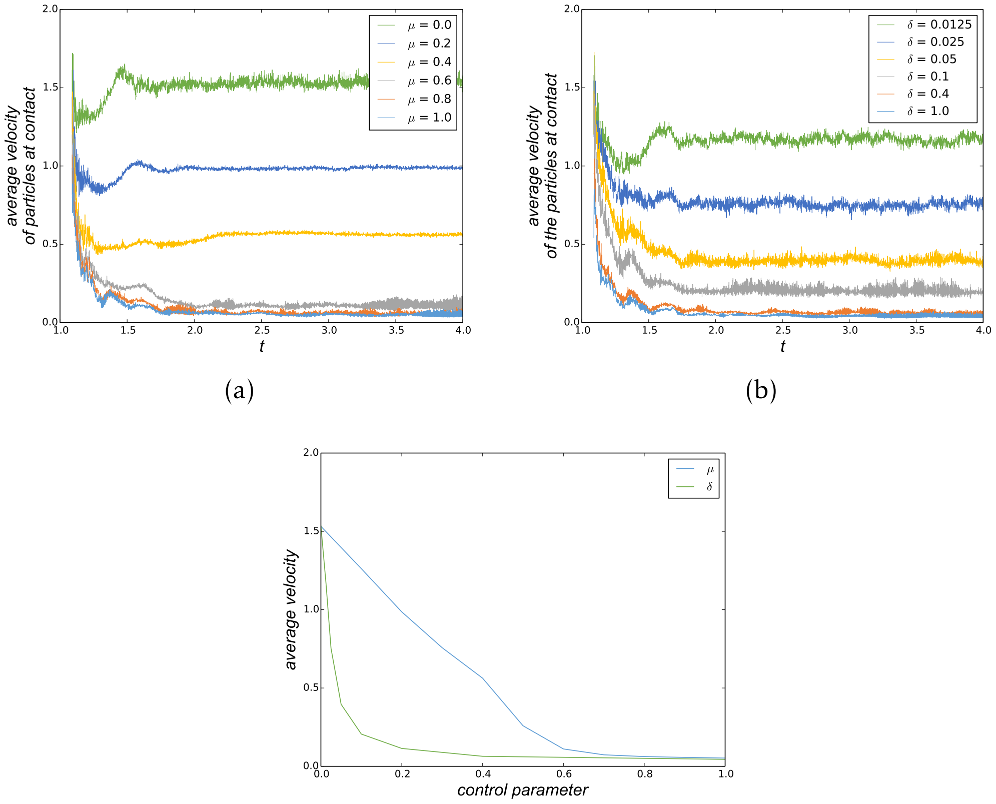


Figure 4.5: Varying parameter μ for the tangential boundary condition Equation 4.5: from the slip condition with $\mu = 0.0$ (a) to $\mu = 1.0$ (f) in steps of 0.2. The images show snapshots from the same simulation time.



(a)

(b)

(c)

Figure 4.6: Temporal evolution of the average velocities of the particles in contact with cloth as shown in Figure 4.5. (a) Velocities using the tangential boundary condition of Equation 4.5 and different values of μ. (b) Velocities using the tangential boundary condition of Equation 4.8 and different values of δ. (c) Average velocities of the particles in contact with the cloth between time 2.0 and time 4.0 for varying μ and δ.

Equation 4.5, starting with the slip boundary condition ($\mu = 0.0$) up to a value of 1.0. Since the tangential velocities of the particles in contact are slowed down differently depending on μ, the resulting animations show noticeable differences. The differences are quantified in Figure 4.6(a): the plot shows aggregated information about the temporal evolution of the velocities for different values of the parameter μ; the aggregation is performed by averaging the velocity magnitude of the particles that are in contact with the cloth. It can be observed that at the beginning of the contact, particles hit the cloth with high velocity and are slowed down considerably by the impulse acting on the tangential component of the velocities. As the particles move along the cloth, the simulation reaches an almost steady state in which the fluid moves with

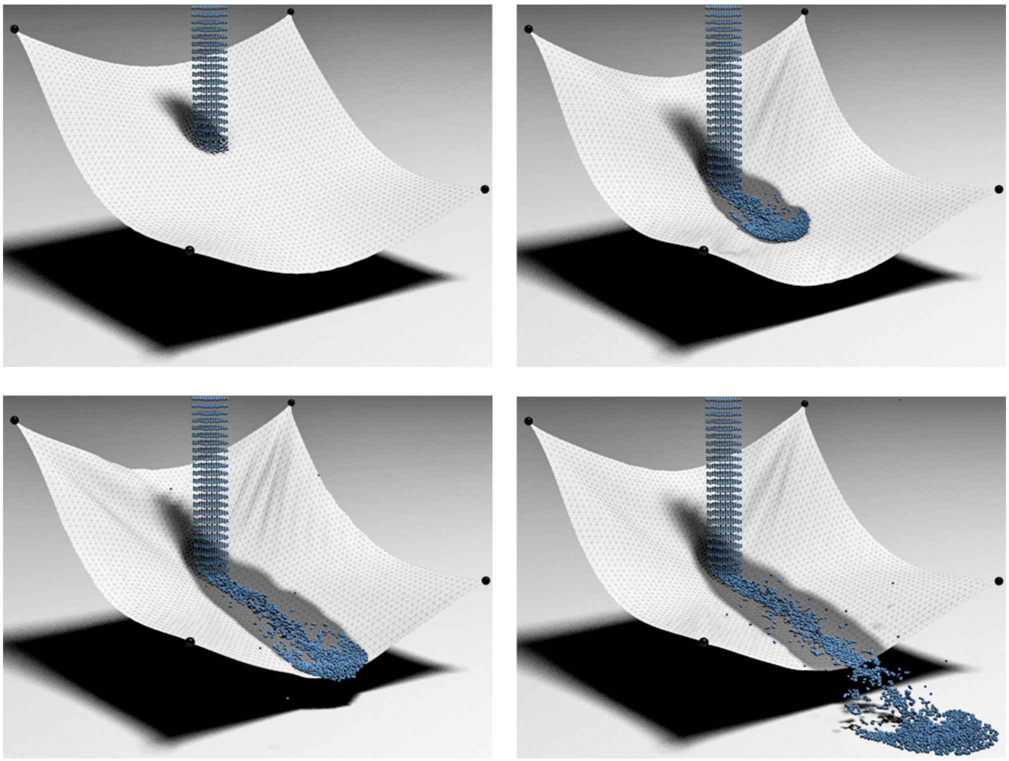
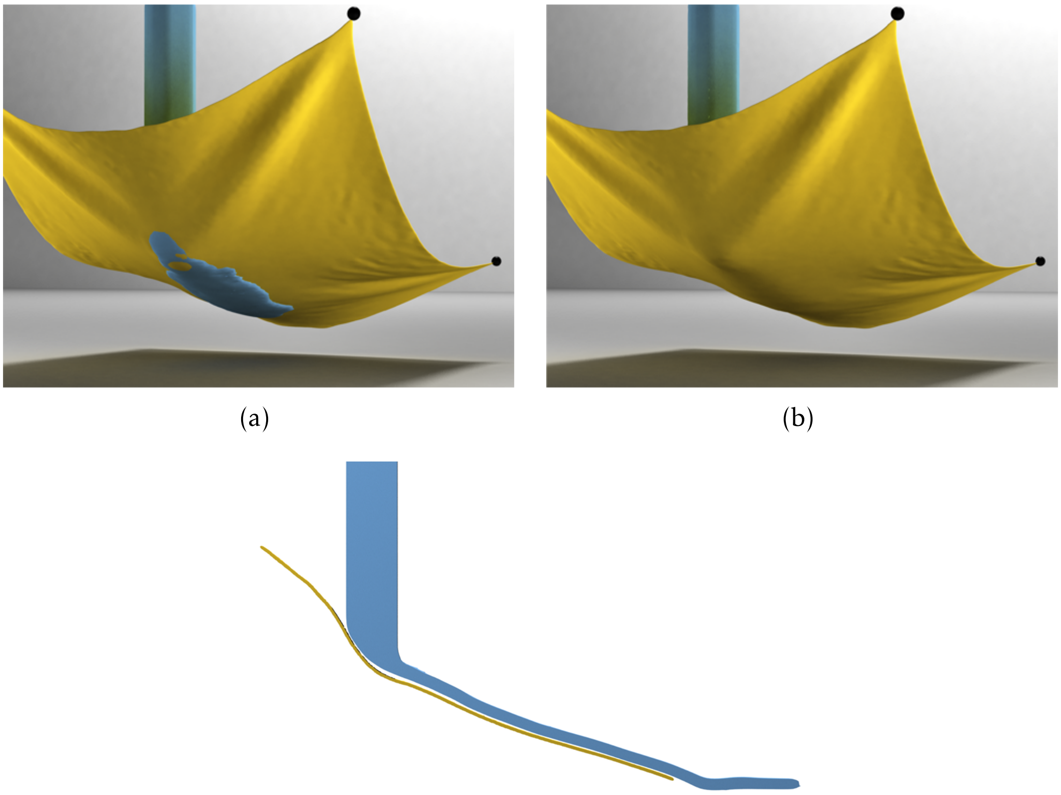


Figure 4.7: Particle representation of cloth-fluid contact considering absorption. The slip coefficient μ is set dynamically dependent on the amount of fluid in the textile, which results in a behavior notably different from static values. Saturated parts of the cloth are color-coded with a gray texture depending on the saturation in the cells.

almost constant velocity. For the alternative tangential boundary condition from Equation 4.8, the average velocities for different values for the parameter δ are plotted in Figure 4.6(b).

Figure 4.6(c) shows the effect of varying control parameters μ and δ for the respective methods for the tangential boundary conditions. The plots show the average velocities of the particles in contact, within the time span from 2.0 to 4.0, i.e., these plots provide a temporal aggregation of the plots from Figures 4.6(a) and 4.6(b). For the given example, the method depending on the normal force at contact (Equation 4.5), shows an almost linear correlation between the average velocity and the control parameter for values from 0.0 to 0.6. The constant velocity in the subsequent range from about 0.7 to 1.0 can be explained by the truncation of the max-operator in Equation 4.5, i.e., the regime of no-slip is reached. Of course, this behavior may differ for different scenarios depending on the velocity at impact, angle of the surface, and other factors. With the alternative boundary condition that depends on the relative



(a)

(b)

(c)

Figure 4.8: View on the backside of the scene in Figure 4.3: (a) Standard surface reconstruction methods can lead to visible intersections. (b) With the modified surface reconstruction algorithm, no intersections occur and there are no restrictions to the viewpoint. (c) The shape of the modified surface reconstruction matches the shape of the cloth subject. An optional repositioning within small distances is possible for rendering requirements. In rendering, the cloth mesh is extruded to account for a realistic cloth thickness.

tangential velocity (Equation 4.8), an exponential decay of the average velocity for increasing δ can be observed. Therefore, from an artistic point of view, the model of Equation 4.5 is preferable because it leads to more predictable animation behavior and is easier to be controlled.

As explained in Section 4.3.5, the slip coefficients μ and δ are allowed to be set dynamically dependent on the wetness of the cloth at contact. Snapshots of the resulting animation for varying μ are shown in Figure 4.7. In this example, the function $\mathcal{F}$ is chosen linear, starting from 0.0. At impact, the particles are slowed down strongly due to the rough structure of dry cloth and absorption. As the textile is getting wet, the impulses that affect the tangential

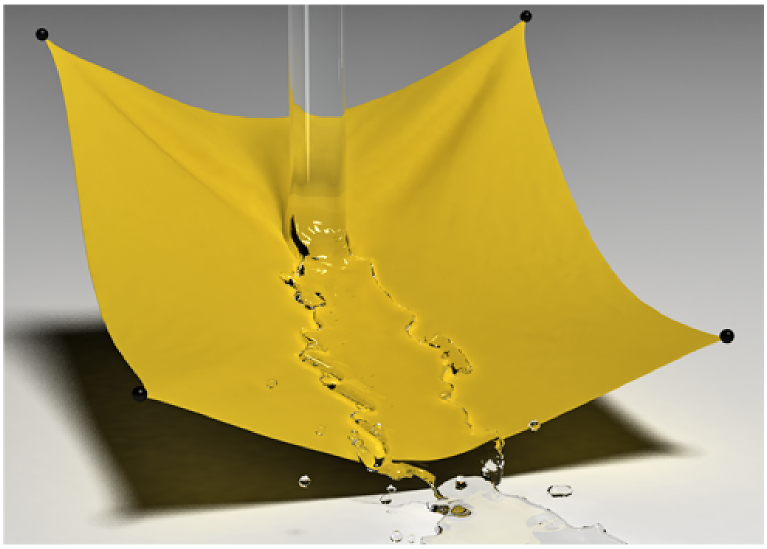


Figure 4.9: Transparent rendered surface of the scene in Figure 4.3.

component of the velocity get smaller and with a fully saturated textile, the no-slip boundary condition is reached. This effect leads to a distinct contact behavior different to the static values of μ in the previous examples.

The quality of surface reconstruction and rendering can also be examined with the first example. Figure 4.3 also shows opaque surface renderings of the first scene using a cell size of the MC grid of 0.004. The frontal view does not reveal any intersections. However, if the camera position changes as shown in Figure 4.8(a), intersections between cloth mesh and surface mesh are observed at the bottom side of the cloth with standard surface reconstruction methods, despite a certain thickness of the cloth. With the modified surface reconstruction algorithm (Figure 4.8(b)), there are no intersections visible and the camera can be moved freely in the scene. As shown in the cross section of the fourth frame of this scene (Figure 4.8(c)), the shape of the fluid's surface at contact matches the shape of the cloth with the proposed method and it is possible to position the fluid's surface on or into the object in contact to adjust for final renderings.

Figure 4.9 shows a transparent surface rendering of the same scene. The modified surface reconstruction algorithm produces visually clean results of cloth and fluid at contact, and as noted in Section 4.4, there are no rendering artifacts due to unwanted reflection or refraction events resulting from mesh intersections.

Absorption and spreading of liquid inside the textile is also shown in Figure 4.7. Due to the mass gained, the cloth behaves differently than in the first example. By moving the surface of the fluid at contact inside the cloth, there is only one solid-fluid interface and as mentioned in Section 4.4, plausible rendering with

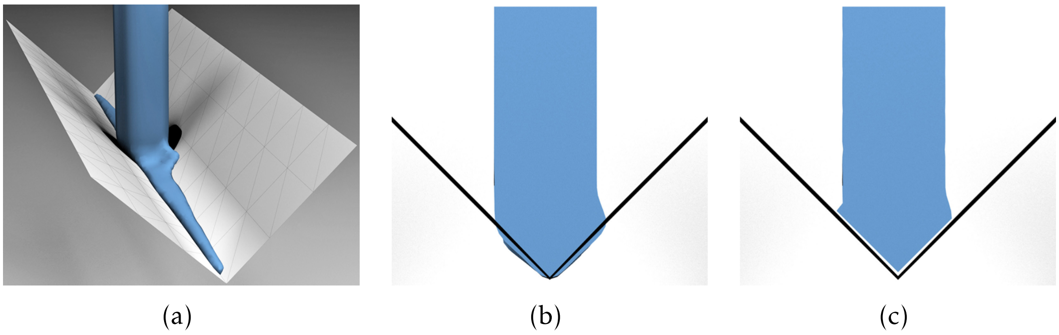


Figure 4.10: (a) A jet of water hits a V-shaped static surface. (b) With standard surface reconstruction algorithms, the surface mesh intersects the rigid plane at contact. (c) With the contact-aware reconstruction algorithm, an intersection-free state is maintained.

transparent surfaces is possible. The wet part of the cloth is visualized with a diffusion texture as presented in [5]. The diffusion cells of Section 4.3.4 are written in a 2D texture that are mapped onto the cloth. For most fabrics, it can be used to imitate the appearance of the darker color when wet.

In the next example, the surface reconstruction is demonstrated at a sharp corner. Again, a cell size of 0.004 is used for the MC grid. A jet of water is placed over a V-shaped rigid plane as shown in Figure 4.10. This synthetic example is chosen to analyze the behavior of the surface reconstruction algorithm at a sharp bend that is common in many cloth scenarios with folds and wrinkles. In Figures 4.10(b) and 4.10(b), cross sections of the respective frame of the animation are shown. If the object in contact is not considered in the surface reconstruction, the resulting mesh overlaps the interacting object. With the proposed method, an intersection-free state is maintained during the animation (Figure 4.10(c)) and the scene can be rendered artifact-free from any perspective.

Table 4.1: Computation times (in seconds) of the simulations and two-way coupling for a time span of 0.5 for the scene in Figure 4.3 with different cloth thickness values τ. With a large value of τ, the continuous intersection test is not needed to prevent leaking.

τ	Cloth	Fluid	Coupling	Repulsions	Intersections
0.0005	26.83	2.97	29.32	2.85	19.53
0.001	26.34	2.95	25.30	2.70	16.31
0.005	27.73	3.66	7.21	1.83	–

Table 4.1 provides computation times of the simulations and the two-way coupling for the first scene as shown in Figure 4.3. A small cloth thickness τ results in higher costs for the continuous intersection test. Using a broad region around the triangles of the cloth mesh, the continuous intersection handling is not needed to prevent leaking in this scene, which explains the substantial drop in computation time for coupling in the last row. Table 4.1 also shows the fractions of the computation times of the coupling for the same scene, broken down into the repulsion step and the continuous intersection handling. As expected, the continuous intersection handling accounts for the major part of the coupling.

4.7 SUMMARY

This chapter presented an approach to model fluids and cloth at contact. With the proposed method for the two-way coupling, robust and efficient bidirectional interaction between particle-based fluids and cloth is achieved. The system combines repulsion with a continuous intersection test to ensure that particles do not leak through the cloth surface. In experiments, it is demonstrated that the system can handle cloth-fluid interaction even at large time steps without using substeps. Additionally, different boundary conditions for the tangential component of the velocities can be modeled that allow one to create animations for a wide range of interacting materials. As an extension, wetting of the textile is considered and the method is adapted to account for the changing interaction behavior. The approach for the two-way coupling is designed to be independent from the underlying simulation systems and can easily be employed in existing simulation pipelines.

Furthermore, a novel method for the surface reconstruction of particle-based fluids at complex shapes is presented. This method can be used for common MC-based surface reconstruction algorithms to generate intersection-free surface meshes at contact. Visible artifacts are eliminated and the flexibility of rendering transparent surfaces in contact with thin deformable objects is considerably increased. The shape of the resulting surface mesh at contact with complex shaped cloth meshes depends on the resolutions of the cloth mesh and the MC grid. The common MC grid resolutions that produce a smooth surface mesh are usually small enough to adjust to a buckled mesh. If, however, the surface mesh should align with a very high resolution cloth mesh with fine wrinkles, a very small cell size of the MC grid is needed. In this case, an adaptive MC grid could be used to avoid long computation times.

CHAPTER 5

Evaluation of SPH Interfaces

5.1 Introduction

Animated surfaces also play an important role in the field of visualization of SPH-based simulation of physical or chemical processes. This chapter presents an evaluation of surface representations for SPH-based simulation of phase inversion in fluid mixtures. The goal of the visualization of the surface is to offer the domain experts who provide the simulation data the possibility to visually analyze the underlying process. For a qualitative and quantitative analysis, the total area of the extracted surfaces is calculated and its development over time is plotted in a line chart. This line chart can be used to verify the simulation method by comparing the temporal development of the area to an analytic power law that describes the decrease of the surface area for the specific phase separation. In the evaluation, two surface extraction methods for SPH simulations are compared: A density field-based approach that uses the same kernel function as the underlying simulation and a distance field approach often used in computer graphics applications. The comparison of the respective surface areas allows to assess the applicability of the two surface extraction methods for the given simulation. The evaluation and comparison has been conducted together with collaborators working in the field of material science considering their requirements.

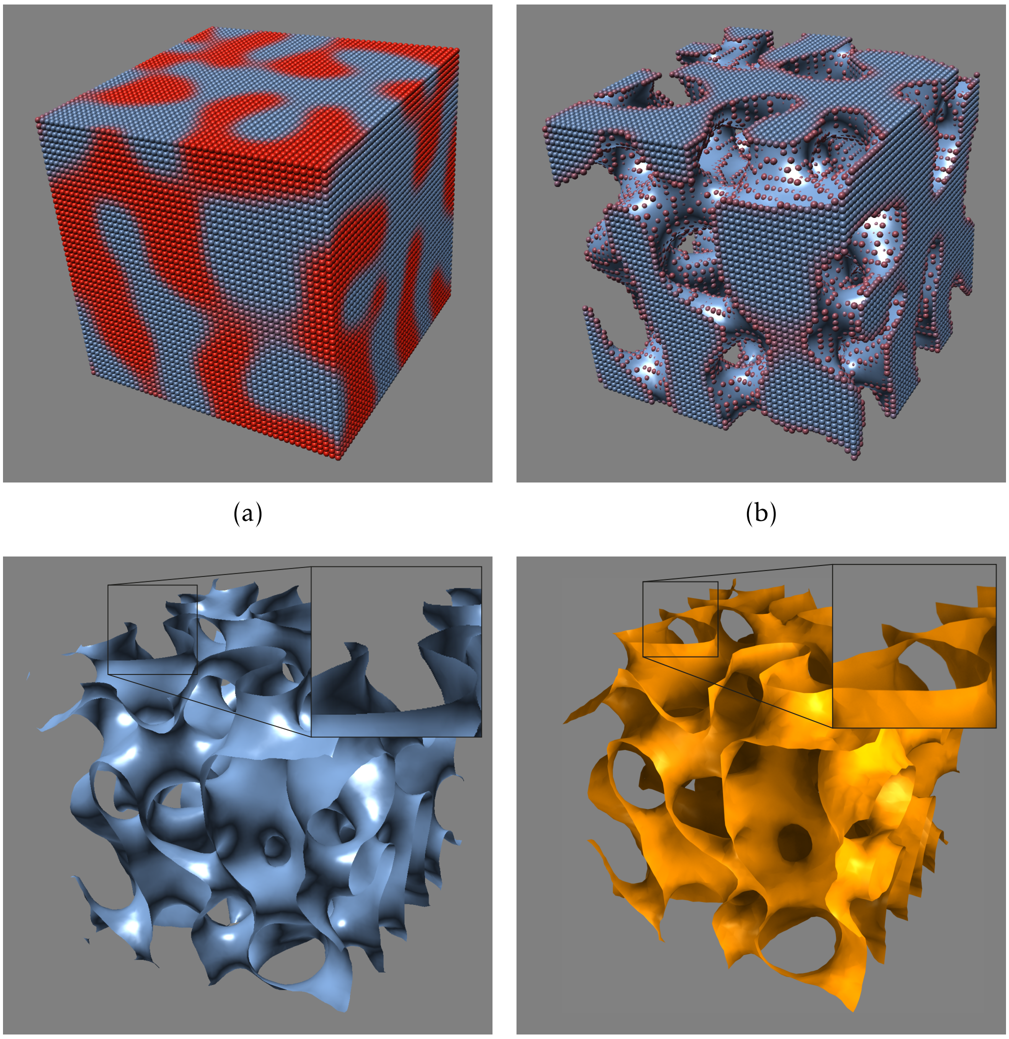
5.2 RELATED WORK

Surface reconstruction techniques for particle-based (SPH) simulations in computer graphics have already been discussed in Section 4.2. These methods focus on the generation of smooth, visually pleasing surfaces particularly for the animation of liquids. As a representative, a distance field approach [88], [125] is considered in this evaluation.

In the context of simulation visualization for analysis, Schindler et al. [98] present marching correctors, a variant of the MC algorithm. Kolb and Cuntz [62] generate a uniform density volume of the particles on the GPU and used point sprites for rendering. Goswami et al. [44] present a CUDA-based SPH simulation and visualize the simulation results by creating a distance field from the particles and render it using GPU-based raycasting. Fraedrich et al. [42] present a method to visualize very large SPH simulations using an adaptive view-dependent discretization of the simulation domain to sample the particle densities. In the work of Molchanov et al. [78], a feature-rich interactive framework for the analysis of SPH simulations in the application area of astrophysics is described. Besides rendering point-clouds of the simulated data, they use a splatting technique to represent isosurfaces. In addition to providing a visualization of the SPH simulation only, this chapter also addresses the verification of a simulation based on the surface area.

5.3 BACKGROUND

Polymer membranes are widely used in chemical engineering applications such as in battery systems. They are semi-permeable, thus allowing only one specific substance to pass through. To develop and improve membranes, it is recommended to predict the morphology in dependence of the manufacturing conditions. In the preparation process, the so-called phase inversion plays an important role. If a homogeneous fluid mixture separates into two stable phases, it is called phase inversion. One way to describe phase inversion is to use the Cahn-Hilliard equation [34], which is a fourth-order PDE for diffusive mass transport. For discretization, the collaboration partners provide an SPH-based simulation using the method of their previous work [52]. For simplicity, a binary, isothermal, incompressible, and equimolar fluid mixture is considered to validate the dynamics of 3D simulations. Usually, SPH particles move on a trajectory but in this special case, the momentum per particle is constant and the particles are fixed at their initial positions. Therefore, only the transport equation for the concentration of one component is solved. Nevertheless, the position of the interface between two phases changes in time.



(a)

(b)

(c)

(d)

Figure 5.1: Particle positions and surface representations of the given simulation: (a) Particle positions of the SPH simulation. Two phases are discernible (red and blue). (b) and (c) The isosurface that separates the two phases. The surface extraction is based on the kernel function that was used in the underlying simulation. (d) The surface that was extracted using the method of Onderik et al. [88]. The magnifications in the two images (c) and (d) show a case where the two methods create different surfaces.

5.4 ALGORITHMS AND IMPLEMENTATION

For the SPH simulation, a Wendland kernel function [118] is used. For visualization, particles are weighted with this kernel function and a density field that represents the simulation data is calculated. The resulting density ρ at position $\mathbf{x}$ is

$$\rho(\mathbf{x}) = \sum_{j=1}^{N} m_j \frac{21}{2\pi h^3} \left(1 - \frac{r}{h}\right)^4 \left(4\frac{r}{h} + 1\right), \tag{5.1}$$

where $r = \|\mathbf{x} - \mathbf{x}_j\|$, $\mathbf{x}_j$ is the position of particle j, and the smoothing length $h = 1.55L_0$ with the particle distance L_0. If $r \geq h$, particle j does not contribute to the density ρ at position $\mathbf{x}$. That is, only neighboring particles within the cutoff radius of $2h = 3.1L_0$ contribute to $\rho(\mathbf{x})$, as in the simulation. The computation of the density volume is implemented using CUDA. Each position $\mathbf{x}$ in the density field (that is, each voxel) has a limited neighborhood that is defined by h. For m voxels, the density ρ can be computed in $\mathcal{O}(m)$. Since ρ of each voxel can be computed independently of all other voxels, the computation can be executed in parallel. All particles are sorted into a uniform acceleration grid with a grid spacing equal to h. For each voxel, only $3 \times 3 \times 3$ grid cells have to be evaluated to find all neighboring particles. The implementation is based on the work of Krone et al. [64], presenting an optimized CUDA implementation that uses a Gaussian density kernel.

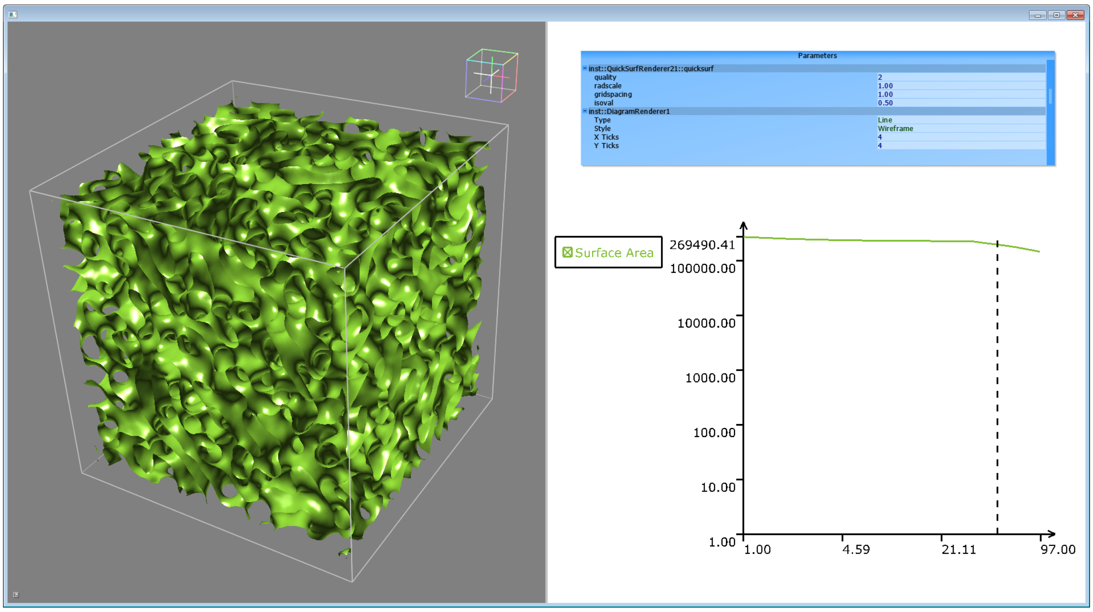
As an alternative definition of the SPH surface, a method based on the approach by Zhu and Bridson [125] is used. The idea is to calculate a scalar value at a given position $\mathbf{x}$ by measuring the distance to a weighted sum of the neighboring particle centers. As discussed by Solenthaler et al. [102] and Onderik et al. [88], it is possible that the weighted centers are located outside of the desired surface, which can lead to extensive visual artifacts. Therefore, the modified implicit surface definition of Onderik et al. [88]

$$\phi(\mathbf{x}) = \|\mathbf{x} - C(\mathbf{x})\| - Rf(\mathbf{x}) \tag{5.2}$$

is used, where R controls the distance of the surface to the boundary particles. R is multiplied with a decay function $f(\mathbf{P})$ as defined in [88] to eliminate artifacts. The weighted sum of the neighboring particle centers is calculated using normalized particle averages with

$$C(\mathbf{x}) = \frac{\sum_{j=1}^{N} \frac{1}{w_j} \mathbf{x}_j W\left(\|\mathbf{x} - \mathbf{x}_j\|, h\right)}{\sum_{j=1}^{N} \frac{1}{w_j} W\left(\|\mathbf{x} - \mathbf{x}_j\|, h\right)}, \tag{5.3}$$

where w_j is determined using the SPH interpolation of the positions of the particles' neighbors and the polynomial smoothing kernel $W(r,h) = (1 - (r^2/h^2))^3$ is used. The isosurface is also extracted using MC.



Parameters

− inst::QuickSurfRenderer21::quicksurf	
quality	2
radscale	1.00
gridspacing	1.00
isoval	0.50
− inst::DiagramRenderer1	
Type	Line
Style	Wireframe
X Ticks	4
Y Ticks	4

Figure 5.2: Screenshot showing the extracted surface using the density field calculated by Equation 5.1 of an SPH-based simulation of 10^6 particles. The line chart to the right shows the area of the extracted surface over time.

The total surface area of the extracted isosurfaces is computed by summing up all individual triangle areas. For each simulation frame, the surface area is plotted over time in a 2D line chart (see Figure 5.2).

5.5 RESULTS AND DISCUSSION

Particle and surface visualizations of the two methods given in Section 5.4 of a multiphase simulation are shown in Figure 5.1. On a test system with an Intel Core i7 (3.6 GHz), 32 GB RAM, and an Nvidia GTX Titan (6 GB VRAM), the surface visualization could be computed at interactive rates for most data sets. With the largest data set of 10^6 particles (Figure 5.2) and a 199^3 grid, performance dropped to 5 fps with the density field using the Wendland kernel (Equation 5.1). The distance field approach (Equation 5.2) has a lower performance because of the additional iteration over all particles.

The surface area of both of the surface generation methods is compared to the analytical function that describes the power law. The growth rate of the area corresponds satisfactorily to the power law. The relative discretization error stemming from the lower grid resolution is negligible ($\sim 10^{-3}$). Figure 5.3 shows a line chart with the surface areas of the four simulations plotted over time and the power law for reference. As observable in Figure 5.3, the distance-based method (dashed red curve) closely matches the corresponding area of

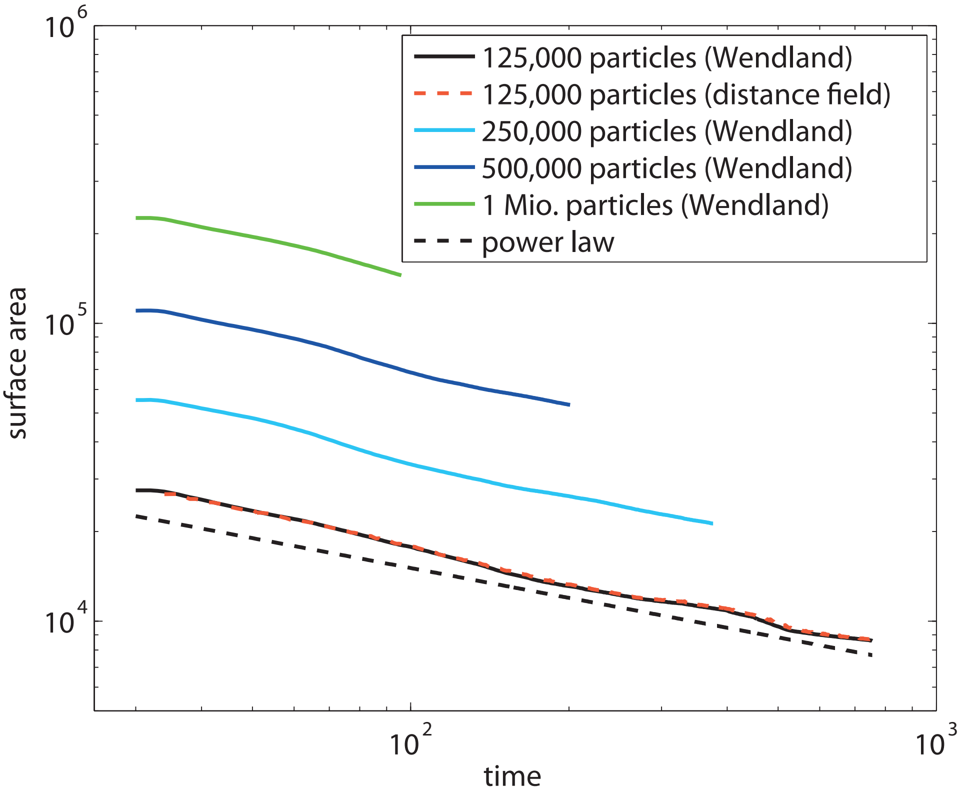


Figure 5.3: Surface area over time for different particle numbers using the volume computation based on the Wendland kernel (both axes have logarithmic scale). The decrease of the area adheres to the power law for all simulations (dashed black line). Note that the larger data sets are simulated over shorter periods. For 125k particles, the chart also shows the curve for the alternative surface based on the distance field (dashed red curve).

the Wendland kernel (black curve). Consequently, it also satisfies the power law. That is, both methods can be used for a qualitative analysis of the SPH simulation data of the collaborators. However, the results for the two surfaces differ by about 1% of the total surface area in some cases. Thus, the deviation of the distance-based method could be too high for a quantitative analysis in some cases. Further, the distance-based method constructs surfaces that differ from the density-based method, and therefore, do not exactly represent the simulation in some cases. Figure 5.1 shows an example where this is the case, even though the overall surface area differs by less than 1%.

5.6 SUMMARY

This chapter presented the evaluation of visualizations for surfaces of SPH-based simulations of phase separation in fluid mixtures. An important aspect of the visualization is the faithful representation of the underlying simulation. Based on the surface area, the correctness of the phase separation simulation is verified since the area has to adhere to a given power law.

For visualization, a density-based surface extraction method is compared to a method that was designed to extract a smooth, visually pleasing surface of a SPH-based fluid simulations [88]. With both methods, the surface area has a low error rate and adheres to the aforementioned power law, and therefore, both are suitable for analysis. For quantitative analysis, though, the density-based surface is more accurate.

CHAPTER 6

Evaluation of Surface Tension Models for SPH-Based Fluid Animations

6.1 INTRODUCTION

Surface tension is a distinctive characteristic of liquids and its effects have been identified as one of the key components to recreate physically plausible and visually appealing fluid phenomena. For SPH-based fluid simulation systems, there exist several models to incorporate surface tension. They address the representation of the diverse effects of surface tension and the specific challenges that occur at free surfaces of particle-based fluids. The underlying approaches of these methods to model surface tension vary and, therefore, show different properties in their behavior and usage. Although some comparisons with other models are conducted in the original papers of the surface tension models, there are no explicit common standards for comparison that allow thorough comparative analysis. This chapter presents a systematic evaluation of surface tension models in combination with different SPH methods using a benchmark test that is intended as a tool for the comparison of such models.

In case of liquids, the shape of the fluid, especially at the interface of the liquid and the gas phase (e.g. water and air), is highly influenced by surface tension. Depending on the magnitude of the surface tension, various effects regarding the appearance and behavior of a liquid can be observed. In typical computer graphics applications, fluids are usually modeled as single-phase free surface flow and there exist numerous challenges that arise with the modeling of surface tension with SPH, such as the underestimation of density at interfaces, large surface tension coefficients, and the handling of thin features. Existing techniques present a number of different approaches to overcome these difficulties: For instance, surface tension can be modeled through forces acting on the surface particles in order to minimize the curvature and hence, its energy [51], [83]. A second approach is to use pairwise forces between neighboring particles [23], [122] or a combination of both techniques [12]. In this chapter, surface tension models are evaluated with particular attention to the combination with different SPH solvers.

The variety of surface tension models in combination with the different SPH approaches makes it difficult to compare the models in detail and to determine a model's suitability for a certain task. Therefore, a benchmark test for the evaluation and comparability of surface tension models is established. Three SPH methods and three surface tension models are chosen as representatives for a class of techniques, respectively. For the evaluation, the benchmark tests are applied to existing models for the evaluation of such models. The goal of this chapter is to identify strengths and weaknesses of these models and understand their suitability for possible applications. Also, key properties of surface tension models should be highlighted to create desired animations and to facilitate the development of novel methods.

The overarching contribution of this chapter is the systematic evaluation of surface tension models using a benchmark test in order to determine the properties of a model not only visually, but also in a quantitative manner. The evaluation process is applied to three existing surface tension models in combination with three up to date simulation systems. For comparability and reproducibility, uniform settings are used for each of the scenarios and complete information, such as kernels and parameters, is provided. Results from the application of the benchmark to the surface tension models are presented and the observed properties of the simulation models are discussed.

The source code of the implementation used for the evaluations is made available to the public along with example initialization files for the benchmark scenes at http://go.visus.uni-stuttgart.de/sphevaluation.

6.2 RELATED WORK

When new techniques are presented, it is common practice to apply them to a certain set of scenarios, e.g., a breaking dam or a fluid pillar for general fluid simulation techniques, different fixed or moving obstacles for boundary handling methods, and interaction with other dynamic objects for two-way coupling models. In case of surface tension, the formation of a drop in absence of gravity and the dynamics of a liquid crown have been commonly used. Although these tests can be considered as standard procedures, they are rarely performed with uniform setups, e.g., regarding simulation methods, particle counts, and parameter values. Different configurations and sometimes missing specifications make it difficult to do a comparative evaluation and regard these scenarios universal benchmark tests.

A typical benchmark in the field of level-set methods is the rotating Zalesak disk or sphere, used to determine the quality of methods for animated surfaces (e.g. [27], [40]). Using this benchmark, evaluation and especially quantification is possible, e.g. by measuring volume loss. However, there are no common benchmarks in fluid animation for computer graphics.

Surface reconstruction techniques for particle-based fluid simulation are listed in Section 4.2 and are usually independent from surface tension. However, it is also possible to calculate surface tension based on the curvature of the surface mesh as shown by Yu et al. [124]. In this chapter, however, only surface tension approaches based on the simulation particles are considered.

6.3 SIMULATION MODELS

The evaluation is performed on combinations of different classes of SPH solvers with surface tension models. In the following, a brief overview of the used models is given.

6.3.1 SPH-based Fluid Simulation

The basic concept of SPH-based fluid simulation is derived in Section 2.1 and a standard fluid simulator using SESPH is described in Section 2.1.3. As discussed in the report by Ihmsen et al. [56], a variety of SPH-based fluid methods exist and the common models in computer graphics mainly differ in the calculation of pressure forces. The methods can be categorized by the pressure model, if it is based on an EOS or based on a pressure Poisson equation (PPE). This chapter incorporates three models, each representing one class: For a non-iterative EOS solver, weakly compressible SPH using a state equation (SESPH) [23] is implemented, for iterative EOS solvers with splitting, PCISPH [101] is used, and IISPH [55] is implemented for the pressure computation based on a PPE.

In addition to incompressibility, Bender and Koschier [25] present a method for iterative solvers that also enforces a divergence-free velocity field. Their divergence-free SPH (DFSPH) further increases stability and performance of incompressible fluid simulations, however, is not considered in this study.

The basic principles of SESPH are shown in Section 2.1.3 and an illustration of its use in an SPH-based fluid simulation loop is given in Section 2.1.5. SESPH is considered here due to its straightforward implementation, its reasonable results, and because it is used in many existing simulation systems.

PCISPH is based on the concept of splitting of force calculations in the particle loops. With splitting, particle velocities and positions are first evaluated without pressure forces, using an explicit Euler step based on the sum of viscosity forces $\mathbf{F}_i^v$ and external forces $\mathbf{F}_i^b$. Then, pressure forces are calculated iteratively. First, density is calculated for each particle based on these intermediate velocities $\mathbf{v}^*$ using

$$\rho_i^* = \sum_{j=1}^{N} m_j W(||\mathbf{x}_i - \mathbf{x}_j||, h) + \Delta t \sum_{j=1}^{N} m_j (\mathbf{v}_i^* - \mathbf{v}_j^*) \nabla W(||\mathbf{x}_i - \mathbf{x}_j||, h). \tag{6.1}$$

Pressure is then evaluated using ρ_i^* in the EOS (Equation 2.35) with a relatively small stiffness constant and pressure forces can be computed (Equation 2.33). With the pressure forces, new intermediate velocities and positions of the particles are calculated. These steps are repeated iteratively until the density error ρ^{err}, defined as the maximum deviation of the calculated density to the rest density, is below a global threshold. Therefore, incompressibility is enforced using multiple iterations and pressure forces are, compared to SESPH with a large pressure stiffness constant, small. The additional computational cost of the iterations is compensated by the possibility of using larger time steps, and increased computational efficiency compared to SESPH. Details of the algorithm can be found in the work by Solenthaler and Pajarola [101]. With PCISPH, an easy-to-implement incompressible fluid simulation is available that has been widely used in SPH-related literature.

In contrast to evaluating pressure using an EOS, IISPH uses the solution of a PPE. Similar to PCISPH, it is also based on the splitting concept. The source term for a discretized PPE is expressed by the compression $\rho_0 - \rho^*$ as

$$\nabla^2 p_i = \frac{\rho_0 - \rho_i^*}{\Delta t^2}. \tag{6.2}$$

After having calculated the intermediate velocities $\mathbf{v}_i^*$ and densities ρ_i^* , Equation 6.2 can be written in terms of SPH and unknown pressure forces $\mathbf{F}_i^p(t)$:

$$\Delta t^2 \sum_{j=1}^{N} m_j \left(\frac{\mathbf{F}_i^p(t)}{m_i} - \frac{\mathbf{F}_j^p(t)}{m_j} \right) \nabla W(||\mathbf{x}_i(t) - \mathbf{x}_j(t)||, h) = \rho_0 - \rho_i^*, \tag{6.3}$$

which leads to a linear system of equations. A detailed derivation can be found in the original work by Ihmsen et al. [55]. In their work, the system is solved using a relaxed Jacobi scheme. With the calculated pressure forces, the particle velocities and positions can be updated using a semi-implicit Euler step (Equations 2.46 and 2.47). Ihmsen et al. [55] show that this incompressible SPH approach is especially suitable and efficient with large time steps and, therefore, useful for high particle counts.

6.3.2 Including Surface Tension

Surface tension in single-phase particle-based fluids can be seen from different viewpoints, either as (molecular) interaction between particles or in terms of energy that causes particles located at the interface to a (virtual) second phase to form a curvature-minimizing surface. Current surface tension approaches differ by their viewpoint and are modeled according to one viewpoint, or as a combination of these. However, they have in common that they result in forces that are integrated in Equation 2.31 either as additional inter-particle forces or body forces.

In the following, four surface tension models that are considered in this study are briefly summarized.

Inter-Particle Interaction Forces

Becker and Teschner [23] propose a microscopic model for surface tension based on the work of Tartakovsky and Meakin [105]. In this model, the fluid particles act as actual particles with the surface tension being modeled as (molecular) forces between neighboring particles. These types of models are often referred to as inter-particle interaction forces (IIFs). According to Becker and Teschner [23], surface tension emerges from cohesion forces between particles and results in velocities

$$\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = -\frac{\varphi}{m_i}\sum_{j=1}^{N} m_j(\mathbf{x}_i - \mathbf{x}_j)W^{st}(||\mathbf{x}_i - \mathbf{x}_j||, h), \tag{6.4}$$

that are added to the present velocities of the particles, where φ controls the magnitude of the surface tension force and W^{st} is the kernel used for the evaluation of surface tension. For a consistent formulation in this section and improved comparability, the surface tension model in Equation 6.4 can be rewritten in terms of forces as

$$\mathbf{F}_i^{st} = -\varphi\sum_{j=1}^{N} m_j(\mathbf{x}_i - \mathbf{x}_j)W^{st}(||\mathbf{x}_i - \mathbf{x}_j||, h). \tag{6.5}$$

Combined Inter-Particle and Surface Forces

Since IIFs can only reproduce a portion of surface tension effects, Akinci et al. [12] use a combination of inter-particle forces and forces based on surface curvature. First, the inter-particle interaction forces are modeled with a cohesion force

$$\mathbf{F}_i^{coh} = -\gamma m_i \sum_{j=1}^{N} m_j C(||\mathbf{x}_i - \mathbf{x}_j||, h) \frac{\mathbf{x}_i - \mathbf{x}_j}{||\mathbf{x}_i - \mathbf{x}_j||}. \tag{6.6}$$

In this force calculation, the usually used SPH kernel is replaced with a function C that includes a repulsion term for close particles similar to Tartakovski and Meakin [105] to avoid particle clustering. The function C is given by

$$C(r,h) = \frac{32}{\pi h^9} \begin{cases} (h-r)^3 r^3 & \text{if } 2r > h \wedge r \leq h, \\ 2(h-r)^3 r^3 - \frac{h^6}{64} & \text{if } r > 0 \wedge 2r \leq h, \\ 0 & \text{otherwise.} \end{cases} \tag{6.7}$$

In addition to the inter-particle forces, a continuum surface force (CSF) is also employed in this model [30], [83]. With the CSF approach, surface tension is modeled as a pressure on the interface between the liquid and the gas phase resulting in a normal force. In contrast to Müller et al. [83], the surface curvature is not calculated explicitly and the normal approximation is evaluated based on the gradient of the density field

$$\mathbf{n}_i = h \sum_{j=1}^{N} \frac{m_j}{\rho_j} \nabla W(||\mathbf{x}_i - \mathbf{x}_j||, h), \tag{6.8}$$

scaling with h. Using h, normals can be calculated independent from the simulation scale. The curvature of the surface is given implicitly by the magnitude of $\mathbf{n}$ and the curvature-based force is given by

$$\mathbf{F}_i^{curv} = -\gamma m_i \sum_{j=1}^{N} (\mathbf{n}_i - \mathbf{n}_j). \tag{6.9}$$

The combined surface tension force is obtained by adding the cohesion and the curvature based force as

$$\mathbf{F}_i^{st} = K_{ij}(\mathbf{F}_i^{coh} + \mathbf{F}_i^{curv}), \tag{6.10}$$

where

$$K_{ij} = \frac{2\rho_0}{\rho_i + \rho_j} \tag{6.11}$$

is a symmetrized correction factor to account for particle deficiencies, e.g., in case of isolated particles or thin features.

Surface Forces

He et al's. [51] surface tension model is solely based on surface energy minimization and is specifically suitable to handle thin features. Similar to CSF models [81], [83], it is based on a color field c that is used to distinguish regions covered by the fluid from others. Usually, c is set to one at the fluid particles and zero everywhere else. With most approaches, the color field is smoothed:

$$c_i^s = \sum_{j=1}^{N} \frac{m_j}{\rho_j} c_i W(||\mathbf{x}_i - \mathbf{x}_j||, h). \tag{6.12}$$

In contrast to the approaches by Müller et al. [83] and Akinci et al. [12], He et al. use the normalized term

$$\nabla c_i^s = \frac{\sum_{j=1}^{N} \frac{m_j}{\rho_j} c_j \nabla W(||\mathbf{x}_i - \mathbf{x}_j||, h)}{\sum_{j=1}^{N} \frac{m_j}{\rho_j} W(||\mathbf{x}_i - \mathbf{x}_j||, h)} \tag{6.13}$$

for the color field gradient to account for particle density underestimation. Using the surface tension energy density $\frac{\kappa}{2}|\nabla c_i^s|^2$, the momentum-conserving surface tension force can be calculated by averaging the energy densities:

$$\mathbf{F}_i^{st} = \frac{\kappa}{2} \sum_{j=1}^{N} \frac{m_i}{\rho_i} \frac{m_j}{\rho_j} \left(\frac{|\nabla c_i|^2 + |\nabla c_j|^2}{2} \right) \nabla W^{st}(||\mathbf{x}_i - \mathbf{x}_j||, h). \tag{6.14}$$

In their work, He et al. [51] also introduce additional air pressure forces without using ghost particles which is not considered in this study.

Modifications to Inter-Particle Interaction Forces

The IIF model of Becker and Teschner [23] is an attractive choice because of its efficiency, simplicity, and easy implementation. With this model however, it is possible that particles group in clusters because attractive forces persist with decreasing distance between particles (see Figure 6.1). The cohesion forces of the model by Akinci et al. [12] eliminate this effect because the forces are modeled as repelling as the distance decreases. If the general model of Becker and Teschner is still preferred, one approach is to cut the cohesive forces at a certain distance to alleviate the problem of particle clustering (see Figure 6.1).

Recently, Yang et al. [122] presented an efficient approach to model molecular surface tension forces while preventing particle clustering with pairwise interactions, which is not considered in this study.

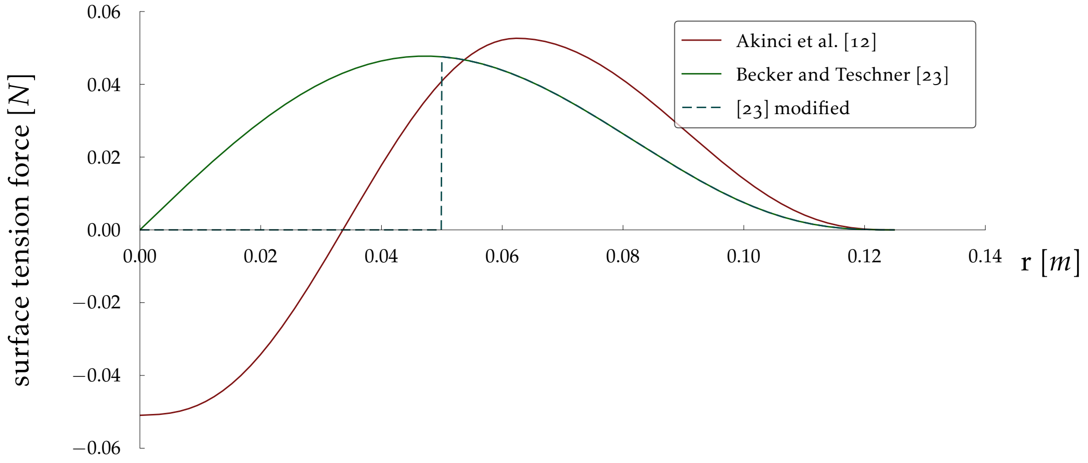


Figure 6.1: Different models of cohesive forces with smoothing length $h = 0.125\,m$. The model by Akinci et al. [12] (red) includes repulsive forces. The model by Becker and Teschner [23] (green) can be modified (blue dashed) by cutting off forces to avoid particle clustering.

6.4 IMPLEMENTATION

As mentioned above, SESPH, PCISPH, and IISPH are used for the simulation of fluids. If not stated otherwise, the SPH kernel as proposed in the work of Müller et al. [83] is used (Equation 2.39), also for surface tension calculations. In all simulation systems, negative pressure values are clipped to avoid attracting pressure forces. Viscosity forces are evaluated with the SPH approximation given in Equation 2.38. As proposed by Akinci et al. [12], viscosity forces are multiplied with the correction term in Equation 6.11 to account for particle deficiencies.

For SESPH, pressure is calculated using the EOS (Equation 2.35). The pressure constant k is evaluated according to Monaghan [80] using $k = |\mathbf{v}|/\eta$, allowing a maximum velocity of $|\mathbf{v}| = 100\,m/s$ and a density fluctuation of $\eta = 0.01$. For both iterative incompressible solvers (PCISPH and IISPH), a maximal compression of 1 % is allowed.

Boundaries, such as container walls or interacting objects, are sampled with particles and the boundary handling method by Akinci et al. [14] is employed.

For the renderings, the level-set technique by Bhattacharya et al. [27] is used to extract the surface for the liquid animations.

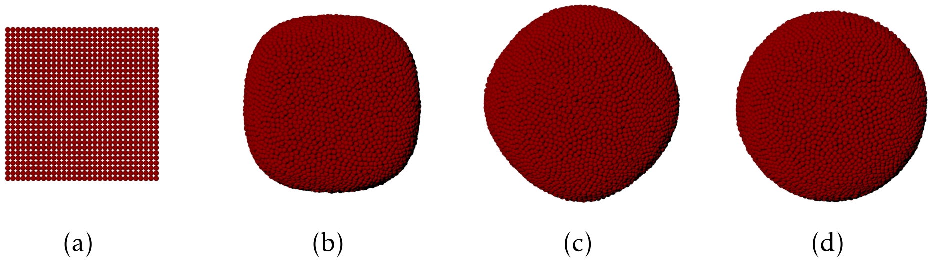


Figure 6.2: Snapshots of an example sequence of benchmark test 1 (Section 6.5.1). Starting from an initial cubic arrangement of particles (a), a spherical drop (d) is formed. In this sequence, the surface tension model by Akinci et al. [12] with $\gamma = 1.0$ is used in combination with IISPH.

6.5 BENCHMARK TEST FOR SURFACE TENSION MODELS

The proposed benchmark tests consist of three typical scenarios that cover settings with high curvature on relatively small surface areas, as well as larger free surfaces.

6.5.1 Test 1: Drop Formation

$27k$ particles are initially arranged in a cube ($30 \times 30 \times 30$ particles) as shown in Figure 6.2(a) and there are no other forces than surface tension forces acting, including gravity. Upon simulation start, the particles should retain a spherically drop form due to surface tension (Figure 6.2(d)). Generally, surface tension forces act to minimize the surface area toward the inside of the sphere. Inside the liquid, pressure forces counteract against these forces until an equilibrium state is reached. This scenario is especially suited to closely observe the interplay between the different types of forces, as there are no external forces present and no interactions with other objects occur that would require explicit boundary handling.

For this scenario, the particle size is set to $0.025\,m$ and the particles are initially arranged with a distance of $0.05\,m$. The smoothing length h is set to $0.125\,m$, a low viscosity coefficient of $\nu = 0.01$ is used, and the simulation step size is $\Delta t = 0.001\,s$.

For the evaluation of the properties of surface tension models, the process of drop formation is analyzed. Besides visual inspection of the animation sequences, average particle velocities, surface tension forces, and pressure forces are measured each time step. Absolute values and their change over time of these important quantities are further analyzed.

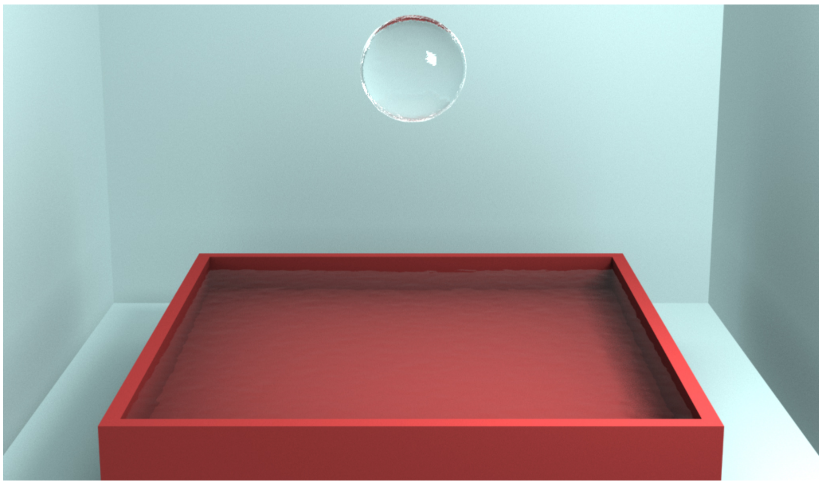


Figure 6.3: Initial setup of benchmark test 2 (Section 6.5.2): a spherical drop is placed over a container of liquid.

6.5.2 Test 2: Liquid Crown

The spherical drop as obtained from the first test with $27k$ particles is initially placed over a container of liquid consisting of 634,980 particles. In a preprocessing step, the drop as well as the liquid in the container are simulated until an equilibrium state was achieved. Under the influence of gravity, the drop falls into the liquid, and a liquid crown will develop on impact. Surface tension influences the shape of the crown and the thin features and smaller droplets that dissolve. The parameters for this scenario are $h = 0.1\,m$, $\nu = 0.01$, and $\Delta t = 0.001\,s$.

In this case, average particle velocities are measured for the analysis of properties of the different surface tension models in a highly dynamic scenario.

6.5.3 Test 3: Water Glass

A liquid consisting of $400k$ particles is poured into a glass sampled with $200k$ particles as an example of a highly dynamic scene with a complex interaction object in a practical application. The particles are initially placed on top of a dipping channel located above the glass. The choice of the surface tension model influences characteristics and form of the jet. Also, the fluid's behavior upon impact of the liquid on the inside of the glass depends on surface tension effects. Snapshots of the animations are shown in Figure 6.10 and are discussed in Section 6.6.3. Here, the parameters are $h = 0.1\,m$, $\nu = 0.01$, and $\Delta t = 0.001\,s$.

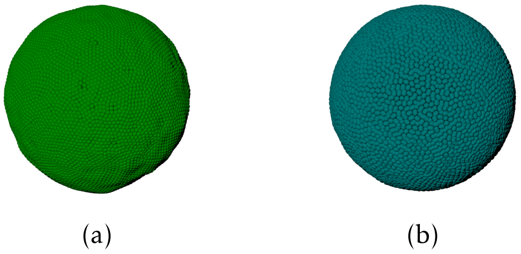


Figure 6.4: Comparison of the surface tension model by Becker and Teschner [23] with the modification of this model. With a surface tension coefficient of $\varphi = 0.08$ in combination with IISPH, the sphere is slightly deformed using their model (a). With the proposed modifications (b), an improved spherical shape is achieved.

6.6 EVALUATION SURFACE TENSION MODELS

The benchmark tests of the previous section are performed with the three surface tension models specified in Section 6.3.2 in combination with SESPH, PCISPH, and IISPH respectively, resulting in up to twelve possible configurations. Configurations and corresponding parameters are given in the following sections.

In the following images and plots, the surface tension model by Akinci et al. [12] are color-coded with shades of red, the model of Becker and Teschner [23] with shades of green, the model of He et al. [51] using shades of brown, and the modifications of the model of Becker and Teschner are represented in shades of blue.

6.6.1 Benchmark Test 1

Benchmark test 1 (Section 6.5.1) is applied to all nine configurations of SPH and surface tension models. Additionally, the simulations are conducted with five different parameters for the surface tension models (Section 6.3.2), respectively, resulting in 60 simulation runs. Unfortunately, the surface tension parameters for the different models have each a different physical meaning and hence, direct comparison is not possible. Therefore, the parameter values are chosen equally spaced for each model, covering a range from low surface tension that slightly affects the shape of the fluid surface, to a very high value that has a major effect while achieving stable simulations. For both the model of Becker and Teschner [23] and the proposed modification, the set of surface tension coefficients is $\varphi = \{0.02, 0.035, 0.05, 0.065, 0.08\}$, for the model of Akinci et al. [12], the coefficients are $\gamma = \{0.2, 0.4, 0.6, 0.8, 1.0\}$, and for He et al. [51], $\kappa = \{0.2, 0.6, 1.0, 1.4, 1.8\}$. The respective units for the different coefficients are omitted under the assumption that length is given in m and time in s.

For a first visual inspection, snapshots of the equilibrium states of all sequences are shown in Figures 6.5, 6.6, and 6.7. Generally, as surface tension is applied, the particles move toward a spherical shape for all different models until an equilibrium state is reached. It has to be noted that the particles do not come to a rest state in this equilibrium, as surface tension forces work against pressure forces. Depending on the surface tension model and parameter value, particles slightly move around, but the overall shape of the fluid is maintained.

Using IISPH (Figure 6.5) and PCISPH (Figure 6.6), a spherical shape is achieved with all models in most cases. Depending on the surface tension coefficient, the process is faster with higher values. However, using the model of He et al. [51] with a low surface tension parameter, the surface tension forces are not large enough to form a sphere. As also discussed by Akinci et al. [12], with the model of Becker and Teschner [23], particles tend to cluster as attracting forces are acting as particle move close to each other (see Figure 6.1). In Figure 6.4, the modification to the model of Becker and Teschner alleviates particle clustering and improves the quality of the sphere. Moreover, it is possible to use higher surface tension coefficients compared to the original model while maintaining an undeformed spherical shape.

Another observation is that the different surface tension models have a different behavior regarding the convergence to the sphere shape. With the models of Becker and Teschner [23] and Akinci et al. [12], the final shape of the equilibrium state is reached within a short period of time. It is noticeable that the drop oscillates in the first few frames with the model of Akinci et al., as opposed to the other models. In contrast, with the model of He et al. [51], the process of drop formation takes considerably longer regarding simulation time.

The observed visual characteristics in the drop formation process can also be identified in measurements of particle velocities. In all cases, the average velocity of the particles is converging to a certain value. Depending on the surface tension parameter, the equilibrium velocity is higher with a bigger parameter. For each model, velocities, as well as surface tension forces, scale almost linearly with the surface tension parameter in this test. However, it has to be noted that this equilibrium velocity also depends on parameters of the underlying SPH simulation, e.g., the smoothing length of the SPH kernel. For each combination of SPH and surface tension model, the aggregated values of the average velocities are calculated using

$$v^{aggr} = \sum_{i=1}^{P} \left(\sum_{j=0}^{M} ||\mathbf{v}_j|| / M \right) / P, \tag{6.15}$$

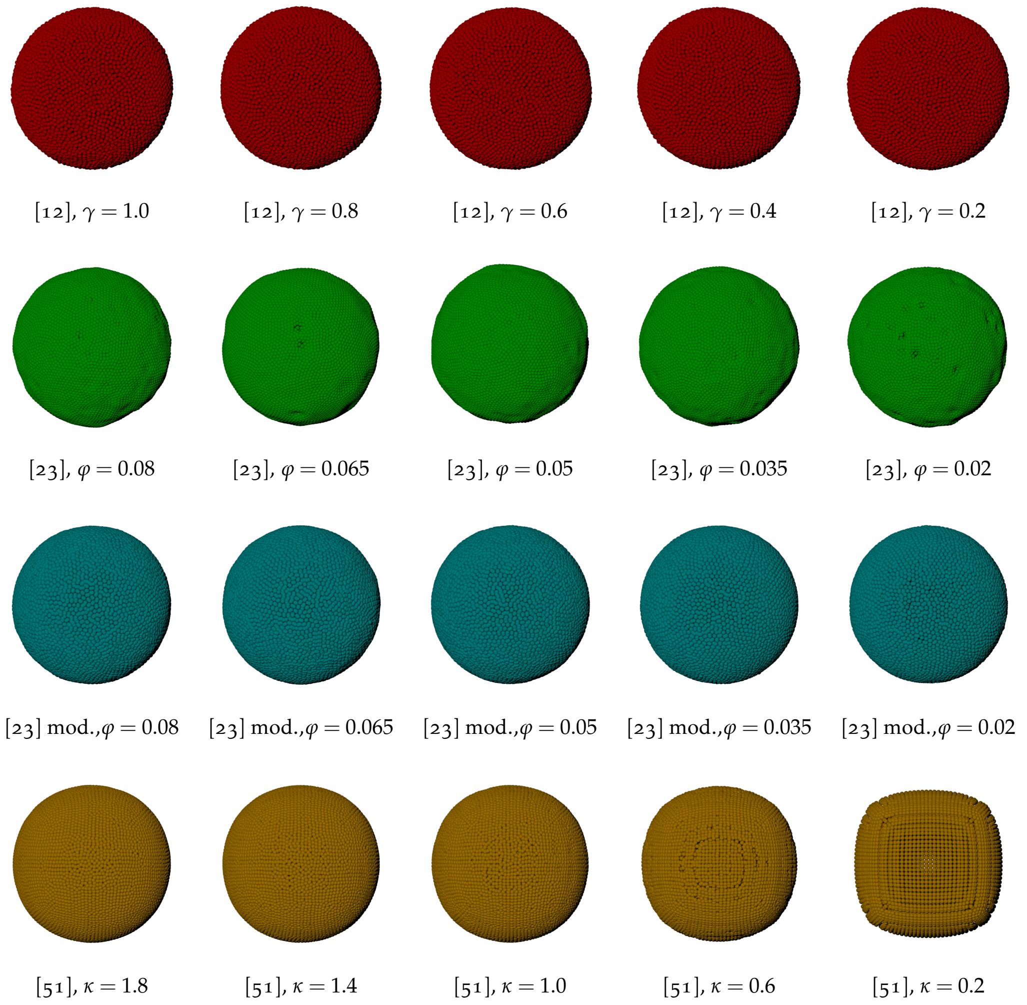
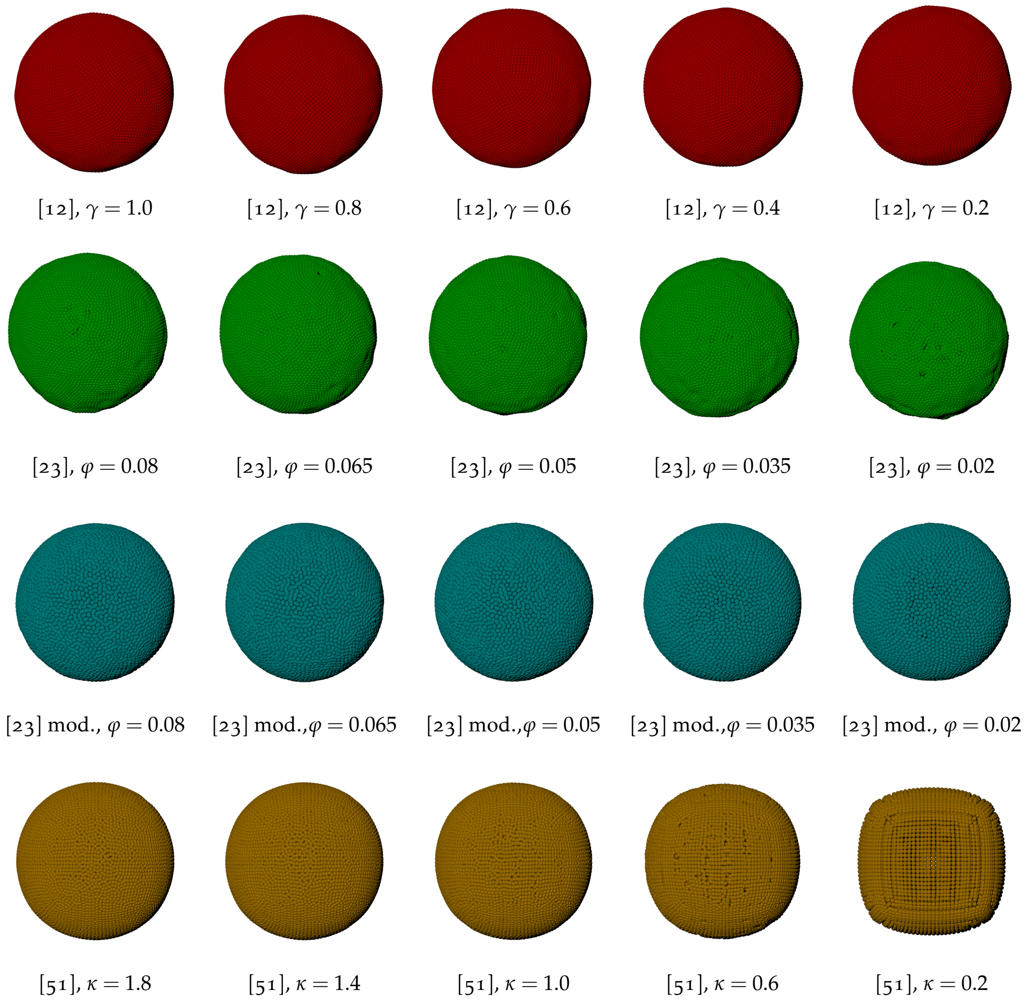


Figure 6.5: Snapshots of the equilibrium state of benchmark 1 applied to all surface tension models in combination with IISPH. Except for low surface tension coefficients with the model of He et al. [51], a spherical shape is achieved with all combinations of models. With the surface tension model of Becker and Teschner [23], deformations of the sphere occur that are resolved with the modifications.



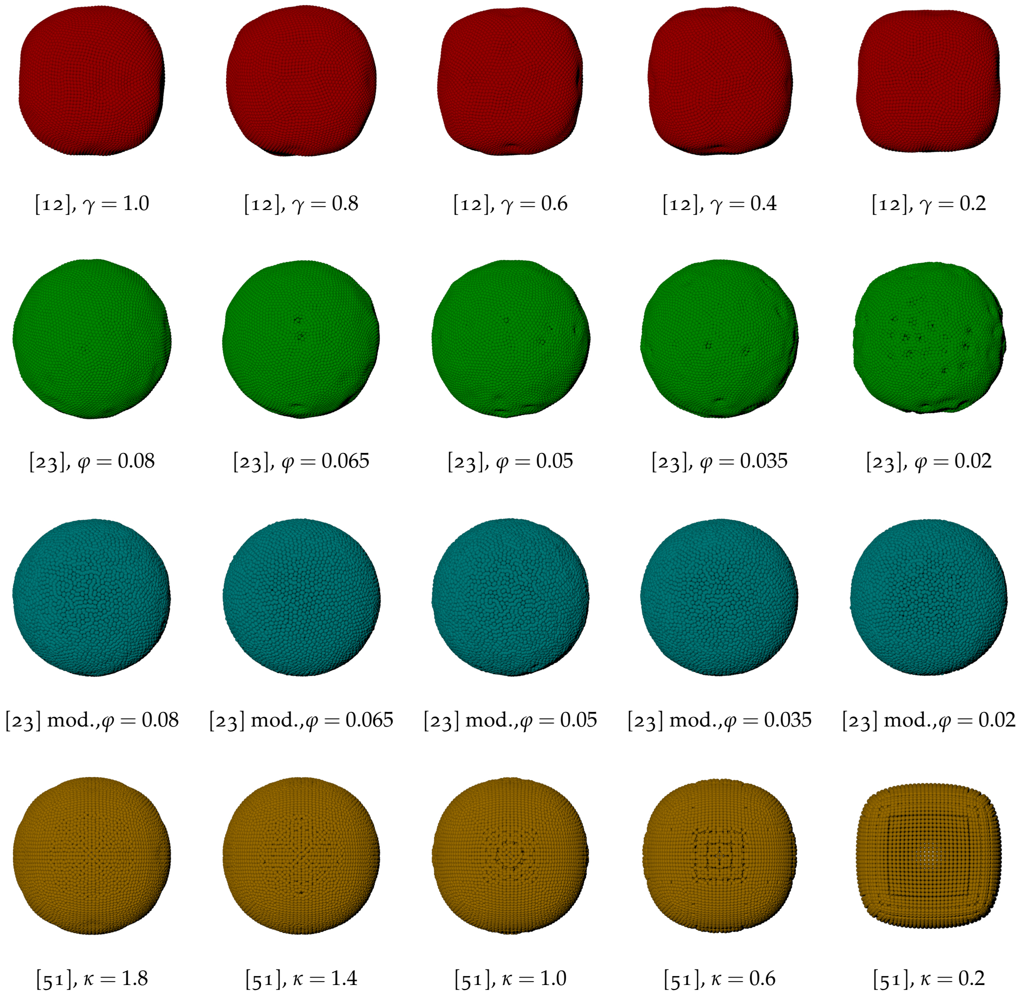
[12], $\gamma = 1.0$ [12], $\gamma = 0.8$ [12], $\gamma = 0.6$ [12], $\gamma = 0.4$ [12], $\gamma = 0.2$

[23], $\varphi = 0.08$ [23], $\varphi = 0.065$ [23], $\varphi = 0.05$ [23], $\varphi = 0.035$ [23], $\varphi = 0.02$

[23] mod., $\varphi = 0.08$ [23] mod.,$\varphi = 0.065$ [23] mod.,$\varphi = 0.05$ [23] mod.,$\varphi = 0.035$ [23] mod., $\varphi = 0.02$

[51], $\kappa = 1.8$ [51], $\kappa = 1.4$ [51], $\kappa = 1.0$ [51], $\kappa = 0.6$ [51], $\kappa = 0.2$

Figure 6.6: Snapshots of the equilibrium state of benchmark 1 applied to all surface tension models in combination with PCISPH. Results similar to the combination with IISPH (Figure 6.5) are achieved.



[12], $\gamma = 1.0$ [12], $\gamma = 0.8$ [12], $\gamma = 0.6$ [12], $\gamma = 0.4$ [12], $\gamma = 0.2$

[23], $\varphi = 0.08$ [23], $\varphi = 0.065$ [23], $\varphi = 0.05$ [23], $\varphi = 0.035$ [23], $\varphi = 0.02$

[23] mod.,$\varphi = 0.08$ [23] mod.,$\varphi = 0.065$ [23] mod.,$\varphi = 0.05$ [23] mod.,$\varphi = 0.035$ [23] mod.,$\varphi = 0.02$

[51], $\kappa = 1.8$ [51], $\kappa = 1.4$ [51], $\kappa = 1.0$ [51], $\kappa = 0.6$ [51], $\kappa = 0.2$

Figure 6.7: Snapshots of the equilibrium state of benchmark 1 applied to all surface tension models in combination with SESPH. With very low values for the surface tension coefficient, no spherical shape is formed. The results achieved with the model of He et al. [51] show almost no difference in combination with other SPH methods.

where M is the number of frames and the $P = 5$ different surface tension coefficients. In Figure 6.8(a), these aggregated values of the average velocities are shown. Regarding the different surface tension models, the method of Akinci et al. [12] results in much higher velocities, whereas with the model of He et al. [51], velocities are up to a factor of 100 lower and the process to a sphere takes much longer. With the modified model of Becker and Teschner [23], lower end velocities are achieved.

Regarding varying SPH models, it can be observed that at least for the combination with the surface tension models using inter-particle forces, the resulting velocities are noticeably higher than with other solvers. Contrary to the IIF models, the method by He et al. [51] is hardly influenced by the SPH method.

In this case, four local maximum values can be identified. In comparison, the models of Becker and Teschner [23] and He et al. [51] have only one local maximum that reflects that no oscillations occur.

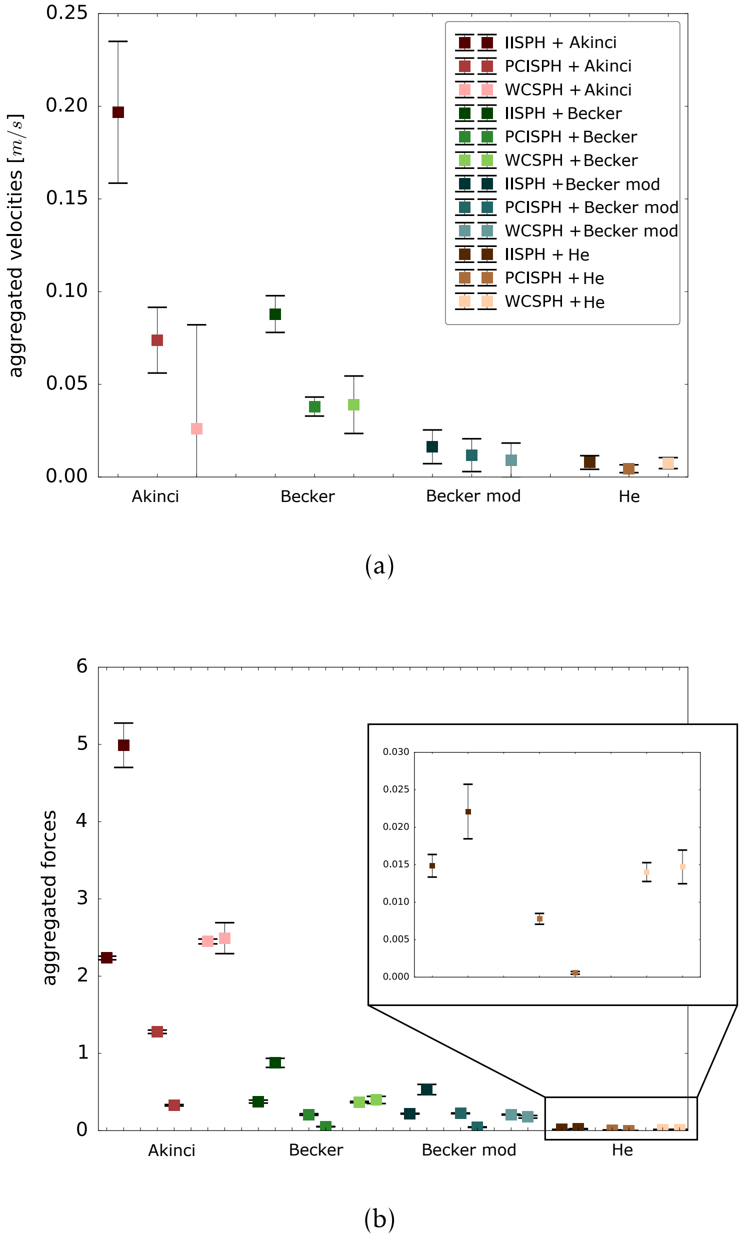
In Figure 6.8(b), aggregated surface tension forces and pressure forces, calculated similar to Equation 6.15, are shown. As expected, increasing surface tension parameter values results in higher surface tension forces, and the pressure forces that act opposite to the surface tension force, also increase. Comparing the different models, it is noticeable that the surface tension forces with the model of Akinci et al. are generally larger than with the other models (see Figure 6.8(b)). Especially with the model of He et al. [51], surface tension forces are considerably smaller, which explains the slower convergence of the particles velocities.

Another observation using SESPH and PCISPH is that pressure forces are lower compared to IISPH. Especially with SESPH, this prevents the sphere forming process with low surface tension coefficient.

6.6.2 Benchmark Test 2

The benchmark test 2 (Section 6.5.2) is applied to all surface tension models in combination with IISPH and SESPH. As the main characteristics of the configurations should be identified, the lowest and the highest values in the sets of surface tension coefficients of the first test are used. Therefore, 16 different simulation runs are analyzed.

In Figures 6.9(a)-6.9(d), snapshots of the liquid crown for the simulation models in combination with IISPH are shown. The shapes of the crown differ considerably at the same frame using different simulation models. Especially with a high surface tension coefficient, the approach by Akinci et al. [12] produces a smooth, flat crown shape where thin features are preserved. With the approach based on inter-particle interaction forces, the crown dissolves in many droplets and there are only minor differences visible between large and



(a)

(b)

Figure 6.8: (a) Aggregated average velocities with standard deviations for the simulations of benchmark test 1 with all configurations calculated using Equation 6.15. For each surface tension model, aggregated velocities are shown in combination with each SPH model, which approximately correspond to the equilibrium velocities. (b) Aggregated forces in N with standard deviations for the simulations of benchmark test 1 with all configurations. For each combination of SPH method, aggregated surface tension forces $\mathbf{F}^{st}$ and pressure forces $\mathbf{F}^{p}$ are depicted in the same color, side by side.

small surface tension coefficients. Shown in Figure 6.9(d), the model by He et al. [51] also leads to a smooth crown shape preserving thin features. As with all surface tension models, the height and slope of the crown increase with a lower surface tension coefficient. Using SESPH (Figures 6.9(e)-6.9(h)), similar effects can be observed. However, the shape of the crown is less extensive in all cases and fewer isolated particles exist at this frame. For the method of Becker and Teschner [23], differences between the surface tension coefficient are more obvious in the resulting animation.

Although there are considerable differences visually in the animations, the differences in average particle velocities between the small and the large surface tension coefficient are only marginal with all combinations and only differ slightly between the individual configurations. Generally, surface tension models smooth the velocities and high frequency oscillations disappear with an increased surface tension coefficient.

6.6.3 Benchmark Test 3

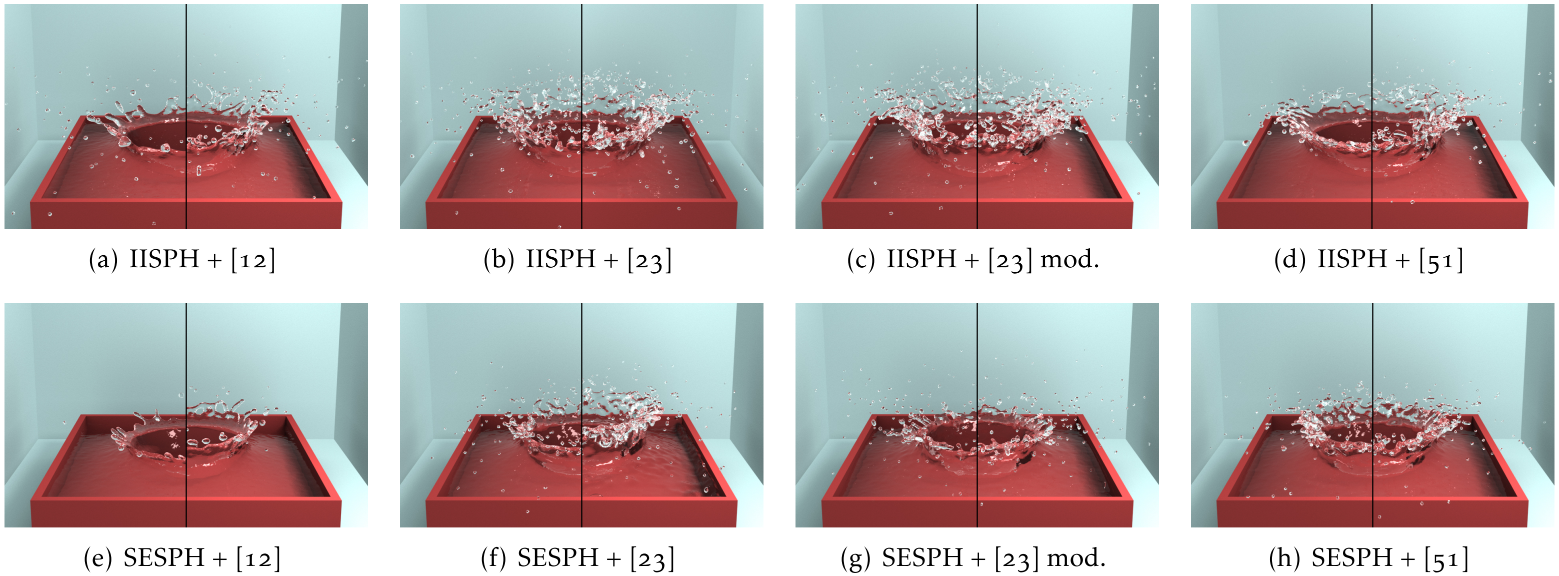
As surface tension effects become less prominent with larger free surface areas, only the largest coefficients of the surface tension models in combination with IISPH is evaluated to illustrate the impact in case of benchmark test 3 (Section 6.5.3).

Again, snapshots of simulations using different surface tension models are shown for comparison in Figure 6.10. As the liquid is poured into the glass, the shape of the liquid using the model by Akinci et al. [12] differs substantially from the shapes resulting from the other models: a smooth continuous jet is formed, whereas with the other methods, many small droplets detach from the liquid jet. The animations created with the surface tension models of Becker and Teschner [23] and He et al. [51] show only minor differences. As in test 2, there are almost no noticeable differences in average particle velocities in this scenario.

6.6.4 Runtime Analysis

For runtime analysis, one second simulation time of the sequence of benchmark test 1 (Section 6.5.1) was simulated with the different classes of surface tension models combined with the IISPH model. All simulations were performed on a standard workstation with an Intel Core i7-3770 processor at 3.4 GHz and 32 GB RAM.

In Table 6.1, timings are given for the computation of surface tension forces, pressure forces, and overall forces computation. The calculation of surface tension forces is primarily coupled to the number of iterations over all particles that have to be performed. As only one iteration is needed for the surface



(a) IISPH + [12]

(b) IISPH + [23]

(c) IISPH + [23] mod.

(d) IISPH + [51]

(e) SESPH + [12]

(f) SESPH + [23]

(g) SESPH + [23] mod.

(h) SESPH + [51]

Figure 6.9: Snapshots of the animations as described in benchmark test 2 (Section 6.5.2) applied to different surface tension models in combination with IISPH (a)-(d) and SESPH (e)-(h). In each image, the largest surface coefficient is shown on the left side, the lowest on the right side. With SESPH, the height of the crown is lower and there are fewer droplets.

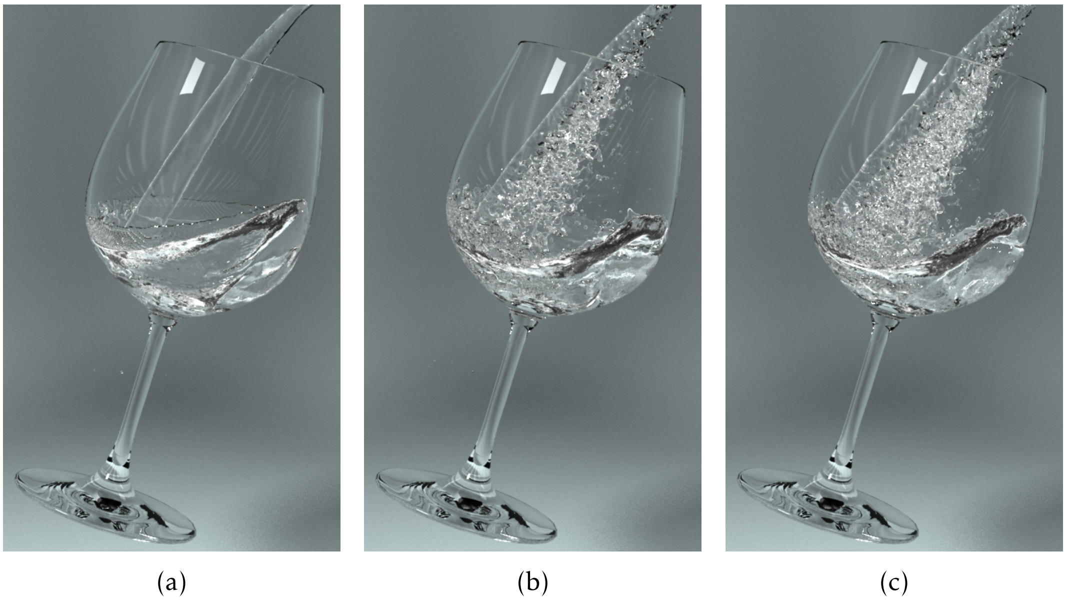


Figure 6.10: Liquid pouring into a glass using IISPH in combination with different surface tension models. (a) With the model of Akinci et al. [12] ($\gamma = 1.0$), a smooth continuous jet of liquid is preserved, whereas (b) with the models of Becker and Teschner [23] ($\varphi = 0.08$) and (c) He et al. [51] ($\kappa = 1.0$) many droplets emerge.

tension model of Becker and Teschner [23], the fastest computation times for the evaluation of surface tension forces are achieved for these models. The proposed modification uses the same approach and no fundamental changes are necessary in the implementation, and therefore, similar runtimes are achieved. For the approach of Akinci et al. [12], two iterations over all particles are necessary, which leads to higher computation times compared to Becker and Teschner. In this implementation, three iterations over all particles are necessary using the model of He et al. [51], as a separate iteration for the calculation of the color field (Equation 6.12) is used, resulting in the highest computation times.

Besides differences between the individual surface tension models, there is no noticeable impact of the value of the surface tension coefficent on the performance of a model. Also, for all simulation models, only low deviations in the calculation of the pressure force calculation are observed.

Table 6.1: Timings in s for a one second simulation time of the benchmark test 1 (drop formation) using the different surface tension (ST) models combined with IISPH. Runtimes are given for the calculation of surface tension forces $\mathbf{F}^{st}$, pressure forces $\mathbf{F}^{p}$, and overall force computation $\mathbf{F}^{t}$.

ST model	ST coefficient	$\mathbf{F}^{st}$	$\mathbf{F}^{p}$	$\mathbf{F}^{t}$
Akinci et al. [12]	$\gamma = 0.2$	26.89	97.48	310.00
Akinci et al. [12]	$\gamma = 1.0$	26.49	106.30	313.00
Becker and Teschner [23]	$\varphi = 0.02$	16.80	98.32	307.10
Becker and Teschner [23]	$\varphi = 0.08$	16.23	93.76	298.23
He et al. [51]	$\kappa = 0.2$	54.79	88.19	329.73
He et al. [51]	$\kappa = 1.8$	52.82	95.48	339.83

6.7 DISCUSSION

Performing the proposed tests on different surface tension models across different SPH implementations reveals that the surface tension models have considerably varying characteristics. These differences not only affect the visual results, but also the physical quantities of the simulations. In addition, the underlying SPH implementation in combination with the surface tension model affects the overall behavior of the simulation and leads to a wide range of factors steering the outcome of the animation. The manifold influences have to be considered for producing artistically controlled simulations bearing a desired look and have a major effect on the process of developing surface tension models.

The results of Section 6.6.1 reveal that the convergence behavior in the process of the drop formation varies greatly for the tested surface tension models. Despite the fact that all of the models generate a spherical shape of the drop with an appropriate surface tension coefficient, the speed and temporal behavior are different. On the one hand, the absolute value of the surface tension force affects the threshold when the final shape is reached. With large surface tension forces exhibited by the models of Becker and Teschner [23] and Akinci et al. [12], it occurs much earlier than with the noticeably smaller surface tension forces of the model by He et al. [51]. On the other hand, smaller forces lead to a much steadier equilibrium state. These smaller forces, however, do not offer the possibility of an oscillating drop as produced by the model of Akinci et al. with a large surface tension coefficient, which rather coincides with the observed physical behavior of a real water drop in absence of gravity.

Especially with the surface tension models by Akinci et al. [12] and Becker and Teschner [23], a dissipative effect when using SESPH is perceivable. The lower pressure forces that counteract surface tension forces lead to lower velocities in comparison with IISPH and PCISPH, as it can be observed in benchmark test 1.

This effect is expected to be more distinct when using a lower EOS constant.

There is a large space of possible animations that not only depends on a parameter, but on the interplay of simulation and surface tension models. The individual approaches to surface tension (Section 6.3.2) reveal considerable differences. Hence, from a production point of view, the choice of a surface tension model and the corresponding coefficient strongly depends on the objective:

- Should animations exhibit a behavior that recreates effects observable in reality, such as the oscillation of a drop, the model of Akinci et al. [12] is especially suitable.
- Should thin features be preserved in combination with smooth surfaces, the model of He et al. [51] produces pleasing results as well.
- For a compromise between computational efficiency and plausible surface tension effects at a small scale, the method of Becker and Teschner [23], especially with the proposed modification, is a good candidate.

The proposed testing environment not only helps regarding the comparability of surface tension models, but represents a useful set of tests that can be used in the development of new surface tension models. Using the benchmark, new models can be directly compared to existing approaches and several properties appear immediately.

6.8 SUMMARY

This chapter presented a systematic evaluation of surface tension models for SPH-based fluid animations. To this end, a benchmark test consisting of three scenarios and selected measurements has been proposed. The uniform setup of the tests not only allow a consistent and reproducible comparison of surface tension models, but it is also suitable to identify the specific characteristics of these models, in particular in combination with varying SPH approaches. Besides a visual comparison on a standardized basis, the proposed measurements allow for quantitative analysis for the examination of configurations from an animation point of view.

Using uniform specifications, some of the specific properties of surface tension models could be identified in a comparative manner. It is also possible to apply the benchmarks to newly developed algorithms to be able to see the performance compared to existing methods.

By providing source code to the benchmarks, researchers will be able to compare their algorithms to the findings in this chapter.

CHAPTER 7

Conclusion

This thesis presented methods for shape description of cloth animations, interaction of fluids with deformable surfaces, and evaluations of surface representations and surface effects in the field of physically-based simulation.

In Chapter 3, a motion-shape signature for cloth animations was presented. The proposed signature is represented by a feature vector that includes the geometry of the underlying mesh as well as measured physical quantities from the simulation. The attributes in the feature vector are stored for sequences of simulation states, and therefore, encode the temporal behavior of cloth animations. By incorporating strain-based and bending-based attributes, the two main deformation modes of cloth are represented in the signature. For each time step in a simulation, spatial aggregation scaled with respect to the deformation element of the individual attributes is performed, so that the metrics are independent from mesh triangulation and resolution. It was shown how to calculate similarities of cloth simulations and the setup was applied to the retrieval of animations from simulation collections by example.

One of the main benefits of the motion-shape signature is that measuring similarity of cloth simulations can easily be performed by calculating the distance between feature vectors. The approach is tailored to the spatiotemporal characteristics of cloth animations and captures the complex shapes of deforming textiles. If a specific goal for an animation is given, the presented approach enables matching of the example with a collection of simulations that, e.g., have been generated by a batch process sampling the parameter space. The task of matching can therefore be performed fully automatically and can assist or even replace difficult and time-consuming visual assessment. Especially with large amounts of simulations, matching can be facilitated and accelerated.

For static surfaces, the proposed signature has been compared to classic shape descriptors based on distributions of geometric properties from previous work in the field of object analysis and retrieval [89]. While similarity scores showed similar results, the presented metrics can be considered as more intuitive as they directly represent geometric and physical properties of cloth. The two modes of planar and bending deformation are encoded separately and the user is allowed to set the weights depending on the application. An additional advantage of the presented approach is that, especially for the use in time-dependent simulation data, the attributes can directly be stored in the feature vector as they have already been computed in the simulation. Other shape descriptors have to be calculated for every simulation run, which requires large computational effort.

Chapter 4 presented boundary handling for interaction between SPH-based fluids and mesh-based cloth. The bidirectional interaction between a thin deformable object with fluid particles is modeled with repulsions. To ensure that particles do not penetrate the cloth surface, the approach was extended with a geometric intersection test of trajectories. For tangential boundary conditions, two methods were presented that allow for either modeling of slip or no-slip boundary condition. Both models also support blending between the two opposite boundary conditions and hence allows the user to control the flow behavior at contact. Additionally, it was shown how typical MC-based fluid surface reconstruction can be modified to extract an intersection-free surface at the contact area that adapts to the highly deformable cloth mesh.

The proposed approach for the two-way coupling using impulses and the continuous intersection test allows robust interaction of fluids and cloth without altering the respective simulation systems. In contrast to other approaches that either simulate deformable objects with particle models [59], [67], [102] or use boundary particles on the deformable object [13], [86], [121], the presented method models interaction directly between fluid particles and the simulation mesh of the textile. Hence, particle sampling of the cloth can be omitted, which is particularly beneficial at large planar deformations of cloth when resampling would be necessary. Using the cloth mesh directly also ensures that all details of the thin surface are included in the interaction and temporal coherence of the boundary handling is given.

Further, the techniques of this chapter allow user control at several points of the boundary handling. At cloth–fluid interaction, the surface of the cloth mesh can be modeled by the user with arbitrary thickness from a very thin surface to thick fabrics without changing the simulation of the material and is not bound to the size of fluid particles. For the tangential boundary conditions, it is possible to model the flow behavior beyond slip and no-slip. A user-defined control parameter allows one to influence the particle velocities at contact to achieve the desired result. Further, the modified surface reconstruction adapts the fluid surface to a deformable surface in contact. This enables the user to control the distance between the two objects, which facilitates the process of rendering of close and intersecting surfaces.

In Chapter 5, an evaluation of surface extraction methods was presented in the context of scientific visualization of SPH simulations. For a multiphase simulation of a phase separation process, the interface is reconstructed and the surface area is compared to a power law. It was shown that a distance field approach, a standard technique in computer graphics, performs similar to a dedicated method from the area of scientific visualization in this quantitative test. An important observation was that despite the similar result in surface area, the shapes of the surfaces generated with the respective methods differ. From a computer graphics perspective, the accuracy relating to the surface area is an important basis, and analysis regarding visual quality of differing surface shapes is an interesting topic for further investigation.

Chapter 6 presented a quantitative and qualitative evaluation of surface tension models for SPH-based fluid simulation methods. For this purpose, three benchmark tests to investigate surface tension effects on different scales regarding surface area were presented. Based on these benchmarks, systematic tests using combinations of three different SPH simulation models with three surface tension models were performed to identify the individual properties of the surface tension models. It was shown that including a benchmark in the development process can provide valuable insights into the properties of the models and their interplay with existing methods.

With the proposed benchmark tests and evaluation procedure, it is possible to do a thorough evaluation of a surface tension models and analyze, how they perform in combination with different SPH models. This is especially useful in the development of new techniques by identifying properties and comparing to existing methods. By making the benchmarks and source code publicly available, other researchers are invited to reproduce the results of this chapter and compare their models and implementations.

In summary, the contributions of this thesis regarding spatiotemporal shape description of animated surfaces, versatile and user-controllable coupling of cloth and fluid, and evaluating surface tension and surfaces of SPH-based fluids address some of the challenges in physics-based animation. The motion-shape signature for cloth presented in Chapter 3 facilitates and automates the task of matching animations and is a promising technique in the context of creating goal-oriented animations. Versatile and robust coupling of particle-based fluids with cloth simulations is possible with the novel approach presented in Chapter 4, which combines physically-based methods at boundaries with the possibility for the user to influence the contact behavior. Extended with the creation of contact-aware surfaces, controllable boundary handling is facilitated. Analysis and evaluation of surface representations and surface effects can lead to an improved understanding of the properties of the underlying models and provide valuable insights that support the development of new techniques. Especially for surfaces, this is an important aspect as they are often represented directly, and therefore, play a major role in the visual quality perceived by the viewer.

The presented ideas in this thesis open several directions for future research. For the motion-shape descriptor, various improvements and extensions are possible as discussed in Section 3.6. In particular, other attributes in the feature vector that address accuracy of the matching quality and generalization pose an interesting challenge. To this end, methods that allow validation of the similarity calculations are of particular significance. User-guided creation of simulations by using the motion-shape signature as a measure in an optimization-driven system or using art-directable simulation methods would also be an interesting research direction.

With the presented approach for two-way coupling between cloth and fluids, it is possible to model very small distances between the cloth surface and the particles, and therefore, particles separated by the thin surface can possibly interact with each other in the SPH computation. This is an undesired effect and should be addressed in future work. Further, exploring methods to control the reconstructed fluid surface at contact could be interesting.

For fluid surfaces, extending the evaluation of the representation and underlying models to perceived quality by the consumer is an important topic for future research. For example, the recent perceptual evaluation of liquid simulations by Um et al. [110] already showed some interesting results for perceived accuracy. Further investigations regarding the crucial features for perceived quality is an interesting research direction itself and could provide a basis for novel simulation models.

In summary, there are further challenging research directions for the creation of animated surfaces with physically-based simulation methods. Goal-oriented control of surface features by the underlying simulation models still is an interesting topic. Also the opposite direction, analyzing and measuring surface properties could influence simulation methods to facilitate the creation of animations.

Co-Authored References

[1] M. Huber, B. Eberhardt, and D. Weiskopf, “Cloth-fluid contact”, in *Vision, Modeling & Visualization*, 2013, pp. 81–88.

[2] M. Huber, B. Eberhardt, and D. Weiskopf, “Boundary handling at cloth–fluid contact”, *Computer Graphics Forum*, vol. 34, no. 1, pp. 14–25, 2015.

[3] M. Huber, B. Eberhardt, and D. Weiskopf, “Cloth animation retrieval using a motion-shape signature”, *IEEE Computer Graphics and Applications*, vol. 37, no. 6, pp. 52–64, 2017.

[4] M. Huber, S. Pabst, and W. Straßer, “Wet cloth simulation”, in *ACM SIGGRAPH 2011 Posters*, 2011, 10:1–10:1.

[5] M. Huber, S. Pabst, and W. Straßer, “Wet cloth simulation: A fast simulation framework for liquid diffusion in porous textiles”, in *Proceedings of Computer Graphics International (Workshop on VFX, Computer Animation, and Stereo Movie Generation)*, 2011.

[6] M. Huber, S. Reinhardt, D. Weiskopf, and B. Eberhardt, “Evaluation of surface tension models for SPH-based fluid animations using a benchmark test”, in *Workshop on Virtual Reality Interaction and Physical Simulation*, 2015, pp. 41–50.

[7] M. Krone, M. Huber, K. Scharnowski, M. Hirschler, D. Kauker, G. Reina, U. Nieken, D. Weiskopf, and T. Ertl, “Evaluation of visualizations for interface analysis of SPH”, in *EuroVis - Short Papers*, 2014, pp. 109–113.

[8] S. Reinhardt, M. Huber, O. Dumitrescu, M. Krone, B. Eberhardt, and D. Weiskopf, “Visual debugging of SPH simulations”, in *Proceedings of the International Conference Information Visualisation*, 2017, pp. 117–126.

[9] S. Reinhardt, M. Huber, B. Eberhardt, and D. Weiskopf, “Fully asynchronous SPH simulation”, in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2017, 2:1–2:10.

References

[10] B. Adams, M. Pauly, R. Keiser, and L. J. Guibas, "Adaptively sampled particle fluids", *ACM Transactions on Graphics*, vol. 26, no. 3, 2007.

[11] W. Aigner, S. Miksch, H. Schumann, and C. Tominski, *Visualization of Time-Oriented Data*, First Edition. Springer London, 2011.

[12] N. Akinci, G. Akinci, and M. Teschner, "Versatile surface tension and adhesion for SPH fluids", *ACM Transactions on Graphics*, vol. 32, no. 6, 182:1–182:8, 2013.

[13] N. Akinci, J. Cornelis, G. Akinci, and M. Teschner, "Coupling elastic solids with smoothed particle hydrodynamics fluids", *Computer Animation and Virtual Worlds*, vol. 24, no. 3-4, pp. 195–203, 2013.

[14] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner, "Versatile rigid-fluid coupling for incompressible SPH", *ACM Transactions on Graphics*, vol. 31, no. 4, 62:1–62:8, 2012.

[15] C. Aliaga, C. O'Sullivan, D. Gutierrez, and R. Tamstorf, "Sackcloth or silk? the impact of appearance vs dynamics on the perception of animated cloth", in *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception*, 2015, pp. 41–46.

[16] J. Anderson, *Computational Fluid Dynamics: the Basics with Applications*, First Edition. McGraw-Hill, 1995.

[17] G. B. Arfken and H. J. Weber, *Mathematical Methods for Physicists*, Sixth Edition. Elsevier Academic Press, 2005.

[18] S. Band, C. Gissler, and M. Teschner, "Moving least squares boundaries for SPH fluids", in *Workshop on Virtual Reality Interaction and Physical Simulation*, 2017, pp. 21–28.

[19] D. Baraff and A. Witkin, "Large steps in cloth simulation", in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, 1998, pp. 43–54.

[20] A. Bartle, A. Sheffer, V. G. Kim, D. M. Kaufman, N. Vining, and F. Berthouzoz, "Physics-driven pattern adjustment for direct 3D garment editing", *ACM Transactions on Graphics*, vol. 35, no. 4, 50:1–50:11, 2016.

[21] G. K. Batchelor, *An Introduction to Fluid Dynamics*, Second Edition. Cambridge University Press, 2000.

[22] G. S. Beavers and D. D. Joseph, "Boundary conditions at a naturally permeable wall", *Journal of Fluid Mechanics*, vol. 30, no. 1, pp. 197–207, 1967.

[23] M. Becker and M. Teschner, "Weakly compressible SPH for free surface flows", in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2007, pp. 9–18.

[24] M. Becker, H. Tessendorf, and M. Teschner, "Direct forcing for Lagrangian rigid-fluid coupling", *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 3, pp. 493–503, 2009.

[25] J. Bender and D. Koschier, "Divergence-free smoothed particle hydrodynamics", in *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2015, pp. 147–155.

[26] M. Bergou, M. Wardetzky, D. Harmon, D. Zorin, and E. Grinspun, "A quadratic bending model for inextensible surfaces", in *Proceedings of the Symposium on Geometry Processing*, 2006, pp. 227–230.

[27] H. Bhattacharya, Y. Gao, and A. W. Bargteil, "A level-set method for skinning animated particle data", *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 3, pp. 315–327, 2015.

[28] J. Bonet and R. D. Wood, *Nonlinear Continuum Mechanics for Finite Element Analysis*, Second Edition. Cambridge University Press, 2008.

[29] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon Mesh Processing*, First Edition. A K Peters/CRC Press, 2010.

[30] J. U. Brackbill, D. B. Kothe, and C. Zemach, "A continuum method for modeling surface tension", *Journal of Computational Physics*, vol. 100, no. 2, pp. 335–354, 1992.

[31] R. Bridson, R. Fedkiw, and J. Anderson, "Robust treatment of collisions, contact and friction for cloth animation", *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 594–603, 2002.

[32] R. Bridson, S. Marino, and R. Fedkiw, "Simulation of clothing with folds and wrinkles", in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003, pp. 28–36.

[33] R. Bridson and M. Müller-Fischer, "Fluid simulation: Siggraph 2007 course notes", in *ACM SIGGRAPH 2007 Courses*, 2007, pp. 1–81.

[34] J. W. Cahn and J. E. Hilliard, "Free energy of a nonuniform system. i. interfacial free energy", *The Journal of Chemical Physics*, vol. 28, no. 2, pp. 258–267, 1958.

[35] Y. Chen, N. Magnenat-Thalmann, and B. Foster Allen, "Physical simulation of wet clothing for virtual humans", *The Visual Computer*, vol. 28, no. 6, pp. 765–774, 2012.

[36] J. Cornelis, M. Ihmsen, A. Peer, and M. Teschner, "IISPH-FLIP for incompressible fluids", *Computer Graphics Forum*, vol. 33, no. 2, pp. 255–262, 2014.

[37] L. D. Cutler, R. Gershbein, X. C. Wang, C. Curtis, E. Maigret, and L. Prasso, "An art-directed wrinkle system for CG character clothing", in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005, pp. 117–126.

[38] M. Desbrun and M.-P. Gascuel, "Smoothed particles: A new paradigm for animating highly deformable bodies", in *Proceedings of the Eurographics Workshop on Computer Animation and Simulation '96*, 1996, pp. 61–76.

[39] P. Du, M. Tang, C. Meng, R. Tong, and L. Lin, "A fluid/cloth coupling method for high velocity collision simulation", in *Proceedings of the ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, 2012, pp. 309–314.

[40] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, "A hybrid particle level set method for improved interface capturing", *Journal of Computational Physics*, vol. 183, no. 1, pp. 83–116, 2002.

[41] O. Etzmuß, M. Keckeisen, and W. Straßer, "A fast finite element solution for cloth modelling", in *Proceedings of Pacific Conference on Computer Graphics and Applications*, 2003, pp. 244–251.

[42] R. Fraedrich, S. Auer, and R. Westermann, "Efficient high-quality volume rendering of SPH data", *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1533–1540, 2010.

[43] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: Theory and application to non-spherical stars", *Monthly Notices of the Royal Astronomical Society*, vol. 181, no. 3, pp. 375–389, 1977.

[44] P. Goswami, P. Schlegel, B. Solenthaler, and R. Pajarola, "Interactive SPH simulation and rendering on the GPU", in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2010, pp. 55–64.

[45] E. Grinspun, A. N. Hirani, M. Desbrun, and P. Schröder, "Discrete shells", in *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003, pp. 62–67.

[46] E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw, “Coupling water and smoke to thin deformable and rigid shells”, *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 973–981, 2005.

[47] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, Second Edition. Springer, 1996.

[48] T. Harada, S. Koshizuka, and Y. Kawaguchi, “Smoothed particle hydrodynamics in complex shapes”, in *Proceedings of the Spring Conference on Computer Graphics*, 2007, pp. 191–197.

[49] M. Hauth, “Visual simulation of deformable models”, PhD thesis, University of Tübingen, 2004.

[50] M. Hauth, O. Etzmuß, and W. Straßer, “Analysis of numerical methods for the simulation of deformable models”, *The Visual Computer*, vol. 19, no. 7, pp. 581–600, 2003.

[51] X. He, H. Wang, F. Zhang, H. Wang, G. Wang, and K. Zhou, “Robust simulation of sparsely sampled thin features in SPH-based free surface flows”, *ACM Transactions on Graphics*, vol. 34, no. 1, 7:1–7:9, 2014.

[52] M. Hirschler, M. Huber, W. Säckel, P. Kunz, and U. Nieken, “An application of the Cahn-Hilliard approach to smoothed particle hydrodynamics”, *Mathematical Problems in Engineering*, vol. 2014, 2014.

[53] P. Huang, J. Starck, and A. Hilton, “Temporal 3D shape matching”, in *Proceedings of the European Conference on Visual Media Production*, 2007, pp. 1–10.

[54] M. Ihmsen, N. Akinci, M. Gissler, and M. Teschner, “Boundary handling and adaptive time-stepping for PCISPH”, in *Workshop in Virtual Reality Interactions and Physical Simulation*, 2010, pp. 79–88.

[55] M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner, “Implicit incompressible SPH”, *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 426–435, 2014.

[56] M. Ihmsen, J. Orthmann, B. Solenthaler, A. Kolb, and M. Teschner, “SPH fluids in computer graphics”, in *EG 2014 - STARs*, 2014, pp. 21–42.

[57] H. W. Jensen, J. Legakis, and J. Dorsey, “Rendering of wet materials”, in *Rendering Techniques ’99*, D. Lischinski and G. W. Larson, Eds., 1999, pp. 273–281.

[58] H. Jeon and M.-H. Choi, “Interactive motion control of deformable objects using localized optimal control”, in *Proceedings of IEEE International Conference on Robotics and Automation*, 2007, pp. 2582–2587.

[59] R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutré, and M. Gross, "A unified Lagrangian approach to solid-fluid animation", in *Proceedings of the Eurographics/IEEE VGTC Symposium on Point-Based Graphics*, 2005, pp. 125–148.

[60] D. Kim, W. Koh, R. Narain, K. Fatahalian, A. Treuille, and J. F. O'Brien, "Near-exhaustive precomputation of secondary cloth effects", *ACM Transactions on Graphics*, vol. 32, no. 4, 87:1–87:8, 2013.

[61] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan, "Efficient collision detection using bounding volume hierarchies of k-dops", *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 1, pp. 21–36, 1998.

[62] A. Kolb and N. Cuntz, "Dynamic particle coupling for GPU-based fluid simulations", in *Proceedings of the Symposium on Simulation Technique*, 2005, pp. 722–727.

[63] D. Koschier and J. Bender, "Density maps for improved SPH boundary handling", in *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2017, 1:1–1:10.

[64] M. Krone, J. Stone, T. Ertl, and K. Schulten, "Fast visualization of Gaussian density surfaces for molecular dynamics and particle system trajectories", in *EuroVis - Short Papers*, 2012.

[65] P. K. Kundu, I. M. Cohen, and D. R. Dowling, *Fluid Mechanics*, Sixth Edition. Academic Press, 2016.

[66] T. Lenaerts, B. Adams, and P. Dutré, "Porous flow in particle-based fluid simulations", *ACM Transactions on Graphics*, vol. 27, no. 3, 49:1–49:8, 2008.

[67] T. Lenaerts and P. Dutré, "Unified SPH model for fluid-shell simulations", Departement of Computer Science, KU Leuven, Report CW 530, Nov. 2008.

[68] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H. V. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, and D. Vandermeulen, "A comparison of methods for non-rigid 3D shape retrieval", *Pattern Recognition*, vol. 46, no. 1, pp. 449–461, 2012.

[69] G. R. Liu and M. Liu, *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific, 2003.

[70] C. T. Loop, "Smooth subdivision surfaces based on triangles", Master's thesis, University of Utah, Aug. 1987.

[71] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3D surface construction algorithm”, *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[72] L. B. Lucy, “A numerical approach to the testing of the fission hypothesis”, *The Astronomical Journal*, vol. 82, pp. 1013–1024, 1977.

[73] G. Luo, F. Cordier, and H. Seo, “Similarity of deforming meshes based on spatio-temporal segmentation”, in *Proceedings of the Eurographics Workshop on 3D Object Retrieval*, 2014, pp. 77–84.

[74] M. Macklin and M. Müller, “Position based fluids”, *ACM Transactions on Graphics*, vol. 32, no. 4, 104:1–104:12, 2013.

[75] S. Martin, B. Thomaszewski, E. Grinspun, and M. Gross, “Example-based elastic materials”, *ACM Transactions on Graphics*, vol. 30, no. 4, 72:1–72:8, 2011.

[76] J. Mezger, B. Thomaszewski, S. Pabst, and W. Straßer, “Interactive physically-based shape editing”, in *Proceedings of the ACM Symposium on Solid and Physical Modeling*, 2008, pp. 79–89.

[77] D. L. Michels, G. A. Sobottka, and A. G. Weber, “Exponential integrators for stiff elastodynamic problems”, *ACM Transactions on Graphics*, vol. 33, no. 1, 7:1–7:20, 2014.

[78] V. Molchanov, A. Fofonov, S. Rosswog, P. Rosenthal, and L. Linsen, “Smoothviz: An interactive visual analysis system for SPH data”, in *Proceedings of the International Smoothed Particle Hydrodynamics European Research Interest Community (SPHERIC) Workshop*, 2013, pp. 350–356.

[79] J. J. Monaghan, “Smoothed particle hydrodynamics”, *Annual Review of Astronomy and Astrophysics*, vol. 30, no. 1, pp. 543–574, 1992.

[80] J. J. Monaghan, “Smoothed particle hydrodynamics”, *Reports on Progress in Physics*, vol. 68, no. 8, pp. 1703–1759, 2005.

[81] J. P. Morris, “Simulating surface tension with smoothed particle hydrodynamics”, *International Journal for Numerical Methods in Fluids*, vol. 33, no. 3, pp. 333–353, 2000.

[82] J. P. Morris, P. J. Fox, and Y. Zhu, “Modeling low Reynolds number incompressible flows using SPH”, *Journal of Computational Physics*, vol. 136, no. 1, pp. 214–226, 1997.

[83] M. Müller, D. Charypar, and M. Gross, “Particle-based fluid simulation for interactive applications”, in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003, pp. 154–159.

[84] M. Müller and N. Chentanez, “Wrinkle meshes”, in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2010, pp. 85–91.

[85] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, "Stable real-time deformations", in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2002, pp. 49–54.

[86] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross, "Interaction of fluids with deformable solids", *Computer Animation and Virtual Worlds*, vol. 15, no. 3-4, pp. 159–171, 2004.

[87] M. Müller, T. Röder, and M. Clausen, "Efficient content-based retrieval of motion capture data", *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 677–685, 2005.

[88] J. Onderik, M. Chládek, and R. Ďurikovič, "SPH with small scale details and improved surface reconstruction", in *Proceedings of the Spring Conference on Computer Graphics*, 2011, pp. 29–36.

[89] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions", *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 807–832, 2002.

[90] S. Patkar and P. Chaudhuri, "Wetting of porous solids", *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 9, pp. 1592–1604, 2013.

[91] T. Popa, Q. Zhou, D. Bradley, V. Kraevoy, H. Fu, A. Sheffer, and W. Heidrich, "Wrinkling captured garments using space-time data-driven deformation", *Computer Graphics Forum*, vol. 28, no. 2, pp. 427–435, 2009.

[92] X. Provot, "Collision and self-collision handling in cloth model dedicated to design garments", in *Computer Animation and Simulation '97: Proceedings of the Eurographics Workshop*. Springer Vienna, 1997, pp. 177–189.

[93] K. Raveendran, C. Wojtan, N. Thuerey, and G. Turk, "Blending liquids", *ACM Transactions on Graphics*, vol. 33, no. 4, 137:1–137:10, 2014.

[94] A. Robinson-Mosher, R. E. English, and R. Fedkiw, "Accurate tangential velocities for solid fluid coupling", in *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2009, pp. 227–236.

[95] A. Robinson-Mosher, T. Shinar, J. Gretarsson, J. Su, and R. Fedkiw, "Two-way coupling of fluids to rigid and deformable solids and shells", *ACM Transactions on Graphics*, vol. 27, no. 3, 46:1–46:9, 2008.

[96] D. Rohmer, T. Popa, M.-P. Cani, S. Hahmann, and A. Sheffer, "Animation wrinkling: Augmenting coarse cloth simulations with realistic-looking wrinkles", *ACM Transactions on Graphics*, vol. 29, no. 6, 157:1–157:8, 2010.

[97] H. Schechter and R. Bridson, "Ghost SPH for animating water", *ACM Transactions on Graphics*, vol. 31, no. 4, 61:1–61:8, 2012.

[98] B. Schindler, R. Fuchs, J. Waser, and R. Peikert, "Marching correctors – fast and precise polygonal isosurfaces of SPH data", in *Proceedings of the International Smoothed Particle Hydrodynamics European Research Interest Community (SPHERIC) Workshop*, 2011, pp. 125–132.

[99] C. Schumacher, B. Thomaszewski, S. Coros, S. Martin, R. Sumner, and M. Gross, "Efficient simulation of example-based materials", in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2012, pp. 1–8.

[100] L. Sigal, M. Mahler, S. Diaz, K. McIntosh, E. Carter, T. Richards, and J. Hodgins, "A perceptual control space for garment simulation", *ACM Transactions on Graphics*, vol. 34, no. 4, 117:1–117:10, 2015.

[101] B. Solenthaler and R. Pajarola, "Predictive-corrective incompressible SPH", *ACM Transactions on Graphics*, vol. 28, no. 3, 40:1–40:6, 2009.

[102] B. Solenthaler, J. Schläfli, and R. Pajarola, "A unified particle model for fluid–solid interactions", *Computer Animation and Virtual Worlds*, vol. 18, no. 1, pp. 69–82, 2007.

[103] G. K. L. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin, "Registration of 3D point clouds and meshes: A survey from rigid to nonrigid", *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 7, pp. 1199–1217, 2013.

[104] J. W. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods", *Multimedia Tools and Applications*, vol. 39, no. 3, pp. 441–471, 2008.

[105] A. Tartakovsky and P. Meakin, "Modeling of surface tension and contact angles with smoothed particle hydrodynamics", *Physical Review E*, vol. 72, no. 2, pp. 026301-1–026301-9, 2005.

[106] M. Teschner, B. Heidelberger, M. Müller, D. Pomeranets, and M. H. Gross, "Optimized spatial hashing for collision detection of deformable objects", in *Vision, Modeling & Visualization*, 2003, pp. 47–54.

[107] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Straßer, and P. Volino, "Collision detection for deformable objects", *Computer Graphics Forum*, vol. 24, no. 1, pp. 61–81, 2005.

[108] B. Thomaszewski, M. Wacker, and W. Straßer, "A consistent bending model for cloth simulation with corotational subdivision finite elements", in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006, pp. 107–116.

[109] B. Thomaszewski, M. Wacker, W. Straßer, E. Lyard, C. Luible, P. Volino, M. Kasap, V. Muggeo, and N. Magnenat-Thalmann, "Advanced topics in virtual garment simulation", in *Eurographics 2007 - Tutorials*, 2007, pp. 795–855.

[110] K. Um, X. Hu, and N. Thuerey, "Perceptual evaluation of liquid simulation methods", *ACM Transaction on Graphics*, vol. 36, no. 4, 143:1–143:12, 2017.

[111] K. Um, T.-Y. Kim, Y. Kwon, and J. Han, "Porous deformable shell simulation with surface water flow and saturation", *Computer Animation and Virtual Worlds*, vol. 24, no. 3-4, pp. 247–254, 2013.

[112] N. Umetani, D. M. Kaufman, T. Igarashi, and E. Grinspun, "Sensitive couture for interactive garment modeling and editing", *ACM Transactions on Graphics*, vol. 30, no. 4, 90:1–90:12, 2011.

[113] P. Volino and N. Magnenat-Thalmann, "Comparing efficiency of integration methods for cloth simulation", in *Proceedings of Computer Graphics International*, 2001, pp. 265–272.

[114] P. Volino and N. Magnenat-Thalmann, "Simple linear bending stiffness in particle systems", in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006, pp. 101–105.

[115] P. Volino, N. Magnenat-Thalmann, and F. Faure, "A simple approach to nonlinear tensile stiffness for accurate cloth simulation", *ACM Transactions on Graphics*, vol. 28, no. 4, 105:1–105:16, 2009.

[116] H. Wang, F. Hecht, R. Ramamoorthi, and J. F. O'Brien, "Example-based wrinkle synthesis for clothing animation", *ACM Transactions on Graphics*, vol. 29, no. 4, 107:1–107:8, 2010.

[117] M. Wardetzky, M. Bergou, D. Harmon, D. Zorin, and E. Grinspun, "Discrete quadratic curvature energies", *Computer Aided Geometric Design*, vol. 24, no. 8, pp. 499–518, 2007.

[118] H. Wendland, "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree", *Advances in Computational Mathematics*, vol. 4, no. 1, pp. 389–396, 1995.

[119] R. White, K. Crane, and D. A. Forsyth, "Capturing and animating occluded cloth", *ACM Transactions on Graphics*, vol. 26, no. 3, 2007.

[120] C. Wojtan, P. J. Mucha, and G. Turk, "Keyframe control of complex particle systems using the adjoint method", in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006, pp. 15–23.

[121] L. Yang, S. Li, A. Hao, and H. Qin, "Realtime two-way coupling of meshless fluids and nonlinear FEM", *Computer Graphics Forum*, vol. 31, no. 7, pp. 2037–2046, 2012.

[122] T. Yang, M. C. Lin, R. R. Martin, J. Chang, and S.-M. Hu, "Versatile interactions at interfaces for SPH-based simulations", in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2016, pp. 57–66.

[123] J. Yu and G. Turk, "Reconstructing surfaces of particle-based fluids using anisotropic kernels", *ACM Transactions on Graphics*, vol. 32, no. 1, 5:1–5:12, 2013.

[124] J. Yu, C. Wojtan, G. Turk, and C. Yap, "Explicit mesh surfaces for particle based fluids", *Computer Graphics Forum*, vol. 31, no. 2pt4, pp. 815–824, 2012.

[125] Y. Zhu and R. Bridson, "Animating sand as a fluid", *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 965–972, 2005.

[126] O. Zienkiewicz and R. L. Taylor, *The Finite Element Method*, Fifth Edition. Butterworth-Heinemann, 2000, vol. 1: The Basis.

[127] J. S. Zurdo, J. P. Brito, and M. A. Otaduy, "Animating wrinkles by example on non-skinned cloth", *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 1, pp. 149–158, 2013.