Visual Prototyping of Cloth

Dissertation

zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Dipl.-Inf. Kai Michael Nikolai Schröder

aus Köln

Bonn, 2013

Universität Bonn
Institut für Informatik II
Friedrich-Ebert-Allee 144, D-53113 Bonn

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

Dekan: Prof. Dr. Ulf-G. Meißner

1. Gutachter: Prof. Dr. Reinhard Klein
2. Gutachter: Prof. Dr.-Ing. Carsten Dachsbacher

Tag der Promotion: 15.12.2014
Erscheinungsjahr: 2015

für Melanie

CONTENTS

ZUSAMMENFASSUNG

Die realistische Visualisierung von Stoffen ist ein wichtiger Anwendungsbereich der Computer Graphik. Eine aktuelle Fragestellung in der Forschung ist, wie man am besten vermessbare Modelle für das Aussehen von Stoffen beschreiben kann. Eine zusätzliche Schwierigkeit tritt dann auf, wenn eine rechnergestützte Konstruktion ermöglicht werden soll. Bisherige Verfahren können zwar dazu genutzt werden, sehr realistische Bilder zu erzeugen, jedoch können die optischen Eigenschaften entweder nur eingeschränkt verändert werden, oder es müssen dafür vorher sehr große Materialdatenbanken vermessen werden.

In dieser Arbeit wird eine computergestützte Pipeline für den Entwurf von Stoffen vorgeschlagen, die direkt auf denjenigen Elementen basiert, die bei der Produktion von Stoffen verändert werden können. Zu diesen gehören die Arten von eingesetzten Fasern und deren optische Eigenschaften, die geometrischen Eigenschaften von Garnen und schließlich deren Zusammensetzung z.B. mittels eines Webmusters.

Dabei schlagen wir ein geometrisches Garnmodell vor, das verschiedene aktuelle Prinzipien aus der Textilienforschung integriert. Weiterhin entwickeln wir ein Verfahren, welches dazu genutzt werden kann, die Parameter dieses Modells anhand eines einzelnen Bildes zu schätzen. Hierbei werden der Verlauf von Garnen, die Garnbreiten, deren Variation und das Webmuster vollautomatisch analysiert. Anhand verschiedener Beispiele demonstrieren wir, dass unser Verfahren in der

Lage ist, das Aussehen im Eingabebild nachzuempfinden. Die Eigenschaften eines so virtuell modellierten Stoffes können beliebig editiert werden. Zum Beispiel kann simuliert werden, wie sich das Aussehen des Stoffes verändert, wenn man andere Fasertypen verwendet, oder die Verzwirnung von Garnen erhöht.

Explizite faser-basierte Modelle können genutzt werden, um eine optische Simulation für kleine Stoffproben vorzunehmen. Jedoch skalieren Verfahren, die auf solchen Modellen beruhen, nicht für größere Proben mit vielen Fasern.
Eine Materialdarstellung, welche in der letzten Zeit erfolgreich für das Erzeugen von photo-realistischen Bildern von virtuellen Stoffen eingesetzt wurde, ist die Bidirektionale Texturfunktion (BTF). Wir stellen eine Methode vor, die die Flexibilität von expliziten Modellen der Mikrogeometrie mit der Effizienz von BTFs bei der Bilderstellung kombiniert. Wir schlagen einen neuen Ansatz für die Berechnung von synthetischen BTFs aus Mikrogeometrie mittels Monte Carlo Pfadverfolgung vor. Wir stellen fest, dass BTFs üblicherweise aus vielen ähnlichen *apparent bidirectional reflectance distribution functions (ABRDFs)* bestehen. Durch das Ausnutzen von strukturellen Selbstähnlichkeiten können wir die Berechnungszeit um eine Größenordnung reduzieren. Dies ist durch eine Technik möglich, die wir *nicht-lokale Bildrekonstruktion* nennen, welche auf nicht-lokalen Mittelwerten (non-local means) basiert.

Abschließend schlagen wir einen neuen generellen Ansatz für die physikalisch akkurate Bilderzeugung bei großen Stoffproben vor. Durch das Nutzen eines statistischen volumetrischen Modells, welches die Verteilung der Fasern beschreibt, können wir eine unerschwinglich teure explizite Speicherung jeder einzelnen Faser vermeiden. Dadurch ist es uns möglich, auch Proben darzustellen, die aus um mehrere Größenordnungen mehr Fasern bestehen, als die größten Proben, die wir auf aktueller Desktop-Hardware explizit beschreiben können. Dabei müssen wir kaum auf Flexibilität und Allgemeingültigkeit verzichten.

Abstract

Realistic visualization of cloth has many applications in computer graphics. An ongoing research problem is how to best represent and capture appearance models of cloth, especially when considering computer aided design of cloth. Previous methods can be used to produce highly realistic images, however, possibilities for cloth-editing are either restricted or require the measurement of large material databases to capture all variations of cloth samples.

We propose a pipeline for designing the appearance of cloth directly based on those elements that can be changed within the production process. These are optical properties of fibers, geometrical properties of yarns and compositional elements such as weave patterns.

We introduce a geometric yarn model, integrating state-of-the-art textile research. We further present an approach to reverse engineer cloth and estimate parameters for a procedural cloth model from single images. This includes the automatic estimation of yarn paths, yarn widths, their variation and a weave pattern. We demonstrate that we are able to match the appearance of original cloth samples in an input photograph for several examples. Parameters of our model are fully editable, enabling intuitive appearance design.
Unfortunately, such explicit fiber-based models can only be used to render small cloth samples, due to large storage requirements.

Recently, bidirectional texture functions (BTFs) have become popular for efficient photo-realistic rendering of materials. We present a rendering approach combining the strength of a procedural model of

micro-geometry with the efficiency of BTFs.
We propose a method for the computation of synthetic BTFs using Monte Carlo path tracing of micro-geometry. We observe that BTFs usually consist of many similar apparent bidirectional reflectance distribution functions (ABRDFs). By exploiting structural self-similarity, we can reduce rendering times by one order of magnitude. This is done in a process we call non-local image reconstruction, which has been inspired by non-local means filtering. Our results indicate that synthesizing BTFs is highly practical and may currently only take a few minutes for small BTFs.

We finally propose a novel and general approach to physically accurate rendering of large cloth samples. By using a statistical volumetric model, approximating the distribution of yarn fibers, a prohibitively costly, explicit geometric representation is avoided. As a result, accurate rendering of even large pieces of fabrics becomes practical without sacrificing much generality compared to fiber-based techniques.

ACKNOWLEDGEMENTS

I thank my advisor, Prof. Reinhard Klein, for giving me the possibility to work on an exciting research topic, which allowed me to extend my knowledge in many different fields. I also thank him for supporting my research with interesting insights, teaching me how to present ideas in a publication and providing me with the necessary tools I needed.

I thank the B-IT research school for offering me a scholarship and generously providing an additional funding to be able to finish my thesis with a further publication. I would like to thank the company RTT for supporting parts of this work.

I thank everybody in our working group for interesting discussions and making work a pleasure. A special thanks goes to Dr. Arno Zinke, who always proved to be a great source of inspiration and ideas and who collaborated with me on several publications. I also want to thank Roland Ruiters for all the interesting technical discussions we had, as well as for exchanging many arguments in a fun way.

I thank my parents for their continuous support. Last but not least, I thank my wife Melanie for her understanding and help during the hardest challenges of a PhD thesis.

Part I

Introduction

CHAPTER 1

INTRODUCTION

Cloth belongs to the oldest man-made materials that exist. It is a ubiquitous part of our everyday life and used in a diverse set of contexts ranging from articles of clothing to functional cloth applied for example to furniture or car seats. Therefore, its design is of high

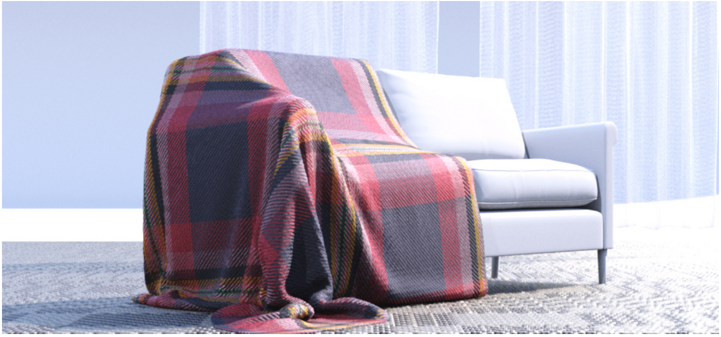


Figure 1.1: *Several large pieces of cloth rendered with Monte Carlo path tracing in less than 50 minutes (for a resolution of* 1000×500 *pixels with 2048 samples per pixel) using a physically-based volumetric approach presented in this work. Three different materials are shown: A hard looking carpet, a soft looking blanket and a translucent curtain. Material properties are linked directly to optical properties of fibers, yarns and the weave pattern.*

importance. On the one hand, the fashion industry aims to create new designs and trends every year. On the other hand, the automobile industry wants to create innovative designs while establishing a certain look and feel for their brand – the choice of cloth in car interiors can make the difference whether a car is perceived as being elegant or sporty.

Computer aided design (CAD) has been invented to improve the speed of innovation and extend the realm of possible designs. Originally used in mechanical engineering, its main goal was to enable the design of the basic shape and form of objects. Visualizations were created so that one could imagine the appearance of an object without having to construct it. Originally, these virtual surrogates focused on basic geometrical properties of objects. Later, important effects of lighting were added to increase the realism. Nowadays, full optical simulations can be computed, predicting the appearance of objects under arbitrary lighting environments. This is possible due to efficient algorithms that have been invented in the field of computer graphics to compute light transport. Recently, even the fine visual details of materials have gained much attention. If we are able to not only change the shape of an object but can also design this fine visual detail, we have full control over the appearance. We call a design process, where we predict the appearance of virtual objects, *visual prototyping*. It allows for applications far beyond mechanical engineering and the approach can in principle be used to design almost any visual aspect of an object.

An enormous effort is currently spent on the design of seat cushions for cars by the automobile industry in a process requiring several iterations. Different variations of a cloth sample are produced and several of them are fully applied to a car seat until a final design decision is found. One of our goals is to be able to replace some of these iterations by the use of visual prototyping.

Ideally, one would like to be able to base editing operations directly on aspects of the physical parameters, a designer of a real material could adjust. These parameters are commonly domain specific and require an understanding of the structure of the material in question. When performing visual prototyping of textiles, we want to be able to select different types of fibers such as synthetics or wool fibers; different types of spun yarns should be available and compositional elements such as weave-patterns should be editable.

To represent the fine visual detail, general appearance models have been proposed that can describe the look of any type of material, invariant to a specific lighting environment. In an ideal world, these models would be directly related to shape and first principles of physics. Even though highly desirable, such a representation is commonly available only for few material types. The acquisition and representation of heterogeneous real world materials is still challenging and often generic image-based techniques such as Bidirectional Texture Functions (BTFs) [DvGNK97] are used to capture complex lighting and shadowing effects. Essentially, a material is photographed many times from different angles with light coming from varying directions. While such measured data can be highly effective for reproducing the appearance of existing material samples, physically plausible editing of purely image-based representations is not easily possible. This is a problem for appearance design and visual prototyping. By focusing on a specific type of material we are able to find more expressive models.

If we want to directly edit parameters for the three inherent scales of cloth (fibers, yarns, compositions), these have to be somehow represented explicitly. In contrast, when they are aggregated in a unified, e.g. image-based, representation, editing can be restricted and difficult.

However, predictive rendering of large cloth samples during the design process still remains a challenging task. Apart from *geometrical complexity*, *optical complexity* presents complications as highly anisotropic single and multiple scattering effects as well as self shadowing effects often dominate the appearance. Many types of fibers are highly translucent and multiple scattering significantly influences the observed color (see Figure 1.2). Since a cloth model may consist of potentially hundreds of millions of fibers, finding a viable level of geometrical abstraction is difficult. While explicit representations work well, when designing details in a close-up view, it is difficult to get an overall impression of a cloth sample as seen from typical viewing distances. More efficient representations are highly desirable. While state-of-the-art methods developed in the field of hair rendering work well when rendering small samples that are explicitly modeled as a fiber assembly, this approach does not scale. We discuss novel approaches for cloth rendering that can handle these challenges. Figure 1.1 shows an image of several large pieces of cloth, rendered using a volumetric approach presented in this thesis.

Although rendering of cloth is a specialized area, we believe that several of the general ideas that are developed in this thesis could be extended to describe other materials with complex micro-geometry as well.

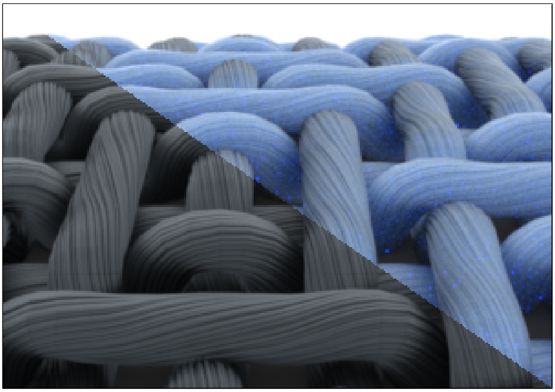


Figure 1.2: *Illustrating the effect of multiple scattering, simulated for a simple cloth model. The lower left part of the image shows only local illumination. The upper right part shows a full global illumination solution. The final color is largely influenced by light scattering inside the yarn.*

1.1 Contributions

The main contributions in this thesis are as follows. We present a pipeline for visual prototyping of woven cloth, in which we

- propose a geometric cloth model that can capture local irregularities and characteristic variations such as local thickening and thinning of yarns over large areas and includes subtle effects such as fly-away hairiness fibers.
- employ a specific fiber scattering model that can be measured for fibers independent from the specific cloth and therefore be replaced for visual prototyping purposes.
- estimate the geometric parameters from one single image only, avoiding the need for creating costly databases of Micro-CT measurements as would be needed by other state-of-the-art methods.

Our resulting model

- clearly separates the different scales of weave level, yarn level and fiber level.
- allows for efficient editing by modifying the corresponding parameters in a way similar to the real production process, in which parameters can be influenced exactly on these levels.
- is able to reproduce the look and feel of the originally photographed material.
- utilizes a novel geometric yarn model that integrates state-of-the-art principles from textile research literature that we introduce to computer graphics, resulting in well-founded model parameters and practical methods for characterizing and measuring these quantities.

To be able to render large cloth models, we propose

- a novel volumetric rendering approach based on a statistical description of micro-geometry.

- a method to efficiently create a synthetic BTF, introducing the new concept of *non-local image reconstruction*, where path tracing samples between different non-adjacent, non-local image pixels are shared that have a similar appearance.

1.2 Outline

In Part I of this thesis, we give background information, discussing related work and describing the structures that constitute the micro-geometry of typical types of cloth and explain how fibers are combined to form yarns and how yarns can be composed to form textiles.

In Part II, we present an approach to automatically reverse engineer woven cloth at the yarn level [SZK15]. This includes the automatic regularization of an input photograph, estimation of the weave pattern and a full segmentation into individual yarns and background. At the same time, we estimate deformation fields, capturing the characteristic look and feel of a cloth sample.

In Part III, we introduce state-of-the-art principles in textile research to computer graphics and propose a procedural cloth model that includes the natural variations of yarns captured from a given sample. Parameters of this model are estimated based on results described in Part I and compared to input photographs. The model is fully editable, allowing for visual prototyping [SZK15].

In Part IV, we present methods for physically-based rendering of cloth to overcome problems of explicitly modeling micro-geometry. We combine the advantages of explicit and image-based representations in a framework to synthesize BTFs. We simulate a virtual measurement device, computing images using Monte Carlo rendering. As rendering tens of thousands of images is costly, we propose a new way to accelerate computations [SMKZ11, SKZ13].
We further present a volumetric approach to render cloth using *virtual scattering events* [SKZ11]: Instead of representing all fibers explicitly,

only their statistical distribution is stored in a volume. Explicit intersection tests of rays with fiber geometry are replaced with a technique that computes locations of intersections by sampling from this statistical distribution. The approach permits using the same descriptions for optical properties of individual fibers as one can use for explicitly modeled micro-geometry. As optical fiber properties are separated from geometric properties, they are independently editable. In this context, we propose a solution for a general problem in volumetric rendering that has not been discussed so far. We introduce the novel concept of local and global visibility. Local visibility is represented by a function we call Bidirectional Visibility Distribution Function (BVDF).

Finally, in Part V we conclude and discuss future work.

CHAPTER 2

BACKGROUND

In this chapter, we give an overview of related work in the context of appearance modeling and we give background information on the elements of cloth and how they are manufactured.

2.1 Related Work

In the following, we describe both general approaches to model the appearance of materials as well as methods specialized to represent cloth.

2.1.1 Appearance Modeling

Optical materials properties essentially describe the appearance of a material under all possible lighting conditions for any arbitrary observer. They are influenced largely by light scattering from micro-geometry and first principles of physics such as absorption or index of refraction. Different approaches exist for modeling optical material properties. The choice of the most appropriate model for a certain material can depend on many different aspects. These can include direct properties of the material itself such as homogeneity, opacity, amount of sub-surface scattering or anisotropy. Sometimes, prior knowledge can be used to select a more efficient representation. This includes knowledge about a typical distance of an observer to the material (scale) or information about the type of lighting, which can for example be either diffuse or be made of strong isolated lights. One can further distinguish between distant and near lights.

Surface Reflectance

Bidirectional Reflectance Distribution Functions (BRDFs) [oSN77] describe how incoming irradiance for an infinitesimal surface patch is transformed to outgoing radiance. For opaque and spatially uniform materials with no time dependency / no phosphorescence and no fluorescence, this information is sufficient to describe light scattering from the surface as long as all scattering happens locally; in other words, when no subsurface-scattering occurs. Many BRDF models have been proposed, starting with phenomenological models [Pho75, Bli77] to others modeling certain physical aspects ([CT82], [LFTG97]). While these models work well for the types of materials they have been designed for, they cannot describe the appearance of arbitrary materials accurately [NDM05]. Therefore, data-driven descriptions [Mat03, RSK12] have been proposed as an alternative. When fitting to measured datasets, these offer more degrees of freedom for describing reflectance at the cost of a possibly more complex description and the danger of over-fitting, when only sparse measurements are available.

To describe non-homogeneous materials, one can use spatially varying BRDF (SVBRDF) descriptions. A problem of SVBRDFs is that they are hard to measure directly as meso-scale surface detail commonly leads to self-shadowing and self-occlusion effects, which – when fitting a BRDF to it – may result in violations of basic assumptions such as reciprocity and energy conservation. Functions that describe such measured data directly at a certain spatial location, are called apparent BRDFs (ABRDFs) [WHON97a]. When storing ABRDFs in a spatially varying form, we call this a Bidirectional Texture Function (BTF) [DVGNK99]. As BTFs are successfully used in many practical applications and as we propose a method to create synthetic BTFs in Chapter 10, we describe them in more detail.

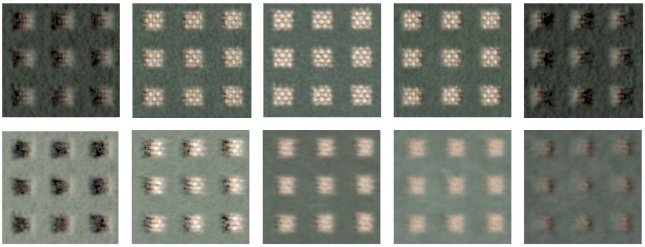


Figure 2.1: *2-dimensional spatial slices through a BTF showing a material sample under several viewing and lighting directions captured using the camera dome of Bonn University.*

Bidirectional Texture Functions (BTFs) BTFs are view- and light dependent textures describing the appearance of a material. Monochromatic BTFs can be represented by a 6-dimensional function

$$\rho : (u, v, \phi_i, \theta_i, \phi_o, \theta_o) \to \mathbb{R}$$

where the angles (ϕ_i, θ_i) represent the direction of an infinitely far away point light, (ϕ_o, θ_o) being the direction of outgoing radiance for an infinitely far away viewer and spatial location (u, v) inside the texture. For spectral rendering, the BTF is parameterized by an additional dimension – the wavelength λ. Figure 2.1 shows spatial slices through a BTF, each representing the appearance of a material for a specific pair of viewing and lighting directions. See [MMS$^+$05, HF11] for excellent surveys on this topic. In the following, we give a short overview:

BTF-measurement devices can capture the appearance of real existing materials. Occlusion and parallax effects, micro-geometry-based self-shadowing and global illumination effects, including local subsurface scattering, are incorporated naturally without the need for any explicit knowledge about these phenomena. This makes BTFs especially attractive if one wants to reproduce complex materials realistically in a virtual setting. They are currently one of the best techniques for reproducing the appearance of measured cloth samples.

BTFs are well suited for many rendering applications. They are known to be efficiently compressible and under certain lighting conditions (i.e. under a few point lights) they can be used for real-time graphics rendering applications even on the web using WebGL [SRWK11].

Unfortunately, BTFs also have inherent limitations: Light diffusion is not modeled properly (especially at shadow boundaries), regions of high curvature are not represented correctly if the BTF is measured with regard to a flat sample, and silhouette information is missing. The quality of highlights strongly depends on the angular resolution of the BTF. Finally, transparency is difficult to measure and rarely considered. Figure 2.2 shows cloth renderings using BTFs.

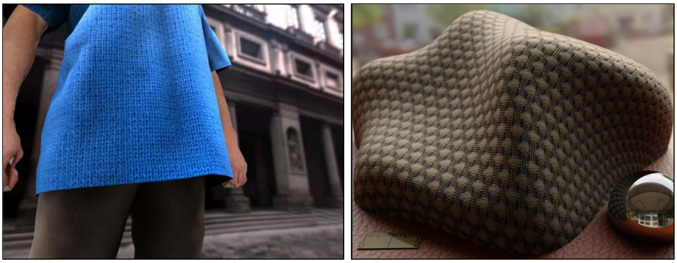


Figure 2.2: *Two renderings of BTFs under environment lighting. Left taken from [MMSK03]. Right, image courtesy of Sensible Graphics GmbH; Material sample by Volkswagen AG.*

Capturing Translucency and Sub-Surface Scattering: The most common use for the models described so far is to describe optical properties for opaque surfaces: light comes from a hemisphere centered around the normal of each point on a surface and is scattered back into that same hemisphere. When we want to model translucency, we can use the Bidirectional Transmittance Distribution Function (BTDF), which describes how light coming from one hemisphere is scattered into the other hemisphere. To describe all light transport for the full sphere, BRDF and BTDF can be combined, giving us the Bidirectional Scattering Distribution Function (BSDF). Extensions to spatially varying descriptions are straight forward, although complications may arise in practice, when measuring such functions.

However, only a limited amount of translucent materials can be described accurately this way as sub-surface scattering is only included in an aggregated form. A more accurate representation can be important in case of thick non-opaque materials such as cloth. The Bidirectional Surface Scattering Distribution Function (BSSRDF) [JMLH01] can describe sub-surface light transport accurately as well. For a pair of points on the surface (p_i, p_o), together with incoming and outgoing direction, it describes the outgoing radiance at point p_o for incoming radiance at point p_i. While BTFs can already describe some amount of local sub-surface scattering in case of infinitely far away directional lighting, BSSRDFs can even model light bleeding if the surface is lit by structured light. However, large storage requirements and complicated measurement setups for BSSRDFs for complex heterogeneous materials are challenging problems.

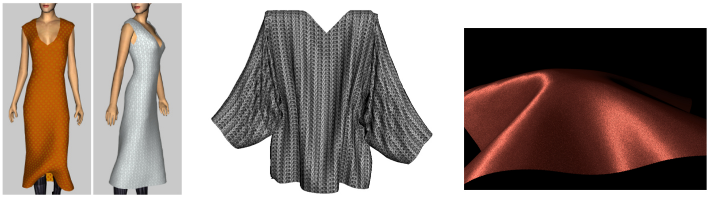
2.1.2 Cloth Rendering

A common way to render cloth is to use a triangle mesh as a base mesh and model the interactions of light with fibers and yarns using surface reflectance models. There are two main directions: The first one uses heuristic methods or procedural modeling to create plausible images. The second one is data-driven and directly based on optical measurements.

Surface-Reflectance-Based Appearance Models for Cloth Daubert et al. [DLH01] model yarns using implicit surfaces and generate a

BTF-like data representation using hardware rendering. They compute lighting using a geometrical model of a stitch. By sampling the stitch regularly within a plane, a view-dependent texture with per-pixel normals and material properties is generated. Adabala et al. [AMTF03] use a simple BRDF model for efficient cloth rendering. They interpret the industry standard *weave information file format (WIF)* to generate reflectance data for arbitrary woven patterns. The appearance of the micro-geometry of spun yarn is visualized by creating a procedural texture with parameters that capture the tightness of twisting and thickness of the thread. An average bidirectional reflectance distribution function (BRDF) is estimated using a micro-facet model which incorporates information from the WIF by analyzing the fraction of the surface that is visible as warp or weft threads. The amount of shadowing that a thread can cast onto neighboring thread facets is captured by a horizon map.
Ray-tracing based methods for modeling cloth at a fiber level have been presented in several works [WAT92, VKKK97]. Their main objective was to use micro-scale simulation to derive a BRDF/BSDF model for a small patch of cloth.

A more sophisticated approach to model the appearance of woven cloth using BRDF and BTF at yarn level was presented by Irawan et al. [Ira07]. The resulting models, which are validated against measurements, yield visually plausible results for a wide range of fabrics. Unfortunately, some of the model parameters are based on ad-hoc assumptions that cannot be directly inferred from optical fiber properties. Sadeghi et al. [SBDDJ13] have proposed a specialized shading model for cloth based on BRDF measurements. Similar to other shading models, this approach cannot separate the optical properties of fibers from other phenomena. Renderings of different surface-reflectance models for cloth are shown in Figure 2.3.



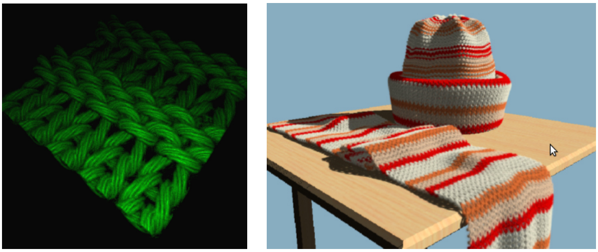
(a) [AMTF03] (b) [DLH01] (c) [Ira07]

Figure 2.3: *Several methods for modeling surface reflectance.*

Data-driven and Image-Based Techniques

Sattler et al. [SSK03] physically acquire a BTF from a rectangular probe and generate a set of view-dependent texture-maps using a principal component analysis of the original data. These maps can even reproduce mesoscopic detail and can then be illuminated and rendered in real-time on graphics hardware. Several variants of this technique have been investigated for realistic cloth rendering [MTCK$^+$04]. A data driven micro-facet-based BRDF approach was taken by Wang et al. [WZT$^+$08]. Here, for each measured surface point, a normal density function model best fitting the observation, is used for rendering anisotropic spatially-varying materials such as cloth.

Volumetric Approaches: Several volumetric techniques have been proposed to represent and render cloth. Early approaches were able to create good looking images of complex pieces of cloth. However, they were not based on physical properties and could therefore not be used for predictive rendering: A volumetric approach for modeling knitwear was proposed by Gröller et al. [GRS95]. By measuring the cross-sectional distribution of yarn fibers, a density field is created and swept along a three dimensional curve to form the entire yarn. A similar idea was presented by Xu et al. [XCL$^+$01], where computations revolve around a structure called lumislice, a light field of a yarn-cross-section.



(a) [GRS95]

(b) [XCL$^+$01]

Figure 2.4: *Two images created using early volumetric rendering techniques. While these methods can process complex pieces of cloth, the realism of renderings is limited, since all scattering by fibers is purely diffuse.*

While previous volumetric methods for rendering cloth already deliver a decent impression of cloth, they are not accurate enough for predictive rendering since the anisotropic nature of the micro-geometry is not modeled properly. This can be seen in Figure 2.4. To overcome this problem, Jakob et al. [JAM$^+$10] recently devised an elegant framework for volumetric modeling and rendering of materials with anisotropic micro-structure. Here, the local aggregate optical behavior of complex materials is modeled as a distribution of well-separated non-spherical particles approximating the phase function of scattering events on a per voxel level. The resulting volumetric representation is then rendered by employing a novel anisotropic diffusion method. This approach is very general and can describe many kinds of anisotropic structures well. It integrates perfectly into modern physics-based rendering systems.

To be of practical use in the context of predictive cloth rendering, suitable particle models that efficiently pre-integrate to phase functions are required. Unfortunately, deriving such models directly from measured optical properties of the yarn fibers, i.e. the underlying fiber scattering functions, is very challenging – in particular in the presence of multiple materials, e.g. different yarn colors or composite materials. Furthermore, while desirable from a theoretical point of view, strictly volumetric approaches, such as the one of Jakob et al. [JAM$^+$10], generally suffer from issues related to high frequency volumetric detail. Small scale structures in global illumination patterns or shadowing artifacts caused by discontinuities at interfaces between optically dense and sparse regions, require a sufficiently high volumetric resolution to model these effects accurately. Especially in woven cloth, yarns with different materials are often located next to each other and therefore high frequency details are common, leading to the necessity of very high spatial resolution of a purely volumetric approach in order to minimize bias.

In concurrent work, Zhao et al. [ZJMB11, ZJMB12] have presented a framework to capture and synthesize cloth based on Micro-CT scans. The aim of this work was to derive the parameters of a sophisticated geometric and optic model for cloth from real measurements. They first capture the geometrical information using Micro-CT scanners and they automatically extract the weave pattern of a cloth sample from volumetric data. Next, they estimate optical properties based on photographs. Finally, they use structure-aware volumetric synthesis to generate several different designs from a database of small measured samples. This framework has produced the most detailed micro-geometry of cloth so far. Unfortunately, Micro-CT scanners can only scan small samples and measurements are time-consuming and costly. Our approach, described in Chapter 8, instead uses one single image only.

Light Scattering from Fibers

Light scattering from fibers can be described similarly to light scattering from surfaces by using a bidirectional scattering function. Optical properties are mainly determined by *absorption*, *refractive index* (e.g. index of refraction for wool 1.576, silk 1.35, polyester 1.53) and *cross*

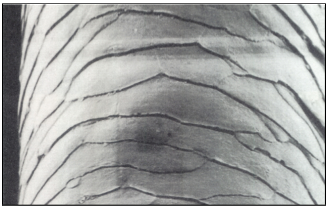


Figure 2.5: *Electron micrograph of a hair fiber. Taken from [Rob94]*

sectional shape (often close to circular e.g. for wool and many industrial fibers) with diameters of 10–100μm (e.g. on average 17-42μm for wool, 15μm for silk, 13μm for polyester) – values according to [MH62]. The micro-geometry of fibers, e.g. the scales of wool hairs, can also have a great effect on a rendered image, strongly affecting scattering. Figure 2.5 shows scales of a human hair fiber in a close-up view.

Marschner et al. [MJC$^+$03] introduce a curve scattering function that describes how light is reflected from such a human hair fiber. Zinke and Weber [ZW07a] describe a complete framework for light scattering from fibers, including the curve scattering function from Marschner et al. [MJC$^+$03] as a special case. They adapt the BSSRDF to better match the properties of fibers. Instead of using a parametrization directly on the micro-geometry of a fiber, they propose the Bidirectional Fiber Scattering Distribution Function (BFSDF), parametrized locally by an infinite enclosing cylinder. This makes the BFSDF independent of macroscopic effects such as curvature of fibers, allowing for several simplifications. First, for fibers with homogeneous optical properties, a dependence on the exact position along the fiber can be neglected – alternatively, a relative distance of the pair of points considered is used. A further simplification is given by the Bidirectional Curve Scattering Distribution Function (BCSDF) [ZW07a], for a distant observer and distant lights. Essentially, the contributions of light scattering are averaged along the width of the fiber. As even individual fibers can often hardly be distinguished for common viewing distances of cloth, the BCSDF makes a good compromise between compactness of representation and accuracy. However, it also has limitations, as we discuss in Chapter 13.

A three component BCSDF model relying on first principles of physics such as absorption and refractive index has been proposed by Marschner et al. [MJC$^+$03], where it has been shown that scattering can be approximated well for hair by accounting for the three strongest scattering modes:

1. R-component describing direct surface reflection.
2. TT-component describing light that gets transmitted through the fibers and is forward scattered.
3. TRT-component describing back-scattered light reflected inside the fiber.

For all of these components, light is scattered to a cone centered around the tangent direction. This effect causes the characteristic anisotropy of cloth, in case of typical yarns with locally parallel fibers. In this work, we use the model described by Zinke et al. 2007 [ZW07a], whenever we represent the optical properties of fibers.

Rendering of Fiber Assemblies: Besides explicit path tracing methods that require each hair strand to be modeled [ZSW04], also very efficient approximations regarding multiple scattering have been presented. All of these approximations rely on the fact that the multiple scattering distribution in hair tends to be smooth with only little high frequency detail [MM06, ZYWK08, MWM08]. Moon et al. [MWM08] present an interesting idea for modeling micro-geometry statistically, related to the cloth rendering approach we present in Chapter 11.

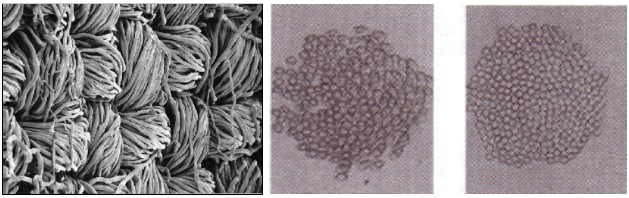


Figure 2.6: *From left to right: Close up of a typical woven cloth. (taken from [S.04]), cross-sectional view of ring spun stable fibers and cross-sectional view of filament yarn (taken from [SB06]).*

2.2 Elements of Cloth

In the following, we provide relevant background information on the elements of cloth. Cloth consists of three natural scales: fibers, yarns and yarn-compositions.

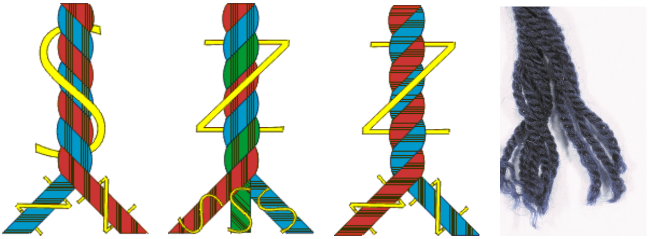
2.2.1 Fibers

The micro-structure of cloth is described by the geometric and optical properties of the small dielectric fibers that constitute yarns. Different types of fibers can be used, affecting both the appearance as well as physical properties such as resistance to wear or heat insulation. Wool, cotton, silk and synthetic fibers are common.

We have described how to represent optical properties of fibers in Section 2.1.2.

2.2.2 Yarns

On the one hand, *filament yarns* are produced by grouping or twisting a few hundred, long, continuous fibers. An example are fibers that are taken from cocoons made by the larvae of the silkworm. These can be



(a) Standard ply (b) Sewing thread (c) Voile yarn (d) Thick yarn

Figure 2.7: *Illustration of several yarns created by different combinations of S- and Z-twist (taken from [Kis12]). In the right-most image you can see a photograph of a yarn made thicker by adding one more layer of plies.*

hundreds of meters long.
Spun *staple yarn* on the other hand is created by twisting hundreds of thousands of short fibers together to form a cohesive thread (see Figure 2.6 for close-up and cross-sectional views). This enables the production of yarns of arbitrary length even though individual fibers may be significantly shorter. For example cotton only has a typical length of 2–3 cm. Smaller fibers typically result in more *uneven* yarns [Sha02]. Even stronger yarns are made up of a number of plies, where each ply is a single spun thread. Different variations exist, concerning the direction in which fibers are twisted to form a ply and the direction in which plies are twisted to form a yarn (Figure 2.7). These directions are called *S*- and *Z-twist*. One can also add even more levels of a twisting hierarchy (Figure 2.7(d)).

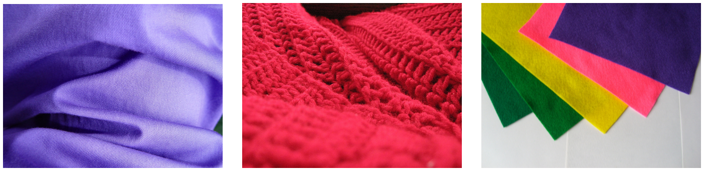
Usually small fibers protrude from a yarn – a property called *hairiness*. Different techniques exist to reduce this often unwanted effect. One of them is mercerization [Sha02]: Yarns created from cotton or hemp are subjected to chemical treatment to reduce hairiness and increase strength and luster. Apart from that, several different colorization procedures and brighteners can be applied to significantly alter the optical appearance of yarns. In the following we summarize the different processes involved during industrial spinning.

Industrial Spinning (according to [Sha02]): Two main types of spinning approaches exist. The simpler of the two is rotor (also called open-end) spinning: Raw fiber material is put into a rotating drum; when fibers are pulled out of the middle of this drum, they are twisted and form a yarn. Rotor spun yarns have different qualities when compared to ring spun yarns – they are ”weaker, more bulky, and more extensible” [Lor71]. An alternative and more commonly used approach is ring spinning.

This process in contrast is divided into several different stages. After cleaning the raw fiber material (commonly done using air pressure in the *blow room*), the first important process is *carding*. Here, carding cloth, with needles sticking out in a regular pattern, is attached to rotating cylinders and fiber bundles are pressed through the needles. The main effects of this step are the isolation of individual fibers from the bundle, giving fibers a common orientation and creating the *sliver* – a thick strand of fibers. When desired, different types of fibers can also be blended and mixed during carding. Common high-quality yarn is then combed. This is mainly done to remove more of the very short staple fibers (some of these were already removed during carding). Additionally, fibers become more parallel. Overall, combing increases the evenness and smoothness of yarns, greatly affecting the appearance.

The combed sliver is then processed by the *drawframe*. Here, two slivers are commonly blended into one to obtain a more homogeneous strand. In *roller drafting*, this thicker sliver then glides over different rollers one after another, each of them rotating faster then the previous one. Fibers are pulled out, influenced by their friction on rollers, the speed of rollers and friction with other fibers, creating a certain linear density within the sliver after elongation. Fibers are further drafted during *roving*. A small twist can be added and the resulting *rove* is wound up onto bobbins for transportation. The final amount of twist and fiber density is then defined during the actual ring spinning process. Again the *rove* is drafted running over rollers operating at different speeds. It is then led through a traveler, moving fast on a ring surrounding the spindle onto which the yarn is coiled upon. The speed of the traveler defines the amount of twist that is added.

2.2.3 Composition



(a) Woven cloth[1] (b) Knitwear[2] (c) Felt[3]

Figure 2.8: *Different types of cloth, characterized by different yarn and fiber arrangements.*

Many techniques have been developed to create pieces of cloth from yarns and fibers. Here we describe three of the most common ones: Woven cloth and two types of non-wovens called knitwear and felt.

Woven Cloth: To manufacture single layer woven fabrics (Figure 2.8(a)) one interlaces two orthogonal sets of parallel yarns, called *warp and weft yarns*. Each set of yarns can have different properties such as color and size. If yarns within one set have different colors, this is called *color effect*.

In the normal direction of a flat single layer woven cloth, no more than two yarns can be on top of each other, assuming a fabric that

[1] Woven-image: CC-BY 2.0 Scott Robinson
[2] Knit-image: CC-BY 2.0 Chris Phan
[3] Felt-image: CC-BY 2.0 Siona Karen

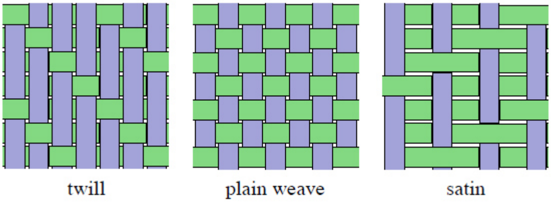


Figure 2.9: *Three common weave patterns are shown (illustration taken from [IM06]).*

does not fold onto itself. This allows for a simple representation of the basic structure: We consider all positions in the projection onto the main plane, where yarns cross. At each position, a binary value describes, which yarn can be seen from above. If we arrange these values in a matrix, we obtain the *weave pattern* matrix. While alternative representations are often used to program looms, these can be computed directly from this matrix. Depending on the setup, yarns can have varying spacings between each other, locally affected by the weave pattern.

Many fabrics consist of a small ever repeating pattern. A few famous named patterns exist such as *plain weave*, *satin weave*, and *twill weave* (Figure 2.9). However, in general, any binary matrix describes such a pattern as long as yarns cross often enough to hold the cloth together. Additionally, a color effect annotation is often used to create a more interesting appearance (see Figure 2.10 for an example). Formal descriptions for pattern and color effect can be used for catalogization and – together with some more information about yarns and the spacing between them – for manufacturing a piece of cloth. More complex pieces of cloth can be created by using several layers as is common for example for ties.

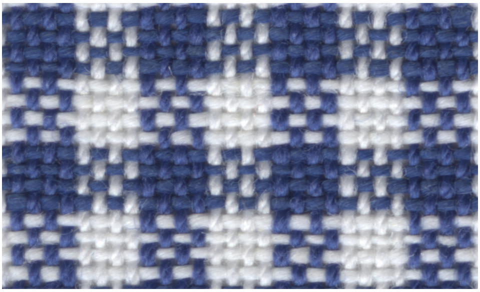


Figure 2.10: *Plain weave with color effect, captured using a flatbed scanner. This cloth has been created by applying a* 2×2 *plain weave matrix* $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ *and using for both warp and weft yarns the color effect description:* $[4 \times blue, 4 \times white]$. *This results in a repeating pattern of* 8×8 *yarns.* 2×3.5 *repetitions of this pattern can be seen in the image.*

As our aim is to automatically analyze woven cloth samples, we give a more detailed description of the weaving process as background information:

Industrial Weaving / Looming according to [Sha07]:
To be able to use the spun yarns as warp yarns in a loom, they are run through further processing steps. Yarns have to be prepared as packages, re-*winding* them from spinning frames, so that sheets with several hundreds of yarns can be wound onto a metal beam. These steps are called *beam preparation* and *warping*. Two types of yarns are found during the weaving process: *Warp* yarns are spun onto the loom and *weft* yarns are interlaced with them.
The final process before the actual weaving on a loom is called *sizing*. The main goals of this step are to strengthen the warp yarns and to reduce the friction during weaving. Warp yarns are lubricated, e.g. using wax, and are flattened using a squeezing roller, altering yarns' cross-section shapes. In contrast, weft yarns are only slightly waxed and wound onto cones.

For weaving, beams of warp yarns are attached to the loom and the yarn ends are drawn in so that every yarn follows a specific path through different parts of the machine.
During *shedding*, warp yarns are raised or lowered so that fill yarn can be pulled through them in specific configurations given by the weave pattern. Different looms support different shedding motions. The simplest one is the *Tappet* – with its 8 shafts, it can produce a maximum weave repeat (weave pattern matrix size in weft direction) equal to 8; only allowing for simple patterns. More complex patterns can be realized by *Dobby* looms, allowing a weave repeat size of 28. In *Jacquard* looms, every warp yarn can be raised or lowered individually, allowing the use of arbitrary weave pattern matrices. In this work, we concentrate on repeating woven cloth created by Tappet or Dobby looms. After a weft yarn has been inserted, the *reed* pushes the new yarn against the end (*fell*) of the already woven part in a process called *beating up*, reducing the spacing between yarns and increasing the compactness of cloth.

Knitwear: In knitwear (Figure 2.8(b)), yarns are arranged in 3-dimensional loops. Two basic stitch types are combined to create different patterns (Figure 2.11). Starting with an initial row of loops, the loops of the next row are created by pulling through the loops of the upper row. One distinguishes between two types of stitches: When the yarn is pulled through a loop from the previous row from below, this is called a *knit* stitch. If it is pulled from above, it is called a *purl* stitch. The three most common ways to combine these stitches are

- the *stockinette*, which only consists of *knit* stitches.
- the *garter* where the two stitch types alternate row-by-row. This is the most common type in hand-knitting – in practice one can always use *knit* stitches, but then one has to reverse the cloth sample after each row.

- the *1-1 rib* where the stitch types alternate within a row and the *2-2 rib* alternating between two *knit* stitches and two *purl* stitches within a row.

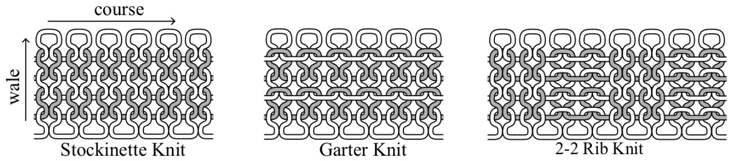


Figure 2.11: *Three standard knitting patterns are shown (illustration taken from [KJM08]).*

The direction along the length of a row is called the *course* direction. Orthogonal to that we have the stacking of rows in the *whale* direction.

Felt: This is one of the few types of textiles where fibers are directly combined instead of first creating yarns. As we have described in Section 2.1.2, wool fibers consist of many scales. If many wool fibers are moisturized and subjected to friction, these scales make the fibers stick together. This results in a material of relatively homogeneous density (Figure 2.8(c)). Other types of fibers can also be mixed into the material as long as there are enough fibers with scales that stick to each other.

Finally, textiles can be created by cutting and sewing of cloth samples of various types. A description of that process is beyond the scope of this work.

Part II

Image-Based Analysis of Woven Cloth

CHAPTER 3

INTRODUCTION

In this chapter, we give an overview of major image segmentation approaches, related to our automatic pipeline for analyzing cloth, and we discuss different capturing setups to be used for an automatic analysis of cloth. Finally, we give an overview of our analysis pipeline.

3.1 Background: Statistical Image Segmentation

The seminal work of Geman and Geman [GG84] introduced the principles for statistical segmentation of images. We want to segment an image $I \subset \Omega$ into a number of regions $\Omega_i \subset I$, separated by a boundary Γ. This is done by applying Bayes' rule:

$$p(\Gamma|I) = \frac{p(I|\Gamma)p(\Gamma)}{p(I)} \tag{3.1}$$

A segmentation is obtained by estimating a maximum a-posteriori (MAP) solution for Γ. The prior $p(\Gamma)$ restricts the curve to certain shapes such as curves with minimum length; more complex shape priors can also be used [CTWS02]. A segmentation can therefore be computed by minimizing:

$$E(\Gamma) = \sum_i \int_{\Omega_i} -\log p(I(x)|\Omega_i, x)\,\mathrm{d}x + \nu R(\Gamma) \tag{3.2}$$

where $R(\Gamma)$ is the regularization based on the chosen shape prior and ν controls the influence of that term. The value of $p(I(x)|\Omega_i, x)$ describes the probability that the intensity value $I(x)$ belongs to the $i-$th region Ω_i – a common choice for $p(I(x)|\Omega_i, x)$ is a Gaussian distribution:

$$p(s|\Omega_i, x) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp(-\frac{(s-\mu_i)^2}{2\sigma_i^2}) \tag{3.3}$$

The μ_i are the means and the σ_i describe the standard deviations of the intensity distributions of the regions.

As shown by Zhu and Yuille [ZY96], the statistical formulation using a Gaussian intensity distribution is equivalent to the Mumford Shah functional [MS88],[MS89] in the so called *cartoon-limit* (popularized by Chan and Vese [CV01]).

$$E(u_i, \Gamma) = \sum_i \int_{\Omega_i} \parallel I(x) - \mu_i \parallel^2 \; dx + \nu R(\Gamma) \tag{3.4}$$

In addition to the Gaussian model, several other probability densities for pixel intensities in regions have been proposed (i.e. Laplace distributions [HS05] and non-parametric densities [KFIY$^+$05]). *Local* region models [BRW05],[LT08] not only describe the intensity homogeneously for a full region but instead they obtain different distributions around each pixel inside local Gaussian windows. This statistical formulation has been related to the full Mumford Shah model by Brox et al. [BC07]. Another direction of research is to frame image segmentation as a Markov Random Field (MRF) model. Especially techniques based on graph cuts [BVZ01] have become popular. Rother et al. [RKB04] alternate between updating statistics of regions and a segmentation of the image based on these statistics via graph cuts. This approach is conceptually somewhat related to how we estimate the weave pattern.

3.2 Capturing Setups

We have tested different setups to capture images of cloth for the automatic analysis. In the following we describe some of their advantages and disadvantages. Cloth can be easily put onto a flatbed scanner. This allows us to capture high-resolution images for relatively large cloth samples. However, especially for slightly thicker cloth samples, the imaging system of the scanner, we have used, cannot perfectly focus on the fibers. This leads to blurry images, where fibers can hardly be resolved.
Another problem with this setup is caused by the anisotropic optical properties of cloth and by highly specular fibers. A flatbed scanner usually consists of a sensor bar located directly next to a bar of light. This leads to strong highlights and strong intensity variations in the image in areas where the yarn is tilted towards the light. These effects make the detection difficult as the intensity of a yarn can vary much.

Alternatively one can use macro photography. Here one has to find a compromise such that one can resolve individual fibers but still obtain photos with a sufficient number of repetitions of the weave pattern. We use a simple measurement setup with a single consumer camera and smooth, indirect lighting to avoid strong regular highlights. The camera is placed above the sample, whose weave pattern is roughly aligned with the image axes.

A great advantage of this setup, when compared to the flatbed scanner, is that we have much better control over the lighting environment. The indirect lighting produces images in which the color intensity of single yarns varies much less than for direct illumination. This allows us to more easily select a set of features that can be used for the analysis. To be able to analyze structural details in bright areas as well as in shadows and to be able to analyze both white and black yarns, we make use of high dynamic range (HDR) photography. We take a series of low dynamic range (LDR) images with varying exposure and combine these to a single HDR image [MP94].

3.3 Overview

Our goal for this part of the thesis is to automatically analyze cloth at the yarn level. More precisely, we estimate the yarns' centerlines, widths and colors, the weave pattern as well as deformation fields that will subsequently be applied to a procedural fiber-based cloth model.

To the best of our knowledge there exists no other current approach that fully reverse engineers woven cloth at this level. The analysis is challenging on many different levels:

1. Cloth can be easily deformed in non-rigid ways due to stretching and shearing. To address this issue we propose a bi-scale regularization framework to remove non-rigid cloth deformations; at the same time we estimate flow fields that give us the characteristics of yarn deformation and yarn width over several repetitions (Chapter 4). These flow fields will be applied during synthesis to model the detailed variations of yarns.

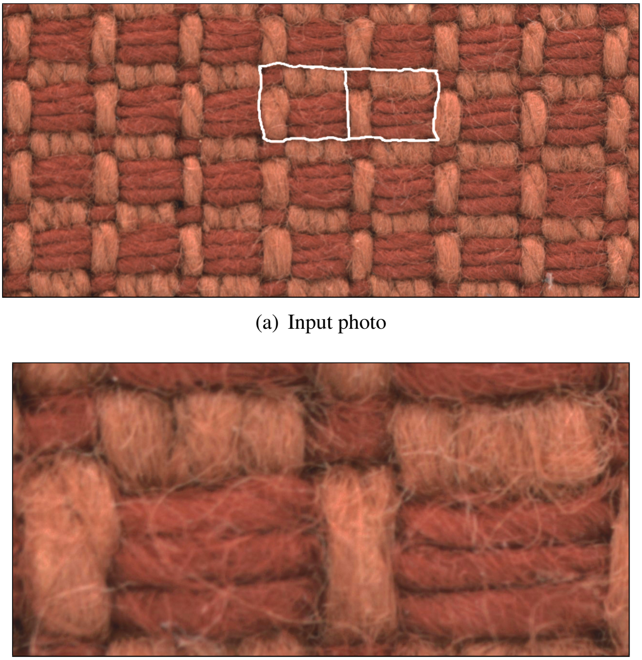
2. As every yarn consists of numerous fibers, the appearance is fuzzy (see Figure 3.1), and the fibers exhibit strong locally varying highlights, even with relatively smooth lighting. Yarns are often partly covered by protruding fibers of neighboring yarns, making them hard to distinguish from each other in an image (see Figure 3.1(b)).This problem is effectively solved by exploiting self similarities and by warping, removing the non-rigid deformation of the cloth sample. When averaging over the regularized repeating pattern, we obtain a clear image with salient yarn borders (Chapter 4).

3. Tracking of yarns in an image needs to account for occlusion by other yarns – this in turn defines the weave pattern for single layer cloth. Using a new graphical model, we are able to automatically extract an underlying weave pattern for complex woven structures (Chapter 5.3)

4. We present for the first time an automatic method to perform

image segmentation of woven cloth into individual yarns, extracting yarn parameters such as their paths and widths (Chapter 5.4).

The following chapters in this part of the thesis are based on the work we have presented in [SZK15].



(a) Input photo

(b) Closeup

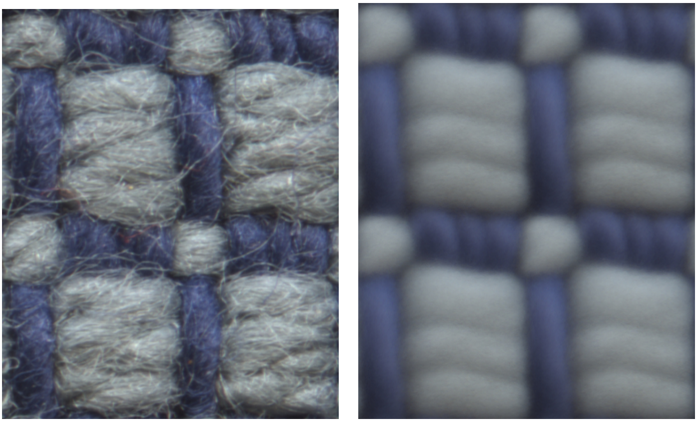
Figure 3.1: *Illustrating a typical input image and pattern repeat. (a) shows woven cloth with a repeating pattern. Due to shearing, internal non-rigid deformations and differences of e.g. yarn texture, instances have varying outlines and structure. Two instances of the repeating pattern are marked in white and are shown as a closeup in (b). Some borders, especially between orange yarns, are completely obscured. High frequency features complicate inference by means of edge detection. Therefore, an automatic analysis is challenging.*

CHAPTER 4

BI-SCALE REGULARIZATION OF WOVEN CLOTH

Due to assumptions of continuity and the Gestalt laws [Wer23], a human can often infer the structure of an object with occluded parts just from looking at it even if he has never seen such an object before. The analysis of woven cloth by an untrained observer is a typical example for this: it is usually clear to someone inspecting a cloth sample, where an occluded yarn becomes visible again. Even though humans can use subtle clues of shading and shadows to decide where discontinuities in depth occur, looking only at the details is often not sufficient: When boundaries between objects become blurred, a correct segmentation cannot be decided by only looking at local information. Instead, a more global view of the objects may be needed to make a correct decision. As an example see Figure 4.1 (a), where the individual yarns can hardly be distinguished locally, however the structure becomes clearer if the surrounding context is analyzed as well. An important clue can be the analysis of symmetries.

In the following, we focus on the common case of woven cloth with a small repeating pattern as created by tappet or dobby looms. When a photograph of a piece of cloth is taken, the repeating pattern manifests as a small (almost) rectangular part of the image. This is repeated several times following a translational symmetry. We present an automatic method, exploiting these structural similarities, to create a



(a) Input image (cutout)

(b) Computed regularized median

Figure 4.1: *On the left you can see a cutout of an input image: Some borders between yarns are completely obscured. Dark edges caused by shadows of a yarn's plys may be more salient than those between yarns. On the right you can see the median prototype image (cloned in a 2×2 grid) computed completely automatically from > 100 repetitions using our method – all individual yarns can be clearly distinguished from each other.*

regularized version of the fabric image. In practice, symmetries are imperfect because of global shearing and stretching of fabrics and local irregularities. Even the same yarn looks different at varying locations, because of the twisting of fibers and plies – these usually do not repeat with the same frequency as the weave pattern does. Fibers protruding from yarns, also called hairiness fibers, may cover neighboring yarns at varying locations.

The problem of cloth regularization is related to the analysis of near regular textures. Much work has gone into analyzing and characterizing these textures. This includes the task of finding an underlying deformed lattice of repeating structures. However, in addition to finding such a lattice, we are also interested in deformations at a finer scale. We not only want to compute an underlying deformed lattice, describing a basic stretching in the context of cloth, but we also want to estimate a fine deformation field, to faithfully capture curvy deformations of individual yarns. For woven cloth for example, warp and weft yarns move in two layers on top of each other. While they move similar for coarse scales showing the basic deformation of the cloth, yarns slide on top of each other in varying directions when looking at local details.

Our contributions in this chapter are:

- We observe that the characteristics of cloth require different types of regularization at different scales.
- We present a framework that can regularize the deformations of cloth at a coarse and a fine scale.
- We generate a clear, noise-free template of the repeating pattern in a process related to non-local filtering.
- We estimate deformation fields, representing coarse and fine characteristics.
- We describe applications that can utilize the computed information.

4.1 Related Work

Skew detection and corresponding image rectification is the basis of several techniques for the analysis of fabrics [WZP06, Jeo08, PGZ08, RWJ09, PG09]. Current approaches for skew rectification compute a global linear or affine transformation that is applied to the whole image. While this an effective coarse approximation to the problem at hand, it neglects the fact that fabrics tend to also have both a smooth global non-rigid deformation as well as local deformations of yarns.

Much work has been invested in finding techniques to handle repeating structures in images [LM96, SZ99]. Fabric images showing a repeating structure can be regarded as instances of near-regular textures containing translational symmetries. Early approaches have concentrated on estimating the generators of an underlying translational symmetry group [LC00], disallowing irregularities with respect to deformations. In the following, we concentrate on approaches that not only handle irregularities with respect to varying textures but can also handle warpings of the repeated elements. Liu et al. [LC01] also detect affine transformations for skewed symmetries. Later work also allowed non-rigidly deformed lattices by performing user-guided detection [LLH04]. Newer approaches compute a deformed lattice completely automatically [PBCL09].

In concurrent work, Hilsmann et al. [HSE11] have also presented an approach to regularize cloth by warping individual repetitions onto a mean sample. However, their approach only analyses cloth at a single coarse scale and does not estimate the fine variations of yarns.

4.2 Overview

Applying image segmentation algorithms on the input image directly is difficult, because of the high-frequency details of fibers. Cloth is easily deformed, complicating an automatic analysis. We want to exploit self-similarities in the image to aid in reverse engineering cloth in the presence of non-rigid deformations.

Our approach first regularizes non-rigidly deformed cloth. Locally, yarns may not be clearly visible because they can be occluded by fibers from neighboring yarns. However, when averaging over several warped repetitions, the structure at a yarn level becomes clearly visible. This can be seen in Fig. 4.1 (b). A segmentation of this image is much easier when compared to segmenting the input image. The estimated deformations will later be used to model the natural variations of yarns.

Our main observation is that we have to solve the regularization problem at different scales. We first find a shearing transformation, removing skew. Next, we estimate a flow field describing a smooth deformation – applying this flow field to the sheared input photograph, we obtain an image, where the individual repetitions are aligned as a grid. Additionally, we compute a fine flow field, capturing the variation of yarn deformation and yarn width – this information is especially important later-on during synthesis. While we have to strictly avoid discontinuities in the smooth / coarse flow field to also avoid discontinuities of yarns, we explicitly want these at a fine scale. Yarns move on top of other yarns in two layers in varying directions. We want to register individual yarns within each repeating pattern onto a corresponding yarn in a median image of the repeating pattern. This median image is not known in advance and is computed in a process alternating between estimating the flow field and the median. In the following we call this image *regularized median* $\mathcal{M}$.

Implementation: First, we find the size of the repeating pattern by the use of auto-correlation analysis [KKO99] and rectify the input image using a linear geometric transformation, obtaining image $\mathcal{I}$. From this we compute an initial estimate of the repeating pattern by computing the median of all repetitions. We tile the pattern in such a way that we obtain an image of the size of the linearly transformed image.
Second, we apply optical flow image registration [WBBP05, SRB10]

at a coarse scale to align the transformed image $\mathcal{I} \subset \Omega$ and the tiled image $\mathcal{M}$:

$$\begin{aligned} F(u) \quad &= \int_\Omega \parallel \mathcal{I}(x + u(x)) - \mathcal{M}(x) \parallel^p dx \\ &+\lambda \int_\Omega \sum_{d=1}^{N} \parallel \nabla u_d(x) \parallel^p dx \end{aligned} \tag{4.1}$$

where $u = u_1, u_2, ..., u_N$ describes the N-dimensional flow field and λ weights the regularization.

Coarse Regularization: Our goal is to register all weave pattern repetitions in the image $\mathcal{I}$ onto the regularized median image $\mathcal{M}$ to obtain a smooth flow field $u = u_c$ without discontinuities. We compute the coarse flow field by penalizing the $\mathbb{L}_2$ magnitude of image gradients ($p = 2$ in Equation 4.1 [HS81] – utilizing the implementation of [SRB10]). A large value is used for λ to enforce smoothness. The algorithm consists of the following alternating steps (two iterations of both steps were sufficient in practice):

1. Compute median image $\mathcal{M}$ for current image $\mathcal{I}$
2. Estimate optical flow field using F and compute warped image $\mathcal{I} \circ u$. Set it as new current image $\mathcal{I}$

Fine Regularization: Finally, we run the same algorithm again but this time we allow for discontinuities in the flow field ($p = 1$ in Equation 4.1). We employ TV-$\mathbb{L}_1$ optical flow [PUZ$^+$07], which is ideal for capturing fine scale discontinuities and obtaining a highly detailed flow field $u = u_f$ representing local variation such as yarn deformation.

4.3 Results

To test the performance of the algorithm, we have acquired images of woven textiles using a flatbed scanner. In Chapter 5, we will show further examples using a camera-based setup. The textile is loosely positioned in a way that yarns run parallel to the image axes.

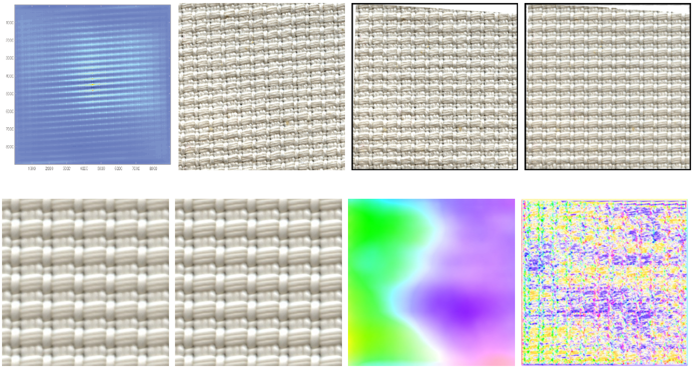


Figure 4.2: *Cloth with non-standard weave pattern and 4 warp and 4 weft yarns in the rapport captured using a flatbed scanner. Note the variation of intensity due to highlights – these are the reason, why we focus on a camera setup with diffuse lighting in the following chapters. From upper left to lower right: Auto-correlation image, Input image, Coarse rectification, Fine rectification, Coarse tiled regularized median (cutout), Fine tiled regularized median (cutout), Coarse flow field (not containing the previously applied linear transformation), Fine flow field. Flow fields are visualized using the common approach of describing the direction using the hue value and the flow intensity by the color saturation.*

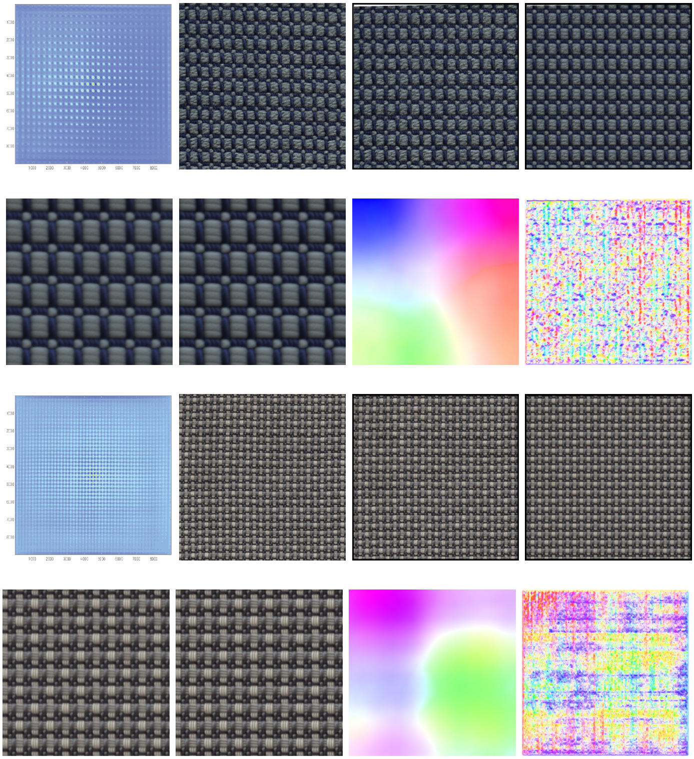


Figure 4.3: *More examples along the lines of figure 4.2. Upper example: A piece of cloth with the same weave pattern as in figure 4.2 with differently colored warp and weft yarns. Lower example: a piece of cloth with a complex weave pattern with color effect consisting of 10 warp and 10 weft yarns in the rapport.*

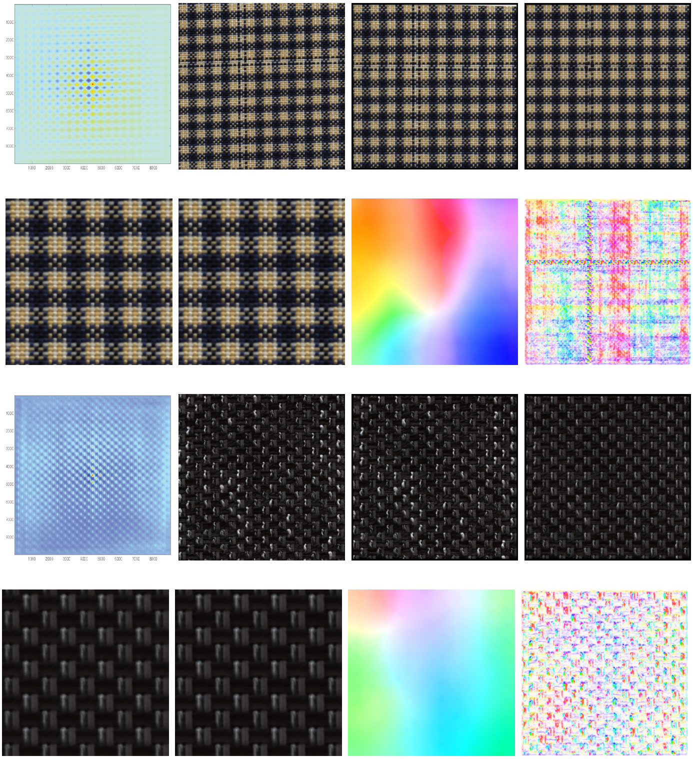


Figure 4.4: *The upper example shows a failure case, violating the assumptions. It does not purely consist of a repeating pattern. The lower example is a corner case, consisting of special yarns that vary in color. The approach still handles these cases reasonably well.*

As a result of our method, we obtain the final regularized median image and a set of repetitions registered onto it. Because fibers are arranged in a slightly different way in every repetition of the pattern, only the major shading of the yarn is left after median filtering and all of the high frequency fiber structure is removed. Yarns look like bent shaded cylinders in this regularized median image. Even if yarn borders are not clearly visible in some of the individual repetitions, they clearly emerge in the median of the aligned repetitions (Figure 4.1 demonstrates this in a closeup view). Figures 4.2 and 4.3 show major steps of the registration pipeline for several examples. Figure 4.4 presents a failure case, which violates the assumption of a repeating pattern.

4.4 Conclusions

We have presented a method to regularize the repeating structure of woven cloth. We obtain a clear image with salient yarn borders that can be used for further processing such as image segmentation as described in the following chapter. While yarn borders already emerge after the coarse regularization step, they show even more contrast after fine regularization. Additionally, we obtain a set of repetitions, registered against a common median image. This means we could analyze corresponding pixel neighborhoods in each repetition and therefore, features such as texture descriptors could be computed more robustly over several repetitions avoiding problems of local defects and blurring in a single repetition alone. While yarns do not look equal in every repetition, their statistical properties should match.

We further estimate two flow fields that describe different properties of the sample. A coarse field describes a smooth non-rigid deformation. A fine flow field describes the variation of yarns. We will apply these fields in Chapter 8 to a synthetic cloth model.

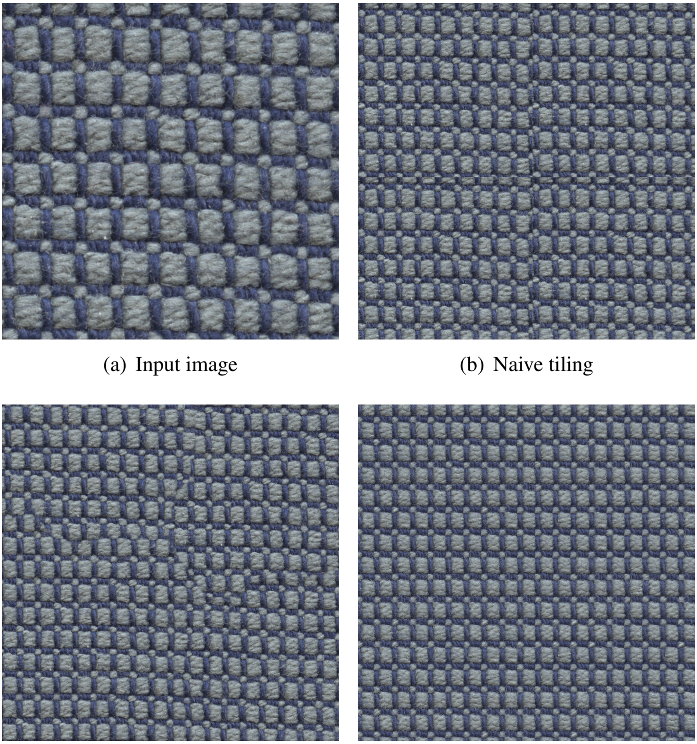
The filtering we obtain after bi-scale regularization could be regarded as a structure-aware variant of non-local means (respectively median) filtering [BCM05]. In standard non-local means filtering, neighborhoods are constructed based on the similarity of color distributions surrounding pixels. In our approach, patches are first warped in a way that minimizes the sum of squared intensity distances between them. This aligns them in such a way that we obtain a near perfect translational symmetry of the features. Then, neighborhoods are constructed according to corresponding elements in these symmetries. Although this greatly reduces the amount of neighbors considered, when compared to standard non-local means, the quality of the neighborhoods are greatly increased, leading to a high-quality filtering.

4.5 Applications and Future Work

Our motivation to implement this method was driven by our goal to completely analyze the structure of a piece of woven fabric as will be discussed in the following chapters. However, there are also more immediate applications for the approach. A common problem in graphics is to create tileable textures. In case of woven cloth, discontinuities of yarns can easily be spotted when the regular structure is interrupted. After having regularized the piece of cloth, we can compute an arbitrarily large tiled median image and then use constrained texture synthesis to add the fine detail on top of it (see Figure 4.5). To further improve synthesis results, one could also apply the estimated deformation fields to model irregularities of yarns – we will do this in Chapter 8, when we explicitly synthesize micro-geometry. Figure 4.6 shows how the decoupling of structures related to yarns and structures related to fibers can be used for image-based editing by increasing or reducing the perceived hairiness and roughness of a sample. Although one could also use other filtering approaches instead, we believe that the domain specific decoupling of structures increases the intuitiveness of the editing process. Another important application related to our approach for cloth regularization is that of defect detection for fabric images (see [Kum08, NPY11] for recent surveys). Here, the basic idea is to estimate the regularity of a sample to detect variations from it. Current techniques do not explicitly take the non-rigid deformations of cloth into account in the way we do. Using our approach, one can estimate repeating elements with the largest distance — for example, simply using sums of squared distances of intensities — to the regularized median. These are candidates for defective parts or dirt on the sample.

Figure 4.4 presents an example where several small patterns are arranged to form the cloth. Instead of creating the median image by tiling the same pattern for the whole image, one could create several different representative prototype patterns and locally select the best one using a clustering method.
Looking at a larger cutout of the cloth in Figure 4.4 would reveal that it actually does contain a perfectly repeating pattern at a different scale. A hierarchical approach could be used to detect all of these patterns.

In the future, one could explore the use of the method for other types of near-regular structures – a related application would for example be the analysis of knitwear.



(a) Input image

(b) Naive tiling

(c) Naive synthesis

(d) Constrained texture synthesis

Figure 4.5: *(b) to (d) show different methods to generate a larger texture from the input image in (a). Even the small bending deformation in (a), makes it difficult to find a solution without discontinuities when naively applying texture synthesis without information about structures and features. The information about the repeating pattern allows us to start with a perfectly repeating image and then only synthesize the high frequency detail by applying constrained texture synthesis (as can be seen in (d)) to avoid errors. A further improvement could be obtained by utilizing the flow field information.*

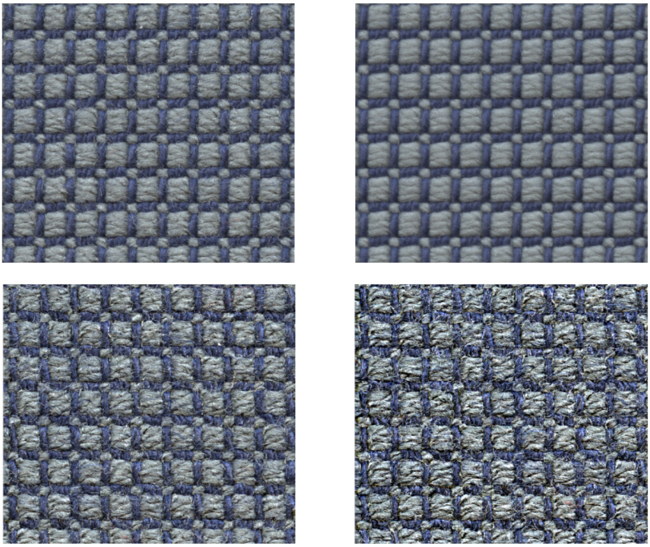


Figure 4.6: *The decoupling of basic yarn shading and structure captured in the median image and the high frequency details of fibers in the residuum, allows us to perform image-based editing. The upper left image shows the original photograph, and the other images show the effect of adding the high-frequency residuum to the median image with different intensities, creating different effects of roughness.*

CHAPTER 5

IMAGE-BASED ANALYSIS OF WOVEN CLOTH AT THE YARN LEVEL

The visually complex characteristics of woven cloth make the automatic detection of weave patterns a challenging task. In this chapter we use the regularization method described in the previous chapter and present the first method that completely segments the regularized median image into individual yarns and background. This gives us the paths of the yarns' centerlines, the average width of yarns and a labeling of regions showing the background. We further analyze yarn occlusion, from which we obtain the binary weave pattern matrix. Combined with information about the types of yarns that are used, this gives us a reverse engineering of woven cloth, which could be used to recreate the sample on a loom as long as we know the types of yarns that have been used.

5.1 Related Work

Zhang et al. [ZXW13] give an overview of the extensive research the textile research community has carried out within the last 30 years in the context of analyzing woven cloth. None of these works obtain a full segmentation of an image into individual yarns and background.

The related work for weave pattern detection splits into two main directions. The first one is to identify one of a few common weave patterns in an image. The alternative is to directly estimate an arbitrary weave pattern matrix.

5.1.1 Classification into Common Basic Weave Patterns

Several papers concentrate on classifying fabrics into one of a few basic weave types (i.e. plain, twill and satin weave): Huang et al. [HLY00] locate areas where warp and weft yarns cross by analyzing interstices of yarns. By summing gray-level values over all pixels of a row they obtain an accumulated value for each column (and vice versa for columns and rows). Minima of this function are assumed to be taken at yarn borders because of shading effects. A grid is constructed by placing lines in rows and columns where minima of these functions occur. A decision tree is built to recognize warp and weft floats. Jeon et al. [JBS03] train a neural network model on an estimated ratio of horizontal and vertical distances of minimal gray values in the input image. Kuo et al. [KT06] calculate texture characteristics by computing the gray-level co-occurrence matrix and use learning vector quantization networks to classify the fabric texture into three basic weave types, single or double knitted fabrics or non-woven fabric. Ben et al. [BN09] use a support vector machine classifier on different texture features obtained using Gabor wavelets, local binary pattern operators or co-occurrence matrices.

Cluster validity analysis can be used after fuzzy c-means (FCM) clustering of yarns in HSL color space ([PGL09]) or CIE-L*A*B* space([PGLW10a]) to obtain the number of yarn colors. Pan et al. [PGL09] introduce a period extraction method based on an entity they call 'Su index' for detection of the color effect. In [PGL^{+}10] they enhance an image by blurring it in the direction of the skew angle using steerable filters, before applying a variant of the gray-projection method, assuming that yarns may be darker at borders. In [PGLW10b], Pan et al. correct the output of a faulty FCM algorithm by comparing to the most similar weave pattern in a database.

In our approach, we can handle any weave pattern matrix.

5.1.2 Estimation of Binary Weave Matrix

Kang et al. [KKO99] use a special setup with light coming from behind the surface to augment positions between yarns where the light shines through. Bright pixels are connected to form a grid. Crossed states are classified by analyzing the aspect ratio obtained from minimum distances of local minima of vertical and horizontal gray-level profiles at a crossing point. They quantize colors at crossing points to obtain the colors for yarns. Kuo et al. [KSL04] create a grid by using a similar method as in [HLY00]. Fuzzy c-means clustering of color statistics in grid cells is used to distinguish between warp and weft yarns. Wu et al. [WZPY05] analyze double layer woven fabrics. They also construct the grid similar to [HLY00] – instead of summing gray-level values directly, they sum gradient values. This is done for obverse and reverse surface images of the sample manually registered against each other to analyze both layers. In contrast to our approach, current work based on clustering image features to obtain regions belonging to warp and weft yarns cannot handle any pieces of cloth with color effect.

Xin and Hu [XH09] propose an *Active Grid Model* in this context. The grid is locally adapted to follow yarn borders based on local gradient information. A neural network is finally trained to classify yarns based on local boundary information. Unfortunately, research on active contour models has shown [CV01] that relying on local image intensity gradients only is commonly not robust. Further, a grid is not a good model, if we want to fully segment an image into individual yarns and background. To find initial locations of yarn borders, Xin and Hu compare the Hausdorff distance of pixel locations as obtained from a binary edge detector with the location of a binary template showing two orthogonal lines as yarn borders. We refine the idea of template matching for estimating initial locations, utilizing more of the available image information within our SANCC filter. Wang et al. [WGP11] use a feature descriptor based on texture for clustering, obtained using principal component analysis.

In comparison to our approach, none of these approaches includes neighorhood information directly. Further, the full cloth segmentation we obtain using our active yarn model, gives us a lot more geometric information about a piece of cloth than is contained in the weave pattern alone.

Apart from techniques that work on 2-dimensional images, Shinohara et al. trace yarns in 3-dimensional computer tomography data ([STOK04],[STOK06] [Shi07], [STOK09]). Zhao et al. [ZJMB12] estimate the weave pattern from Micro-CT data of cloth.

5.2 Notation

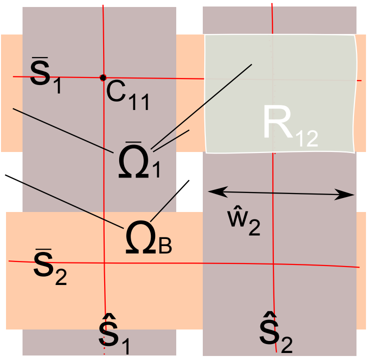


Figure 5.1: *Illustrating symbols used in this chapter.*

Having extracted the regularized median image M in region $\Omega \subset \mathbb{R}^2$

in the previous chapter, showing a *single* repetition, we now want to compute a maximum a-posteriori (MAP) estimate for yarn widths, centerlines and weave pattern. Please see Figure 5.1, for an illustration of the notation. We assume that Ω lies on a torus (coordinates wrap around image boundaries). We mark properties belonging to horizontally running yarns with a bar (as in $\bar{Y}_i$) and vertically running yarns with a hat ($\hat{Y}_j$). If we leave out the mark and use k as an index, (as in Y_k) we mean either type of yarn. We define the positions where the yarns' centerlines cross in the image as points $c_{ij} \in \mathbb{R}^2$. We define each yarn Y_k as a tuple (S_k, w_k) where S_k is a spline with control points in $\{c_{ij}\}$ and w_k is the average yarn width. Additionally, we want to estimate the weave pattern matrix $L = (L_{i,j})$ describing pair-wise, which horizontal yarn occludes which vertical yarn: $L_{i,j} = 1$ if the horizontally or $L_{i,j} = 0$ if the vertically running yarn is visible at crossing point c_{ij}. We define Ω_k as the union of all image regions showing yarn Y_k. These regions are commonly disconnected due to occlusion by other yarns. Finally, we define regions around each c_{ij} as: $R_{ij} = \bar{\Omega}_i \cap \hat{\Omega}_j$, where $\bar{\Phi}_i$ and $\hat{\Phi}_j$ are the extension of regions $\bar{\Omega}_i$ and $\hat{\Omega}_j$ to those image regions, where the corresponding yarns are occluded by other yarns. We define the background image region as $\Omega_B := \Omega \setminus (\bigcup_i \Omega_k)$. We now want to segment the image into yarns and background to find the projected geometry of yarns.

5.3 Weave Pattern Estimation

In this section, we propose a novel method to detect a weave pattern by using a Markov Random Field to not only cluster yarns using features such as colors but also taking local yarn neighborhoods into account. An asymmetric pair-wise prior is introduced that respects the structure of single layer woven cloth. To obtain the number of yarns, find initial locations for the later optimization procedure and as a feature to distinguish horizontally and vertically running yarns, we use the output of a specialized normalized cross correlation.

Symmetry-Aware Normalized Cross Correlation (SANCC): With nearly diffuse lighting, yarns are brightest along centerlines. The intensity gradually falls off going outwards towards each yarn's borders, showing a local mirror symmetry across the width. While this assumption may not be strictly valid for an individual repetition, it can clearly be observed in the regularized median image.
We use cylindrical templates that are either tuned to detect elongated horizontal or vertical structures. During normalized cross correlation, we emphasize the response on symmetric structures by only considering the minimum response of the two symmetric slopes of the template.

Let X_y be the result of filtering with the template detecting vertical structures. Specifically, we define SANCC operator X_y^m based on normalized cross correlation, emphasizing mirror-symmetries along y-axis:

$$\begin{aligned} X_y^m(u,v) &= \max(0, \bar{X}_y^m) \qquad (5.1) \\ \bar{X}_y^m &= \frac{1}{n\sigma_G\sigma_t} \sum_{y=-m}^{m} \big(h(0,y) + \sum_{x=1}^{m} 2\min(h(x,y), h(-x,y)) \big) \\ \text{with } h(x,y) &= (G(u+x, v+y) - \overline{G_{u,v}})(t(x,y) - \bar{t}) \end{aligned}$$

where G is a gray scale intensity image, computed from regularized median image M. Template t has size $(2m+1) \times (2m+1)$ and we define that the middle of the template has coordinate $(0,0)$. $\bar{t}$ is the mean of the template and $\overline{G_{u,v}}$ is the mean of G, restricted to a window of the template size around location (u,v); σ_t and σ_G describe the standard deviation of template and image under the current window. We sum over x,y under this window, utilizing at boundaries that we observe a repeating structure. As we have a template that shows a mirror symmetry along the y-axis, we can penalize non-symmetric locations in the image by taking the minimum of values at x and $-x$. We further clamp the value as we are only interested in positive correlations. One can define the corresponding operator $X_x(u,v)$, which mirrors along the x-axis, analogously. The normalization makes the filter relatively independent of the yarns' intensity / colors. We assume that cloth samples are captured in a way to have a known yarn width and use a fixed value for m. However, the output of SANCC is not too sensitive with respect to filter width. To find the initial locations of yarns, we sum X_y along the x-Axis: $l_y(x) = \sum_y (1 - X_y(x, y))$. Local minima of $l_y(x)$ give us the initial locations of vertically running yarns (analogously for $X_x, l_x(y)$).

Graphical Model: Based on the previous step we now estimate the binary labeling for the weave pattern matrix L. A challenge is that we have to locally identify yarns as warp or weft yarns, even though both may show the same color, as for example in Figure 5.2 and especially in Figure 5.5. The SANCC described in the previous section not only gives us an initial estimate for the locations of yarns but also hints about the direction (vertical or horizontal) a yarn is running along. However, this information is not reliable, particularly at yarn crossings. For that reason, we use a graphical model to combine different domain specific features that we solve using *max-product* belief propagation. The graphical model $G = (\mathcal{V}, \mathcal{E})$ consists of nodes $\mathcal{V} = (1, 1), ..., (m, n)$ associated with crossing points c_{ij} and regions R_{ij}. Edges $\mathcal{E}$ are given by the edges within the grid of crossing points such that each node has a four-connected neighborhood. The corresponding energy function E_L consists of a pairwise term W on neighboring regions and a data term D that compares feature distributions of regions (colors) with those of yarns: $E_L(L) = D(L) + W(L)$.
As we do not know the properties of yarns in advance, we estimate them iteratively in an expectation maximization framework. After each step of calculating belief propagation, we use max-product marginals to re-estimate properties, specifically colors, of yarns by taking a weighted average of corresponding regions.

5.3.1 Implementation

Pairwise Boundary Term:
Let us consider two horizontally neighboring regions: They likely have the same labeling if first, there is no border between them and second, they both are labeled to show a horizontally running yarn. For all other labellings (both are labeled to run vertically or they are labeled differently) they are likely separated by a visible image boundary. A related argument holds for two vertically neighboring regions. We penalize boundaries using:

$$W(L) = \sum_{i,j} \sum_{i^*,j^*} \exp\big(A(L_{ij}, L_{i^*j^*})\, \mathrm{g}(c_{ij}, c_{i^*j^*})\big) \tag{5.2}$$

with $(i^*, j^*) \in \{(i-1, j), (i, j-1)\}$. The asymmetric prior $A(L_{ij}, L_{i^*j^*})$ describes for what labellings we want to penalize boundaries in which direction:

$$A(L_{ij}, L_{i^*j^*}) = \begin{cases} 1 & \text{if}(L_{ij} = L_{i^*j^*} = 1) \wedge (j^* = j-1) \\ 1 & \text{if}(L_{ij} = L_{i^*j^*} = 0) \wedge (i^* = i-1) \\ -1 & \text{otherwise} \end{cases} \tag{5.3}$$

Boundary detector: We estimate the existence of a boundary between regions based on the color distance of their mean color and based on SANCC. For two neighboring regions $R = R_{ij}$ and above of it $R_{up} = R_{i-1,j}$, we define

$$\begin{aligned} g_{up}(i,j) = \lambda_c \parallel \overline{M(R)} - \overline{M(R_{up})} \parallel + \lambda_y |\overline{X_y(R)} * \overline{X_y(R_{up})}| \\ + \lambda_x |(1 - \overline{X_x(R)}) * (1 - \overline{X_x(R_{up})})| \\ + \lambda_{xy} |\overline{X_y(R)} * (1 - \overline{X_x(R_{up})})| \end{aligned} \tag{5.4}$$

where e.g. $\overline{M(R)}$ describes the average of M under region R. g_{left} is defined in the same way, switching roles of X_x and X_y. We set $g = g_{up}$ for vertically and $g = g_{left}$ for horizontally neighboring regions.

Data Term:
As we do not know yarn colors in advance we estimate them iteratively using a process based on deterministic annealing. We alternate between the estimation of the unobserved labeling L and the computation of the unknown colors of yarns based on the current labeling.

We compute yarn features based on weights:

$$w_{ij} = \frac{w_{ij}^{(1)}}{w_{ij}^{(1)} + w_{ij}^{(2)}} \tag{5.5}$$

with

$$w_{ij}^{(1)} = e^{\frac{-P_{ij}}{T}}$$

$$w_{ij}^{(2)} = e^{\frac{-(1-P_{ij})}{T}}$$

and temperature T, given P_{ij} describing the probability that region R_{ij} shows a horizontally running yarn as obtained from the max-product marginals (this means $1-P_{ij}$ is the probability that a vertically running yarn is visible). We simply initialize w_{ij} to 0.5 for the first iteration. In every iteration we recompute the colors of yarns as $\bar{C}_i = \sum_j \bar{I}(R_{ij})w_{ij}, \hat{C}_j = \sum_i \bar{I}(R_{ij})(1 - w_{ij})$ and then reduce T with $T = \alpha T$ and $\alpha = 0.9$ for our results. We define the data term:

$$D(L) = \sum_{i,j} P_{ij} d_f(R_{ij}, \bar{Y}_i) + (1 - P_{ij}) d_f(R_{ij}, \hat{Y}_j) \tag{5.6}$$

where $d_f(\cdot, \cdot)$ describes the similarity of region R_{ij} with yarn $\bar{Y}_i$, respectively $\hat{Y}_j$. In practice we simply use the $\mathbb{L}^2$ distance of mean colors.

The results of this step are a binary matrix, and for each yarn we obtain a color C_k.

5.4 Cloth Segmentation Using an Active Yarn Model

We present for the first time a method to automatically fully segment an image of woven cloth, obtaining the paths of yarns and their widths, given weave pattern L and given color features C_k of yarns, as obtained from regions R_{ij} in the previous section. We further obtain image regions where the background is visible, giving us the spacing between yarns – an important clue for the transparency of a piece of cloth. We represent each yarn as a tube with width w_k around its center-path spline S_k. The spline $S_k : [0, 1] \to \Omega \subset \mathbb{R}^2$ has boundary condition: $S_k(0) = S_k(1)$; please note that Ω is assumed to lie on a torus. For each tube we subtract all tubes, for which their corresponding yarns occlude the current yarn given the weave pattern. We use the rough location estimate obtained by SANCC to initialize all S_k. In the following, $\mathcal{E}_M$ gives the inverted output of an edge detector (e.g. Sobel) for M and $\Gamma(\Omega_k)$ describes the boundary of Ω_k. We segment the image by minimizing:

$$E(\{c_{ij}\}, \{w_k\}|I, L, C_k) = \lambda_{\mathcal{E}_M} \sum_k \sum_{x \in \Gamma(\Omega_k)} \mathcal{E}_M(x) \tag{5.7}$$

$$+ \lambda_B \frac{1}{|\Omega|} \sum_{x \in \Omega_B} \parallel C_B - I(x) \parallel \tag{5.8}$$

$$+ \lambda_R \sum_{i,j} \frac{1}{|R_{ij}|} \sum_{x \in R_{ij}} (L_{ij} \parallel \bar{C}_i - I(x) \parallel + (1 - L_{ij}) \parallel \hat{C}_j - I(x) \parallel) \tag{5.9}$$

$$+ \lambda_Y \sum_k \frac{1}{|\Omega_k \setminus \bigcup R_{ij}|} \sum_{x \in \Omega_k \setminus \bigcup R_{ij}} \parallel C_k - I(x) \parallel \tag{5.10}$$

$$+ \lambda_X \big(\sum_i \sum_{p \in \text{tb}(\bar{S}_i)} (1 - X_x(p)) + \sum_j \sum_{p \in \text{tb}(\hat{S}_j)} (1 - X_y(p)) \big) \tag{5.11}$$

$$+ R_{Sn} + R_{Sh} \tag{5.12}$$

The boundary of the tube should be located at yarn borders, as found by an edge detector (5.7). The background region Ω_B should be small and similar to a predefined color C_B depending on the capturing setup – black in case of the photos we have taken (5.8). Regions R_{ij} should have a mean color corresponding to color C_k of the yarn they are associated with given weave pattern L (5.9). The remaining visible part of the tube for yarn Y_k, excluding regions R_{ij} directly around a crossing point, should also have a mean color similar to C_k (5.10). This part consists of all yarn regions where the yarn is not above any other yarn but only above the background. With $p \in \text{tb}(S_k)$, we mean all pixels under a small tube around S_k. These tubes are 3 pixels wide in our implementation, and they should show a low value (high correlation) for the corresponding filtered images X_x or X_y (5.11). Values for the different λ_* were found through experimentation. We use the classic regularization term for snakes [KWT88] for all curves as R_{sn} and we regularize shearing of cloth similar to spring-based cloth simulation using R_{sh} (5.12).
We further avoid overlapping of yarns using linear inequality constraints, comparing the distance of control points for neighboring yarns with the width of corresponding yarns.

In the following, we describe the terms used to regularize the active yarn model in more detail:

$$R_{Sn}(\mathcal{S}) = \sum_k \left(\int_0^1 \parallel S_k^{'}(s) \parallel^2 \, ds + \kappa \int_0^1 \parallel S_k^{''}(s) \parallel^2 \, ds \right) \tag{5.13}$$

κ controls the influence of the *rigidity* and, as is commonly done [CKS97], we have set $\kappa = 0$ for all the results presented in this work. We also want to regularize shearing of cloth. Therefore, to keep diagonals short in the grid of crossing points we add:

$$R_{Sh}(c_{i,j}) = \sum_{i,j} \parallel c_{i+1,j+1} - c_{i,j} \parallel^2 + \parallel c_{i+1,j} - c_{i,j+1} \parallel^2 \tag{5.14}$$

for $i+1$ in the residue class modulo n: $i+1 \in \mathbb{Z}_n$ (similarly $j+1 \in \mathbb{Z}_m$).

We perform constraint optimization using linear inequality constraints. Yarn width should be greater than zero: $\forall k = 1...m{+}n\ : w_k > 0$. We avoid overlapping of yarns, constraining the distance of control points for neighboring yarns under the width of corresponding yarns:

$$\begin{aligned}
&\forall i = 2...n{+}1\ : c^y_{i,j} - c^y_{i-1,j} > \tfrac{1}{2}(\bar{w}_i + \bar{w}_{i-1}) \\
&\text{with } c^y_{n+1,j} := c^y_{1,j} + p && (5.15) \\
&\forall j = 2...m{+}1\ : c^x_{i,j} - c^x_{i,j-1} > \tfrac{1}{2}(\hat{w}_j + \hat{w}_{j-1}) \\
&\text{with } c^x_{i,m+1} := c^x_{i,1} + q && (5.16)
\end{aligned}$$

within an image M of size $p \times q$ where $c_{i,j} = (c^x_{i,j}, c^y_{i,j})$.

As a numerical solver we use an interior point method (Matlab's [MAT11] function fmincon).

5.5 Results

The effectiveness of our pipeline is demonstrated on four different examples showing different weave patterns. These have been selected to represent a large variety of different types of single layer woven cloth: A 2×2 plain weave pattern in Figure 5.4), a non-standard 4×4 weave pattern (Figure 5.3), an 8×8 plain weave with color effect where warp and weft yarns with the same color cross (Figure 5.5) and a complex 10×10 weave pattern with several different colors (Figure 5.2). The examples show different amounts of initial deformation.

For all of these examples, we are able to correctly estimate the weave patterns and perform an image segmentation into individual yarns fully automatically using a single set of parameters.

5.6 Conclusions

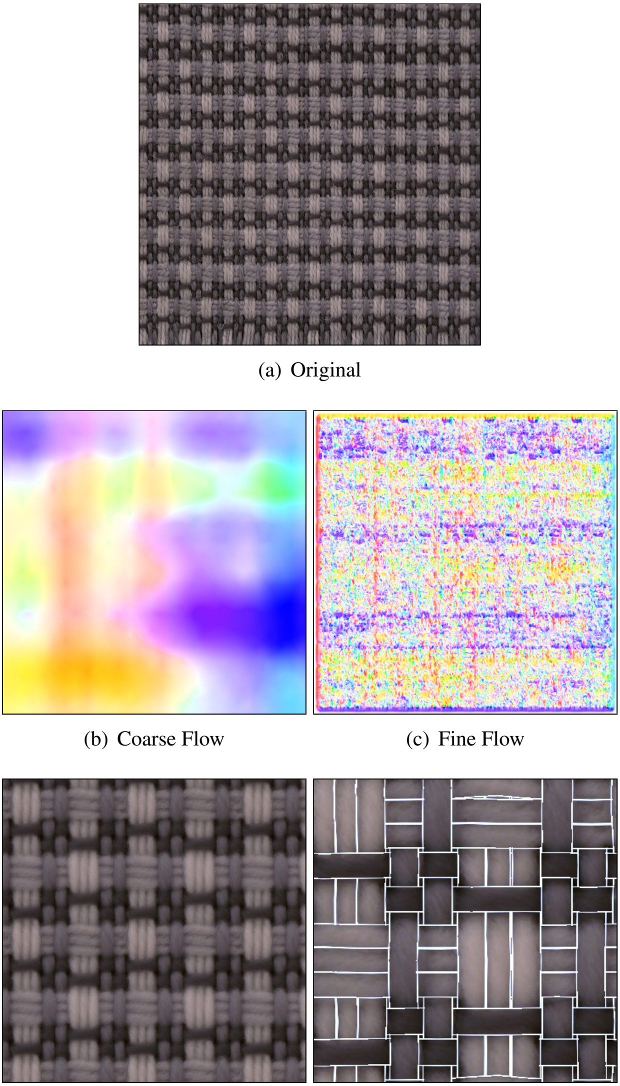
We have demonstrated the successful recognition of the underlying pattern of several samples of woven cloth, together with a full segmentation of the image into individual yarns – all using a single set of parameters.

Although our segmentation method works well in practice for the examples we have tested, there is the danger of running into local minima. Other strategies instead of the combination of belief propagation with an interior point method could be explored. For small patterns even the exact junction tree algorithm [LS88],[JLO90] may be an option instead of using loopy belief propagation. For the active yarn model it would be interesting to evaluate if one could use techniques such as convex relaxation or graduated non-convexity.

So far we only estimate the mean color of regions in the data-term within the active yarn model. Several improvements could be made to better identify regions with yarns. Different probability densities for pixel intensities in regions have been proposed (i.e. Laplace distributions [HS05] and non-parametric densities [KFIY$^+$05]). *Local* region models [BRW05],[LT08] obtain different parameters for Gaussian distributions around each pixel inside local Gaussian windows. This statistical formulation has been related to the full Mumford Shah model by Brox et al. [BC07]. It would fit well to the smoothly varying gradients in our median prototype image.

We could also extract texture information from the aligned repetitions and use this to further guide the labeling, especially during weave pattern estimation. The mean orientation of fibers can give a further indication of yarn directionality. However, depending on the capturing setup and the fineness of yarns, individual fibers, producing the texture, may not be distinguishable in the input image. In case of flatbed scanners one can hardly avoid this. In case of macro-photography we have more control over the capturing setup; however being able to see more details commonly comes at the cost of being able to capture less repetitions; unless stitching of several images is performed. Sharpness can vary in different regions due to focus effects. Even if fibers are visible, their orientations may vary strongly due to *hairiness fibers*. To avoid such problems, one could average texture features over all repetitions aligned with the warped median image. The fine flow field may have to be adapted in a way that it does not suppress the details at the fiber level. Giving a higher weighting to the regularization in TV-$\mathbb{L}_1$ optical flow may be sufficient.

Instead of using a capturing setup with indirect lighting only, one could capture the sample under varying lighting orientations. While we have mentioned that the anisotropic shading effects can be a problem in case of using a flatbed scanner, we could also use them to our advantage. Shading variation could be used as an indicator for yarn orientation and twisting angle. Stereo photography, photometric stereo or structured light could further be used to also estimate depth and the 3D-structure of cloth.



(a) Original

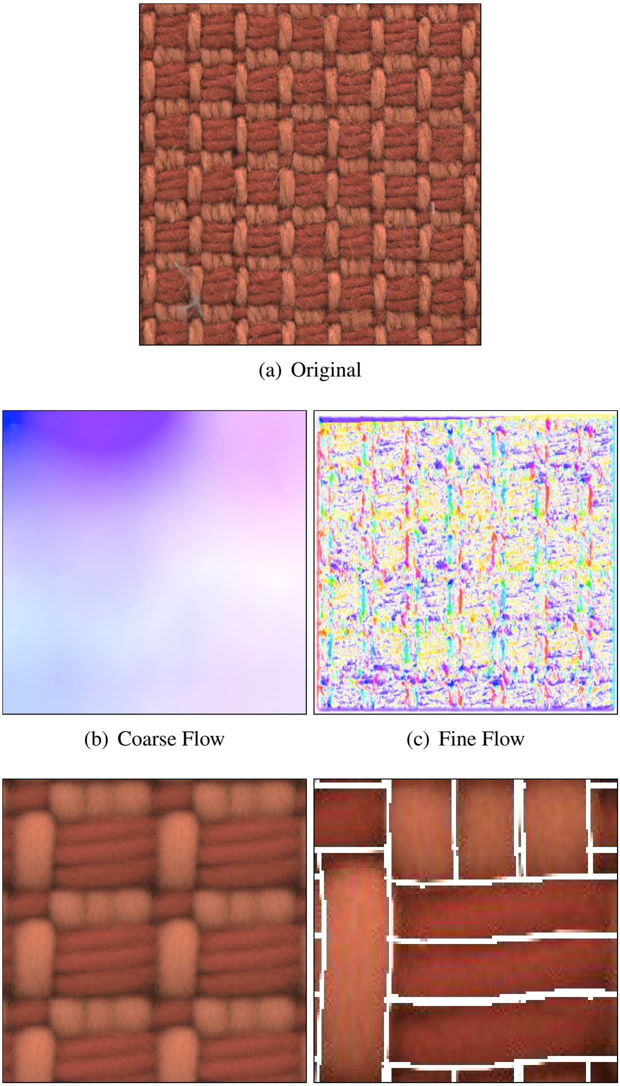
(b) Coarse Flow

(c) Fine Flow

(d) Reg. Median

(e) Segmentation

Figure 5.2: *Showing analysis steps for* 10×10 *weave with complex weave pattern and color effect.*



(a) Original

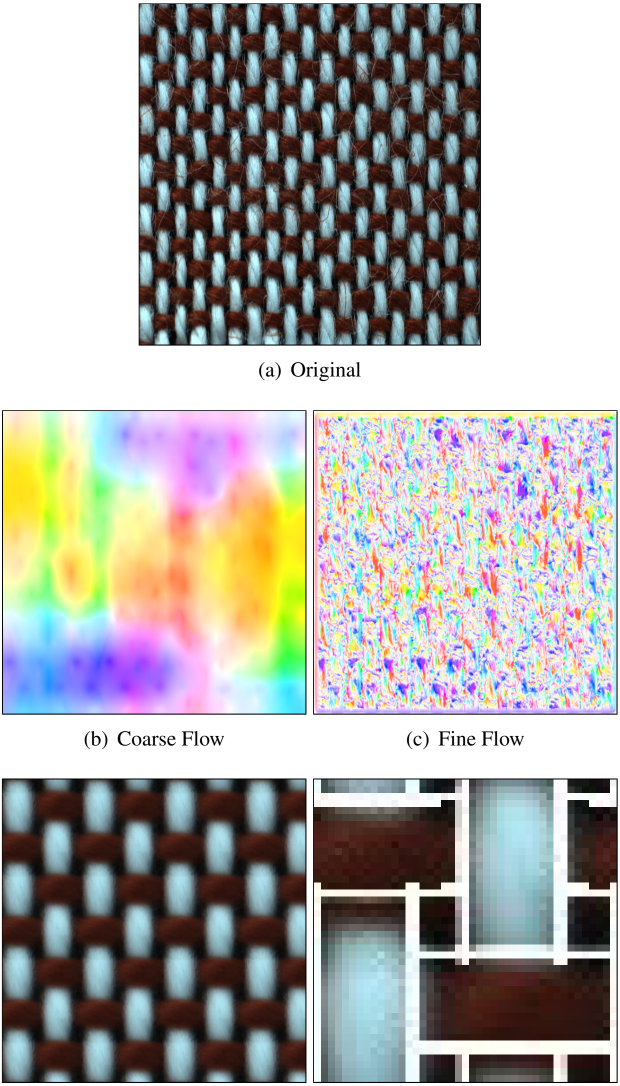
(b) Coarse Flow

(c) Fine Flow

(d) Reg. Median

(e) Segmentation

Figure 5.3: *Non-standard* 4×4 *weave pattern. Note how the cloth is bend in the original input image.*



(a) Original

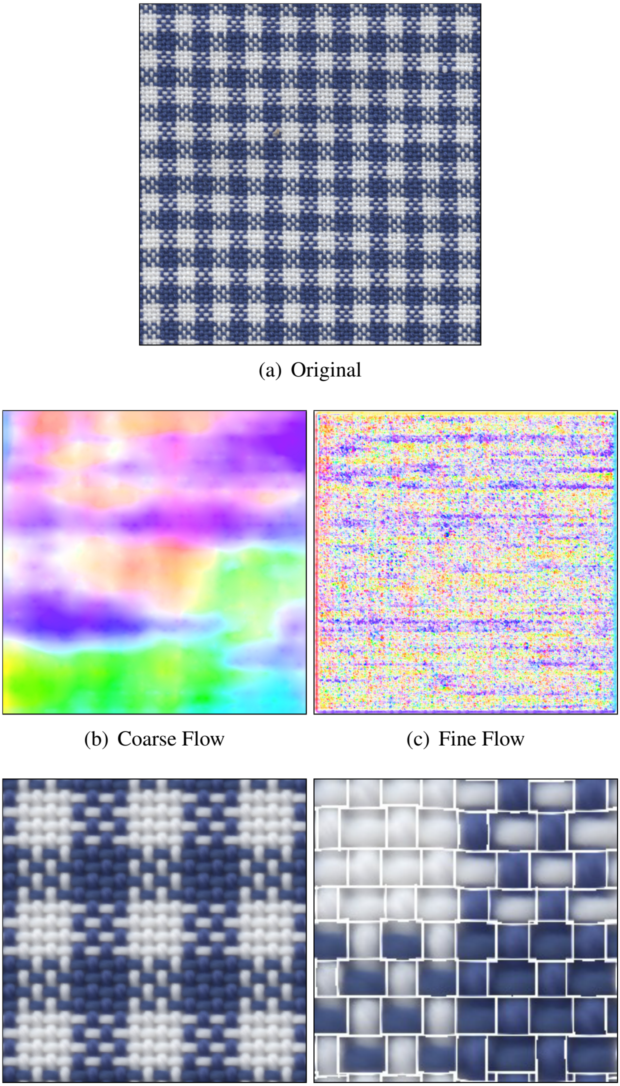
(b) Coarse Flow

(c) Fine Flow

(d) Reg. Median

(e) Segmentation

Figure 5.4: 2×2 *plain weave*



(a) Original

(b) Coarse Flow

(c) Fine Flow

(d) Reg. Median

(e) Segmentation

Figure 5.5: 8×8 *plain weave with color effect, captured on a flatbed scanner. While this setup is more challenging than a camera setup with diffuse lighting, as specular fibers lead to highlights, our method was still able to automatically segment the image.*

Part III

Visual Prototyping of Woven Cloth

CHAPTER 6

VISUAL PROTOTYPING OF CLOTH: OVERVIEW

The acquisition and representation of heterogeneous real world materials is challenging. Often, generic image-based techniques such as Bidirectional Texture Functions (BTFs) are used to capture complex lighting and shadowing effects. While measured data can be highly effective for reproducing the look of existing material samples, physically plausible editing of purely image-based representations is not easily possible – i.e. one cannot change the weave pattern. This is a problem for appearance design and *visual prototyping*. Ideally, one would like to be able to base editing operations directly on the elements of cloth.

To be able to realistically reproduce cloth, we not only have to reproduce its basic structured texture, but also its characteristic variations created during the many stages of yarn processing (see Chapter 2.2). These introduce local irregularities influencing texture and shading. A complete physically-based simulation of these processes would be costly.

Recently, Zhao et al. [ZJMB11, ZJMB12] have presented an elegant framework for computer aided design of cloth (discussed in Chapter 2.1.2). Unfortunately, Micro-CT scanners can only scan small samples and measurements are time-consuming and costly.

In the previous chapters, we have discussed how to estimate the properties of woven cloth at the yarn level from one single image only. We would now like to extend this captured information to a full renderable model. What is missing, is a representation of individual yarns at the fiber level, incorporating the effects of spinning. In the following, we propose a yarn model based on state-of-the-art textile research (Chapter 7). In Chapter 8, we combine this model with the results from our automatic analysis, obtaining a pipeline for visual prototyping of woven cloth. Figure 6.1 gives an overview of all steps. Using this pipeline we can not only reproduce the appearance of existing cloth samples, but we can also intuitively edit their properties. Part III is based on our publication [SZK15].

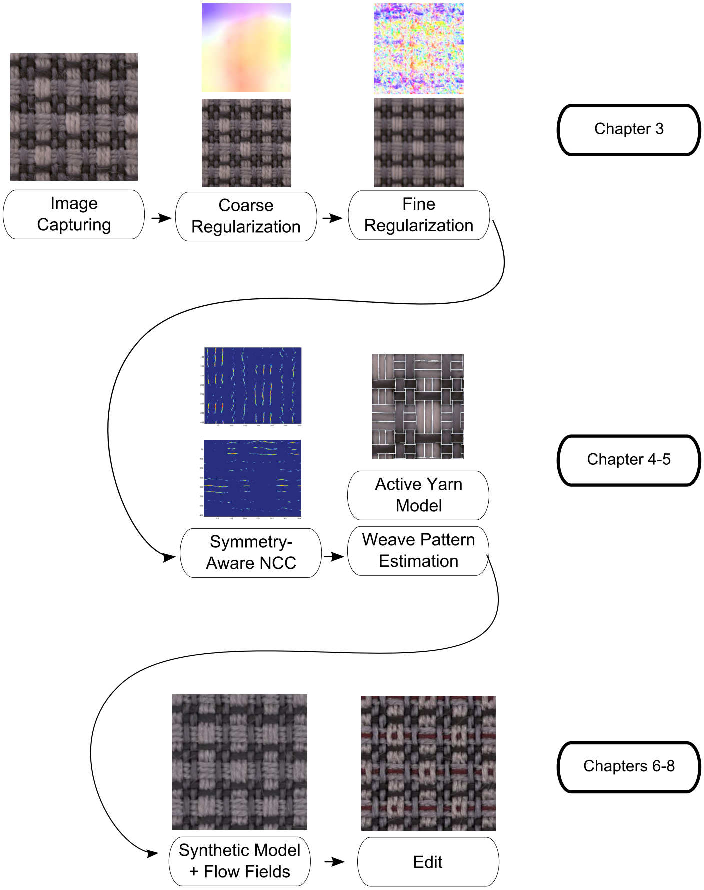


Figure 6.1: *Illustrating all major steps in our pipeline for visual prototyping of cloth. Only a cutout of the images is shown. From left to right, top to bottom: Input image / Coarse optical-flow field describing a smooth deformation and corresponding regularized image / Fine flow field describing local yarn deformations and corresponding regularized image / Visualization of the output of SANCC, demonstrating that we can effectively estimate initial center lines of yarns / Output of segmentation using active yarn model after automatic weave pattern segmentation / Rendering of reverse engineered model with manually set parameters for fiber distribution within yarns and for fiber materials / Example edit, changing weave pattern and color effect.*

CHAPTER 7

A PROCEDURAL YARN MODEL

Our goal is to create a pipeline for visual prototyping of cloth based on representing the micro-geometry as a procedural model of a fiber assembly whose parameters are based directly on the elements of cloth and their properties. Therefore, we need to be able to model the properties of yarns. As we have discussed in Chapter 2.2.3, spinning and weaving consist of several complex mechanical processes. In the context of graphics, it is important that a realistic cloth model can capture all major effects that directly affect the appearance. In the textile research literature, properties of yarns have been analyzed and different models have been proposed. At a high-level, our goal within this chapter is to model each yarn as a fiber assembly, where each fiber is associated with a BCSDF [ZW07b]. We combine the basic principles from the models of Sreprateep et al. [SB06], who describe a fiber-based yarn model for single ply yarn; Keefe et al. [Kee94] and Grishanov et al. [GLH^{+}97, GSC11a], incorporating their work on strand compression addressing effects of tension in double rove (two-ply) yarn and finally, we formulate a generative model for hairiness, loosely based on the model of Voborova et al. [VGNI04]. The resulting model looks plausible even in extreme closeup views and in contrast to existing data-driven approaches, we can easily synthesize yarns of arbitrary length, avoiding discontinuities of fibers.

Cross-Sectional Fiber Distribution: Regarding the fiber distribution within cross-section shape, it was observed [GLH+97] that fibers typically lie denser in the middle of yarns and become more coarse with increasing radius. Morris et al. [MMR99] have proposed the following distribution describing the probability of the existence of a fiber at radius R:

$$p(R) = (1 - 2\epsilon)\left\{\frac{e - e^{R/R_{max}}}{e - 1}\right\}^{\beta} + \epsilon \tag{7.1}$$

The function is parametrized by R_{max}, describing the maximum radius, where a fiber is observed and values ϵ, β controlling the density. It includes the uniform distribution as a special case ($\epsilon = \beta = 0$). We define β_i, ϵ_i and R^i_{max} for each ply i.

To avoid the intersection of different fibers, it is common [SB06, SGCB09] to divide the cross-section of yarns into a discrete number of virtual locations. Intersections can thus be avoided by making sure that each of these only contains a single fiber. However, as long as only the appearance of yarns is of concern, we have found the problem of fiber intersections to be of minor importance even in close-up views. This can be seen in the results, where hardly any fiber intersections are visible. Therefore, we simply use uniform sampling with rejection sampling to sample from the distribution in Equation 7.1.

Early work has described the twist of fibers in yarns using a coaxial helix model [Law03]:

$$x(\theta) = R(\theta)\cos(\theta), y(\theta) = R(\theta)\sin(\theta), z(\theta) = \frac{\alpha}{2\pi}\theta \tag{7.2}$$

where $R(\theta) = R$ is constant and equal to the distance of a fiber to the yarn center in the initial cross-section defined by Equation 7.1; α describes the length of one turn. When creating a thicker yarn from two spun threads / plies, we use this model when combining plies. In contrast, at the fiber level, we use a different approach:

Fiber Migration: Later, it was discovered that fibers do not always stay at a constant distance (radius) from the yarn center [Tre65, Law03] and also that the length of a turn can vary. Staple fibers migrate and can take a cone-like form along yarn length – long filament fibers instead behave more similar to a perfect helix. This was found to be an important factor for yarn appearance (especially visible for melange yarns, where fibers in a single yarn can have different colors [SGCB09]). While Siewe et al. [SGCB09] and Grishanov et al. [GSC11b] again use the concept of virtual locations to avoid intersections, we follow the approach from Tao et al. [Tao96] (as in [SB06]) and simply modulate the helix radius per fiber using the following function for $R(\theta)$:

$$R(\theta) = \rho^i_{min} r + \frac{\rho^i_{max} r - \rho^i_{min} r}{2} \Big(\cos s_i \theta + 1 \Big) \qquad (7.3)$$

parametrized by $\rho^i_{min}, \rho^i_{max} \in \mathbb{R}$ scaling the initial radius r (as sampled from Equation 7.1) of the current fiber and s_i represents the speed of fiber migration for each ply i. s_i has originally [Tao96] been split into migration height h and wavelength λ: $s_i = \frac{h}{2\pi\lambda}$, but we treat it as a single parameter per ply in our model. These parameters can be measured for real yarns (e.g. using a method similar to [HKR01]). Typical values can be found in Figure 7.1.

Two-Ply Yarns / Strand Compression: A naive approach to computing two-ply yarns works as follows: We can first compute a path for a ply's center by following the path of the yarn at some distance and at the same time rotating around it. We then evaluate Equation 7.1 for each ply, obtaining fibers within the ply's cross-section. These fibers then follow paths rotating around the ply's center path. However, this looks unnatural – in reality the strands push against each other and compress due to the tension from drafting and twisting. This effect has been modeled by Keefe et al. [Kee94] by representing the strands as ellipses where a line along the shorter diameter points to the yarn's center. Grishanov et al. [GLH$^+$97] refine this for the case of double rove yarn using a more physically-based approach where strands flatten at contact. While our model for strand compression is based on the one from Keefe et al., we incorporate the idea of strand flattening, by not only allowing the ellipses to touch each other (as in [Kee94]), but instead we allow them to intersect – it is hardly visible from outside if true flattening is computed (see Figure 7.1(d)). We

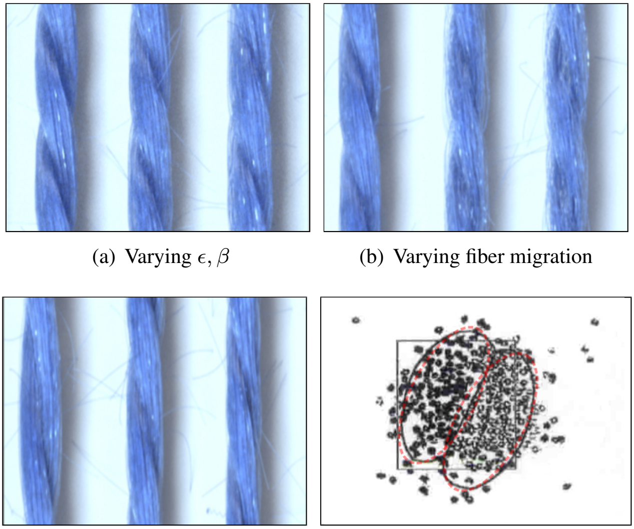
scale all cross-sectional positions by d_i along the axis defined by the line going through the ply's center and the yarn's center for the current cross-section. e_i gives the initial radius – respectively the other radius of the ellipsis. The index i again identifies the ply. In all of our examples, all plies of a yarn have equal properties.
Hairiness: The final part of a realistic yarn model is defined by the small fibers protruding from the yarn, called hairiness. Different types of hairiness exist [VGNI04]. One type is created by fibers leaving a yarn's body and then reentering it at another position forming loops. These commonly still follow the main orientation of the yarn. This type of hairiness is addressed to some extent by isolated fibers created when sampling from Equation 7.1 and especially by setting $\rho_{max} > 1$ for fiber migration.
Another important type of hairiness is given by fibers that fly away from the main body in various directions. Voborova et al. [VGNI04] describe an analytical stochastic model of yarn hairiness and an automatic method to derive model parameters from yarn images. They estimate the packing density of hairiness fibers on the main yarn body and for cylinders at greater radii. This gives a distribution describing at what distances from the body fibers end. We formulate a generative model based on these concepts and model hairiness fibers using particle simulations. We randomly select fiber vertices uniformly and let hairiness fibers grow, following the tangential direction. Fibers are terminated using Russian Roulette, evaluating a normal distribution $\mathcal{N}(\mu_i, \sigma_i)$ for mean fiber length μ_i and standard deviation σ_i for ply i. As a rough approximation to modeling *crimp* (fiber waviness / sudden orientation changes), we let a 3D Perlin noise [Per85] force field act on the fibers, with manually set parameters. Table 7.1 summarizes how common properties of yarns relate to parameters in our model.

2D Yarn mechanics	Section 3
Linear density	Fiber-count / -radius
Twist levels	s_i, α
Tension	e_i, d_i, α
Diameters of strands	d_i
Cross-sectional fiber distribution	$\beta_i, \epsilon_i, R^i_{max}$
Hairiness (loops)	$\beta_i, \epsilon_i, R^i_{max}, \rho_{max}$
Hairiness (fly,protruding)	μ_i, σ_i
Crimp of hairiness fibers	Perlin noise
Weave yarn compression	γ

Table 7.1: *Associating common properties of yarns with parameters in our model.*

7.1 Results

Figure 7.1 shows renderings of our yarn model. Rendered with a BCSDF, the results look photo-realistic and natural. The figure illustrates the effect of varying the different parameters. One can see that we can model both dense tightly twisted yarns, common for industrial woven cloth, as well as fluffy yarns, common especially in the context of knitwear. In the following chapter, we will describe a model for woven cloth that makes use of this yarn model.



(a) Varying ϵ, β

(b) Varying fiber migration

(c) Varying tension

(d) 2-Ply Cross-Section

Figure 7.1: *Renderings of our procedural yarn model, illustrating the effects of different parameters. (a-c): Varying different parameters of the procedural yarn model from left to right (a):* $\epsilon = 0, \beta = 0, R_{max} = 100\%$ */* $\epsilon = 0.03, \beta = 0.7, R_{max} = 115\%$ */* $\epsilon = 0.01, \beta = 1.2, R_{max} = 135\%$*. Shadows between plies become less pronounced and individual fibers start to become visible at yarn silhouettes (no fiber migration used for illustration of the effect, 250 fibers per ply). (b): no fiber migration /* $s_i = 0.7, \rho_{min} = 0.2, \rho_{max} = 1.2$ */* $s_i = 1.2, \rho_{min} = 0.2, \rho_{max} = 1.2$ *– other parameters: 250 fibers per ply,* $\beta = 1.0, \epsilon = 0.03$ *(c): Increasing twist and reducing shorter radius of ellipses to simulate increasing tension. Note how yarn silhouettes become flat due to the flat ellipses of plies. (d): Cross-sectional shape of double rove yarn (taken from [GLH$^+$97], overlaid with illustration for our model). Comparing measured fiber locations using unfilled circles for the first ply and filled circles for the second one in cross-section with the model from [GLH$^+$97] (black line) and theoretical boundary for our approximation (red dashed line). In practice we synthesize a fiber based model so that the boundaries of the ellipses are not well defined; in this case ellipses naturally intersect each other to some extent – the right most yarn in (c) uses a distribution comparable to this one.*

CHAPTER 8

VISUAL PROTOTYPING OF WOVEN CLOTH FROM A SINGLE IMAGE

In this chapter, we combine results of the previous chapters to propose a full pipeline for visual prototyping of woven cloth. We describe a woven cloth model that makes use of the automatic analysis from Chapters 4 and 5 to create a model that can not only capture the basic structure of a given cloth sample, but also contains its natural variations, creating its specific look and feel.

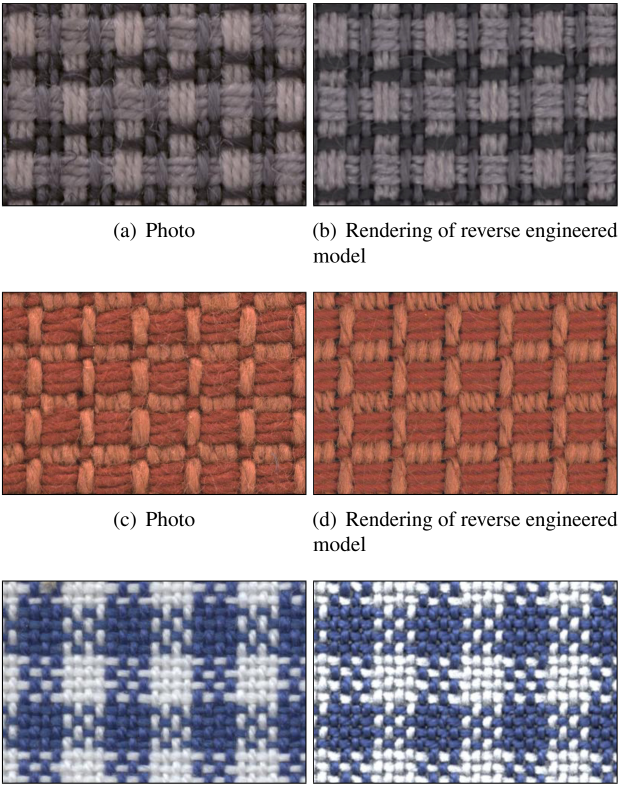
Combined, we obtain a full procedural model of micro-geometry for a photographed cloth sample. Parameters for the yarn-model are set manually, using a few well defined parameters. It is very common that only two different types of yarns – e.g. having different colors for warp and weft yarns – are given, making this highly practical. To describe optical fiber properties, we manually set parameters for bidirectional curve scattering distribution function (BCSDF) [ZW07b]. We concentrate on geometric aspects in this part, because we believe that such fiber scattering functions will be provided by yarn manufacturers in the future, making an automatic analysis of the basic cloth structure an even more important part of a cloth design pipeline.

Once we have obtained parameters of the model, we can not only match the appearance of an acquired cloth sample in the input image, but we can also intuitively edit it by changing e.g. the weave pattern, the color effect (fiber material) or geometric yarn properties.

8.1 Woven Cloth Model

To model woven cloth, we let all yarns follow the paths, extracted from the image segmentation process interpolated using B-splines. To some extent, this lets us incorporate the mechanical properties of yarns. However, it does not give us the 3-dimensional shape of yarns. Due to *sizing* (see Chapter 2.2.2) and weaving, yarns are compressed – especially along the normal direction of a flat cloth sample. In practice, the amount of compression changes depending on bending and contact with neighboring yarns. Gong et al. [GOS09] compare an elliptical fit of the cross-sectional shape of yarns with x-ray scans of yarns in plain-woven fabric. They found an ellipsis to be a suitable model, and they create a yarn model by connecting consecutive variable cross-sections. An alternative is to perform physically-based simulations and finite element analysis [VBP10].
In this work we approximate compression as follows: We create cloth with yarns of circular cross-section in case of single-ply yarns; for two-ply yarns the cross-section consists of the two ellipses as described above. We scale all vertices by amount γ along the main normal direction of the flat sample. The basic up-down movement of yarns is given by the weave pattern. At cross-section c_{ij}, we create $3D$ control points for yarns $\bar{Y}_i$ and $\hat{Y}_j$ from $c_{ij} = (c_x, c_y)$: $\bar{p}_{ij} = (c_x, c_y, (-1+2L_{ij})0.5\bar{w}_i)$ and $\hat{p}_{ij} = (c_x, c_y, (1-2L_{ij})0.5\hat{w}_j)$. To avoid yarn intersections, we add additional spline control points for $\bar{Y}_i$: for all i, j and $\bar{p} = \bar{p}_{ij}$ we add points $(\bar{p}_x - \omega\bar{w}_i, \bar{p}_y, \bar{p}_z)$ and $(\bar{p}_x + \omega\bar{w}_i, \bar{p}_y, \bar{p}_z)$ (similar for y-coordinates and $\hat{Y}_j$). For all results, we choose $\omega = 0.5\gamma$.

We compute reverse flow fields for the fields computed in Chapter 4 using cubic interpolation. These can then be applied to obtain the effects of both the low-frequency cloth deformation and the local changes of yarn paths and yarn widths.



(a) Photo

(b) Rendering of reverse engineered model

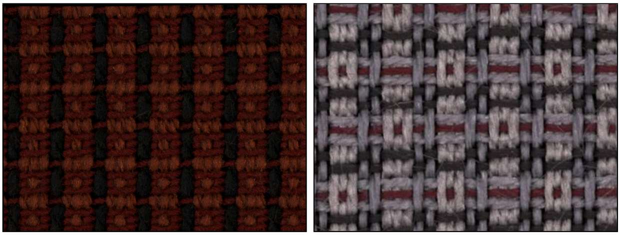
(c) Photo

(d) Rendering of reverse engineered model

(e) Photo

(f) Rendering of reverse engineered model

Figure 8.1: *Comparing photos (after shearing) with our synthetic model. Note that colors and lighting setup have been matched manually. Our model is able to capture the basic appearance of all of these types of cloth. Except for small differences in intensity and contrast, resulting from the manual modeling of lighting and optical properties, the fine visual detail of yarns is captured well.*



(a) Edit of 4 × 4 cloth

(b) Edit of 10 × 10 cloth

Figure 8.2: *This figure shows two edits of samples, we have reverse engineered.*

8.2 Results

We present synthesis results for all cloth samples presented in Chapter 5. During the synthesis step, the procedural model combines the automatically extracted yarn level information, taking the deformation field as input. We have set parameters for the geometric yarn model (twist, hairiness, cross-sectional fiber distribution / density, fiber migration, tension and fiber radius) as well as optical fiber properties (BCSDF) manually. Parameter selection was performed using an interactive user interface, taking only a few minutes.

Renderings of our model using Monte Carlo path tracing of a fiber assembly are compared to input images in Figure 8.1. The basic characteristics of the samples are well matched when compared to the input images. The remaining difference is to some degree related to having to match both lighting and optical fiber properties manually.

Figure 8.3 compares our full approach with first, a simple procedural yarn model and second, with our yarn model but not using the yarn variation obtained from deformation fields. One can see that our full approach looks most natural.

Moreover, Figure 8.2 and Figure 8.4 (b) and (c), show drastic edits of the estimated cloth models.

8.3 Limitations

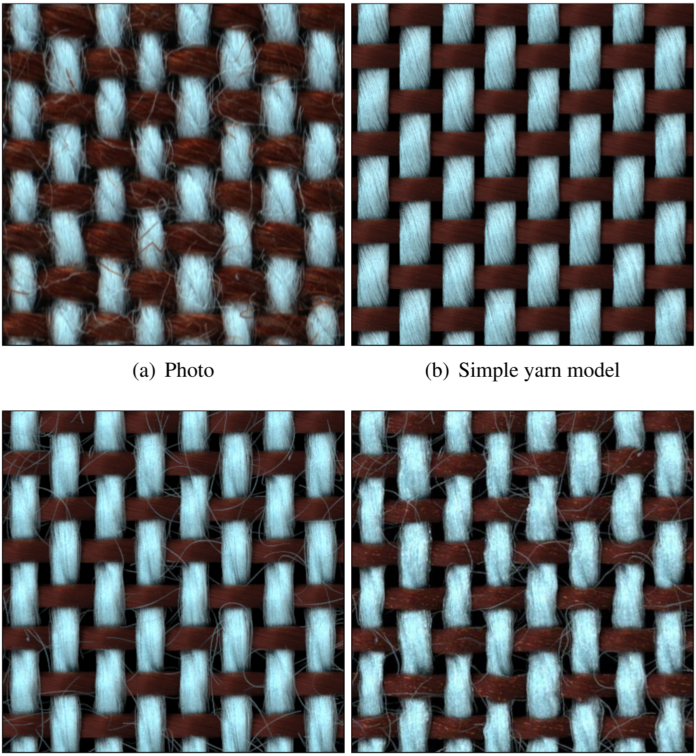
A major limitation of our approach is that parts of the synthesis pipeline still rely on the manual selection of model parameters. While being an advantage for editing, finding suitable values for reproducing a desired appearance will require a certain amount of user experience.

While estimating the reverse flow fields is a good first step to modeling deformations of yarn geometry, these cannot directly be used to exactly infer the width variation of yarns. The main reason is that different effects leading to width variation (twisting, perspective effects and yarn unevenness) cannot be separated. In addition, the inverse flow field applied to the cloth model is only capturing 2D deformation in the image plane. While this may be a reasonable approximation as stretching & shearing mostly matters in the tangent plane, some information gets lost. Further, the procedural yarn model itself exhibits a width variation, which would have to be taken into account.

When changing the weave pattern, we currently only change the values of matrix L. However, in reality also the distances between yarns can change. If two yarns next to each other follow the exact up/down rhythm as given by the weave pattern, there does not have to be any spacing between them as no orthogonally running yarn has to fit between them. This is why twill weave cloth can have a higher *yarn count*, having more yarns per area, than for example plain weave [Sha07].

8.4 Conclusions and Future Work

In this section, a pipeline for analysis and synthesis of woven cloth based on a single image has been presented. By combining research on fiber-based yarn models (Chapter 7) with an automatic approach to analyze the structure of cloth at the yarn level (Chapters 4 and



(a) Photo

(b) Simple yarn model

(c) No warping

(d) Rendering of our full approach

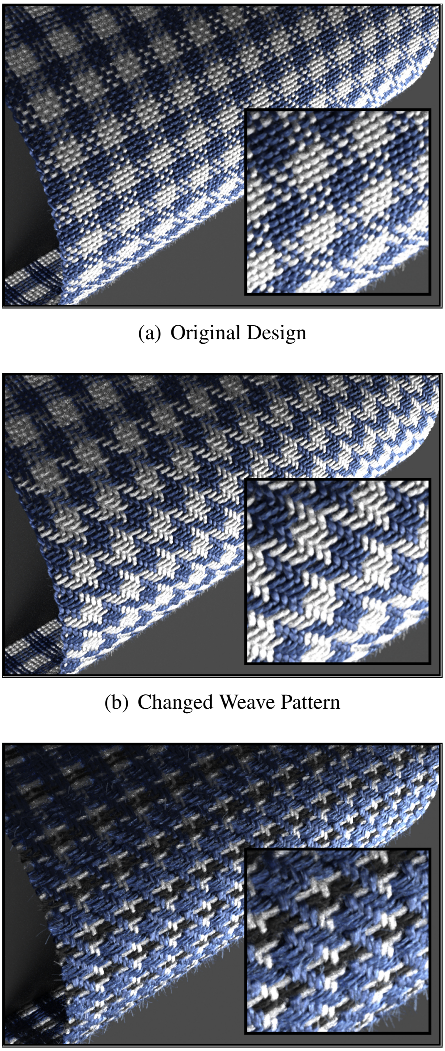
Figure 8.3: *Demonstrating the effects of a complex yarn model and deformation fields. From left to right, top to bottom: Input photo (cutout) / Using a standard procedural single helix yarn model / Our yarn model, but no application of deformation fields / Our full approach. As our approach also captures the typical noisy deformations, the variation of yarn widths and the characteristics of yarn geometry, it looks more natural than simple procedural models.*

5), we were able to model the appearance of several existing cloth samples, closely matching the input image. This model could then be used as a starting point for editing operations, allowing for much faster iterations in textile design when compared to the still dominant practice of explicitly manufacturing cloth prototypes.

In computer graphics, there is a large body of physical cloth simulation, which at best deals with simulation at a yarn level [KJM08, YKJM12] that could help to further improve physical plausibility. Also the behavior of yarn cross sections during the weaving process could be based on a physical simulation (as in [LZS+12]). To synthesize geometry of arbitrary size using the current system, one has to synthesize larger flow fields. This can be easily done by applying constrained texture synthesis to the fine flow field in a way similar to Figure 4.5(d), where we have synthesized fine detail of fibers onto the median image. The smooth coarse flow field could be extrapolated. An alternative approach to representing yarn variation would be to estimate parameters of a noise model describing changes in yarn widths and locations.

Because the described problem is already very challenging, we have for now concentrated on flat samples – this has the additional advantage that we can e.g. put them onto a flatbed scanner. It will be interesting to combine the presented approach with cloth simulation and surface flattening [ABW94],[ADBW96],[ABW01],[WTY05], [WT07] in the future.

Instead of measuring a database of woven cloth samples using Micro-CT scanning (as in [ZJMB11]), one could create a database of individual yarns with different properties using Micro-CT scanners and use this to estimate parameters for the procedural model. Compared to [ZJMB11] this would have the advantage that not all pair-wise combinations of yarns next to each other with varying weave patterns have to be captured and that closeup views would still look plausible as no discontinuities occur as for volumetric patch based synthesis, where often not all fibers are exactly continued in a neighboring patch.



(a) Original Design

(b) Changed Weave Pattern

(c) Changed Color Effect and Yarn Geometry

Figure 8.4: *Example for visual prototyping of cloth using our framework. Starting with an input photo, we synthesize and render cloth based on a physically-plausible procedural model (a). Our framework allows for intuitive editing (b),(c).*

Part IV

Physically-Based Cloth Rendering

CHAPTER 9

CHALLENGES IN PHYSICALLY-BASED CLOTH RENDERING

Predictive rendering of cloth is a challenging task. As already discussed, *geometrical complexity* and *optical complexity* have to be handled. Highly anisotropic single and multiple scattering effects, as well as self shadowing effects, dominate the appearance. As many types of fibers are highly translucent, multiple scattering significantly influences the observed color. This is especially true for light-colored fibers with low absorption.

As a cloth model may consist of potentially hundreds of millions of fibers, finding a viable level of geometrical abstraction is challenging. When designing a physically-based cloth rendering solution in this work, we have two main applications in mind: The first one is to design a piece of cloth virtually using visual prototyping and being able do interactively change the appearance. The second one is to efficiently render images of a given piece of cloth with fixed properties. We use different material representations for these tasks. For visual prototyping on the one hand, we want to be able to edit and, therefore, represent the three main entities / scales [1] of cloth individ-

[1] optical fiber properties, yarn geometry and compositional structures

ually. For pure rendering applications of cloth with fixed material properties on the other hand, more computationally efficient aggregate representations can be used.

In the previous chapter, we have presented an approach to obtain a cloth model as an explictly represented fiber assembly.

Explicit methods If we have obtained a fiber-based model with associated material information, e.g. in the form of a *Bidirectional Curve Scattering Distribution Function* (BCSDF), a straight forward idea is to use Monte Carlo path tracing for the explicit model, as used in the previous chapter. While this can be done relatively efficiently, within the limits of path tracing based solutions, the approach does not scale to rendering large pieces of cloth. Even small patches with 50 warp and 50 weft yarns easily take up dozens of gigabytes of memory. Depending on the application, geometry instancing could be imagined as a work around. However, this approach has limitations. Even if a repeating weave pattern is observed, which is not always the case, the geometry of spun yarns commonly does not repeat with the same frequency. This is due to the fact that cloth is a soft material that can be drastically deformed resulting in stretching and squeezing of yarns. Furthermore, *fancy yarns* may include spatial variations themselves along their length, leading to the specific look and feel of a cloth sample. Even if we assume a repeating pattern and ignore spatial variations, instancing is non-trivial because the geometry will be different everywhere due to bending.

Image-based techniques such as BTFs are a common approach to render measured cloth samples. As light transport within the material is captured in the images, the fibers do not have to be represented explicitly and also multiple scattering between fibers does not have to be calculated when rendering with BTFs. This makes BTFs efficient for rendering on graphics hardware, allowing for real-time rasterization in case of simple lighting setups, while modeling many important effects. However, BTFs cannot easily be edited.

In Chapter 10, we present an approach combining the advantages of micro-geometry based rendering with those of image based approaches, by creating a virtual gonio-reflectometer setup to capture synthetic BTFs. To make this practical, we present a method that employs self-similarity to accelerate the computation.

A different rendering approach is presented in Chapter 11. It allows us to render large pieces of cloth even if they cannot be represented by a repeating pattern. We present a framework for volumetric rendering of cloth based on a statistical description of micro-geometry. It uses the same fiber scattering models we have applied to fully explicit models. The approach allows us to interactively change the properties of individual fibers and simulate a change in cloth appearance.

CHAPTER 10

Non-Local Image Reconstruction for Efficient Computation of Synthetic Bidirectional Texture Functions

While explicit representations of micro-geometry work well when designing a material in a close-up view as presented in Chapter 8, they may easily become too costly in terms of both memory consumption and rendering times, when applied to whole objects and large scenes. In this chapter, we propose an efficient method to synthesize BTFs based on known micro-geometry, combining the strengths of both approaches. During the design phase, the micro-geometry can be rendered explicitly for a small patch and afterwards a BTF can be synthesized for interactive inspection.
To compute a BTF, a virtual gonio-reflectometer setup can be constructed, which computes the angular reflectance data for a virtual material sample. Essentially, light transport has to be simulated for thousands of viewing and lighting directions. A straight forward solution is to build a virtual half-dome multi-view multi-light measurement device and render e.g. 70×70 images (”BTF-slices”) of a flat material sample for 70 viewing and 70 lighting directions.

Unfortunately, physically-based rendering of thousands of images, commonly done using a path tracing approach, is extremely costly. Similarly, the acquisition of BTFs of real materials, where samples are

usually photographed under several viewing and lighting directions, is time consuming.
As BTFs contain much redundant information, a natural question that arises is, whether all view and light directions have to be captured with equal quality. For a real measurement device, quality and measurement time can be controlled by e.g. setting the exposure time. This controls the amount of photons that hit the sensor, resulting in different noise levels. Similarly, for a virtual setup, the amount of computed samples per pixel controls both the quality and computation time. In this chapter, we use directions captured with high-quality to improve images for directions captured with a lower quality.

Recent work in the image filtering literature has shown that internal

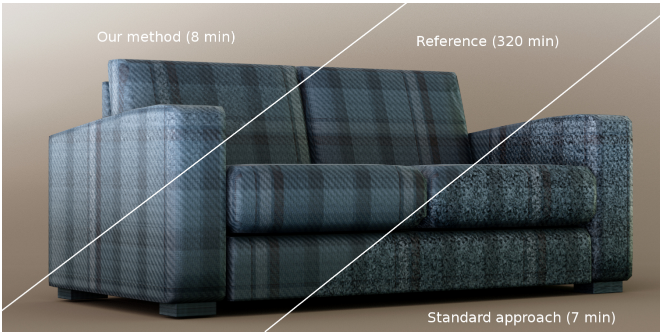


Figure 10.1: *A couch textured with a repeating BTF pattern, illustrating BTF synthesis results. From left to right: Our method using 1 sample per pixel (spp) and non-local image reconstruction for BTF generation; unbiased Monte Carlo reference with 128 spp; unbiased Monte Carlo path tracing result with 1 spp. Synthesizing the full BTFs took 8 minutes, 320 minutes and 7 minutes respectively on an Intel Core i7 CPU 950 (only counting the time for rendering the BTFs and not for rendering the images of the couch). The remaining variance in our BTF reduces to imperceptible levels with 2 to 4 spp.*

image statistics can be a powerful prior for image denoising. The repetitions in an image often contain more information about image patches than data obtained from large external image databases [ZI10]. This property is used in the popular non-local means (NL-means) [BCM05] filtering technique. Here, the original value of a pixel is replaced by a weighted average of all pixels in the image showing a similar intensity distribution of surrounding pixels. Unfortunately, the technique breaks down for renderings with a very low signal-to-noise ratio. We were the first to adapt this general idea to accelerate rendering [SMKZ11, SKZ13]. Similar in spirit to NL-means, we take advantage of the fact that BTFs tend to contain many very similar *apparent bidirectional reflectance distribution functions* (ABRDFs [WHON97b]). Exploiting this property, we reduce variance by sharing radiance samples across different BTF texels. Essentially, we build a neighborhood graph in appearance space, connecting texels of similar appearance, and averaging image samples among them.

10.1 Related Work

In the context of animation rendering, different techniques have been proposed to reuse computed light paths for neighboring pixels, which are spatially or spatio-temporally close to each other [MFSS06, SC04, CS10, HDMS03]. Although we do not need any explicit handling of light paths in our algorithm, our approach shares some conceptual similarity to these techniques. However, the proposed solution is even simpler as we do not need to store information about paths.

Much work has been invested in finding methods for variance reduction of path tracing solutions. Several authors have proposed different local filtering techniques [RW94, JC95, McC99, XP05].

Variance reduction for Monte Carlo path tracing is also similar to denoising of natural images. A major difference is the type of noise distribution commonly observed. Classical image denoising techniques such as *bilateral filtering* [TM98] can directly be applied. Nearby

pixels are averaged if and only if they have a similar color. Different variants of bilateral filtering have been applied to variance reduction [WKB$^+$02, SIMP06, LSK$^+$07].
Cross-bilateral filtering attempts to identify similar pixels by not only using the color of a pixel but also by using information of e.g. other sensors, which can be related to pixels in the original image [PSA$^+$04, ED08]. In the context of rendering, this can be information from a computed normal map or depth buffer ([DSHL10, BEM11] use a similar idea). Recently, Sen and Darabi [SD12] have presented an algorithm for denoising rendered images that estimates local weights for a given set of features by using knowledge about the random number generation process of the Monte Carlo path tracer. Lehtinen et al. [LAC$^+$11] have developed a special reconstruction filter to reduce the variance for effects such as depth-of-field and motion blur. Another method loosely related to our work was proposed by Hašan et al. [HPB07] to speed up rendering in the context of the many light problem.

Previous work has used appearance similarity based clustering in the context of BTFs and spatially varying BRDFs. Suykens et al. [SvBLD03] synthesize BTFs by rendering micro-geometry and afterwards they cluster ABRDFs to obtain prototypes that form a compact representation of BTFs for hardware rendering. We instead use the similarity of ABRDFs directly during BTF synthesis. Zickler et al. [ZREB06] combine BRDFs measured for different spatial locations to obtain a higher reflectance sampling in the angular domain. They assume smooth spatial variation and angular compressibility of BRDFs to be able to recover dense reflectance data from sparse angular samplings. In contrast, we do not assume any spatial smoothness and we use angular compressibility only to be able to identify similar ABRDFs, while the angular variation of ABRDFs themselves is just restricted by the discrete sampling.

Another strategy for variance reduction is to automatically find image parts with high variance and adaptively shoot more rays to these locations [Pur87, Mit87, TJ97, Mac02, RFS03, HJW$^+$08, ODR09, RKZ11]. Our method is orthogonal and could be combined with adaptive sampling.

BTF synthesis by rendering of micro-structure could also be regarded as an instance of precomputed radiance transfer (PRT) (see [KSL05] for a survey).

None of the above techniques exploits non-local information in a way we do.

10.1.1 Non-Local Means Filtering

The aim of the non-local means algorithm [BCM05] is to compute an estimate $\mathrm{NL}(I') \in \Omega \subset \mathbb{R}^2$ of an original image $I \in \Omega$ for which a noisy measurement $I' \in \Omega$ has been observed. This is done by combining intensity values at arbitrary image regions that show similar structures. Appearance similarity is measured by comparing the intensity values of neighborhoods (patches P) around pixels.

$$\mathrm{NL}(I'_{p'}) = \sum_{p \in \Omega} \frac{w\left(d(P_{p'}, P_p)\right) I_p}{\sum_{\hat{p} \in \Omega} w\left(d(P_{p'}, P_{\hat{p}})\right)} \tag{10.1}$$

Here, d is a distance measure of patch similarity, computed as sums of squared differences, weighted by the distance to the center pixel p, respectively p':

$$d(P_{p'}, P_p) = \sum_{i \in \mathcal{N}} G_h(\| \, p - p' \, \|) \left(I_{p'}(i) \; - \; I_p(i)\right)^2 \tag{10.2}$$

where $\mathcal{N}$ is a (patch) neighborhood of pixels p, p' and G_h is a Gaussian with variance h.

In the classic *non-local means* algorithm the existence of additive white Gaussian noise with zero mean is assumed [BCM05]: $I'_p = I_p + G_p$.

The noise generated by Monte Carlo path tracing in our application has a very different characteristic, including salt and pepper noise and

may show extremely low SNR, particularly in the presence of a high dynamic intensity range. Different techniques have been proposed to handle various noise distributions in the non-local means framework. In Equation 10.1, w is a Gaussian filter function with a manually selected variance σ: $w(x) = \mathrm{e}^{\frac{-x^2}{2\sigma}}$
Every pixel in the denoised image I is formed by a combination of all pixels in the observed image I' weighted by w. To reduce higher amounts of noise, larger values for σ have to be used. This increases the chance that less similar patches are averaged. To minimize this effect, several authors [GLPP08, PGM10] have explored the use of different robust functions instead of the Gaussian filter. Another variant of NL-means uses adaptive dictionaries in a Bayesian framework [KBC07]. This formulation can be used to handle speckle noise [CHKB08]. Other variants of the approach work on patches of pixels to both speed-up computations and find better neighborhoods [DFKE06].

Non-local means filtering has originally been inspired by texture synthesis by non-parametric sampling [EL99]. In this work, we apply the concept of sharing information non-locally to a different type of texture synthesis, namely synthesis by rendering micro-geometry. Related problems can be found in the field of inpainting, being even more challenging because there is no information available at all for some image regions. Information from probably disconnected image regions can be combined to fill the unknown regions with plausible content (see e.g. [WSI07]).

10.2 BTF Synthesis Using Non-Local Image Reconstruction

Our goal is to reduce the computational effort for synthesizing a BTF by reusing samples gathered during Monte Carlo simulation in a non-local fashion. More precisely, we wish to exploit structural self-similarity in the BTF domain. In the general case, this would mean a sharing of samples in the 6D spatio-angular space. However, to reduce complexity, we consider 2D-subspaces, given by spatial slices for fixed viewing and lighting direction pairs. Despite the fact that this is a quite natural choice inspired by the way BTFs are commonly measured, this particular choice has several advantages. Neglecting global effects and assuming a flat sample, this intuitively means that similar materials show a similar appearance. Thus, if two ABRDFs behave comparably for certain viewing/lighting directions, this will increase the chance that they behave comparably in general. Note that being able to identify such similarity based on a sparse set of appearance samples is of fundamental importance to our approach.

In the following, we describe how thousands of images, representing ABRDF slices, can be rendered efficiently in a path tracing framework, using an approach, we call non-local image reconstruction. Here, every radiance sample contributes to several pixels at completely different locations in an image (see Figure 10.2). We exploit redundancies in the spatial domain, in the spirit of non-local means, minimizing the required sampling rates for generating high-quality BTFs. For every pixel, we compute a set of k nearest neighbors according to similarity of appearance. Every radiance sample that is computed for an image pixel also contributes to all of its neighbors.
While standard non-local means filtering cannot know the correct neighborhood in appearance space and patch similarity is used to establish correspondences for noisy data, we have the possibility to estimate neighbors more reliably by using the assumption of compressibility of ABRDFs in the angular domain. Essentially, we compute noise free feature vectors – to make this computationally feasible we only compute them for a sparse angular sampling. Therefore, we replace the search for appearance neighbors in a set of noisy samples with a search for neighbors in a sparse but (almost) noise-free approximation of the original data and therefore, we become independent of the underlying noise distribution of images rendered with low sampling rates when finding neighbors.

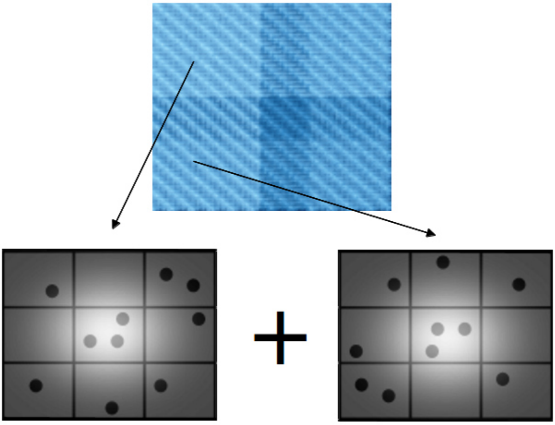


Figure 10.2: *Every pixel is connected to a number of similar pixels in a graph. In the image we show two such pixels. During rendering, we compute different radiance samples for each of them. By distributing samples of one pixel to the other one, we essentially get a higher number of spp.*

10.2.1 Non-Local Image Reconstruction

To generate a slice, we shoot rays into the scene, returning n radiance values L_i. In the following, $\Omega_{v,l} \subset \mathbb{R}^2$ is the image space of a spatial slice $A_{v,l}$ through the $6D$ BTF, depending on light direction l and viewing direction v. Reconstructing a single pixel of a slice can be done as usual by averaging radiance values L_i, multiplied by a value of a function F_i that depends both on the pixel location p and on the location of the current sample i:

$$\forall p \in \Omega_{v,l} : A_{v,l}(p) = \sum_{i=1}^{n} L_i F_i(p) \tag{10.3}$$

Raytracing renderers commonly use a local image reconstruction filter $F_i := F_i^L$ that computes the contribution of the i-th sample to pixel p, normalized by the contributions of all samples with respect to a possibly unnormalized filter f_i:

$$F_i(p) := F_i^{\mathrm{L}}(p) = \frac{f_i(p)}{\sum_{j=1}^{n} f_j(p)} \tag{10.4}$$

In practice, most samples do not contribute to a certain pixel because common filter functions such as box filters, cropped Gaussian filters or the Mitchell filter [MN88] have finite support. To further reduce variance, we instead define a non-local image reconstruction function $F_i := F_i^{NL}$, associated with local filter function f_i, where each sample contributes to several pixels connected by a k-nearest neighbor graph:

$$F_i(p) := F_i^{\mathrm{NL}}(p) = \frac{\sum_{p' \in \Omega} w(p,p') f_i(p')}{\sum_{j=1}^{n} \sum_{p^* \in \Omega} w(p,p^*) f_j(p^*)} \tag{10.5}$$

with weights:

$$w(p,p') = \begin{cases} 0 & \text{if } p' \notin \mathcal{N}_p \\ e^{-d(V_p, V_{p'})^2/2\sigma} & \text{if } otherwise \end{cases} \tag{10.6}$$

where $\mathcal{N}_p = \{p' \,|\, p' \in \text{k-nearest neighbors of } p\}$. The distance $d(V_p, V_{p'})$ computes the similarity of feature vectors belonging to pixel p and pixel p'. In the following section, we discuss these features in more detail.

10.2.2 Appearance Space Similarity

To compute the set of neighbors in appearance space, we first compute high-quality, sparsely sampled ABRDFs — full spatial slices for a small set of fixed viewing and lighting directions. These will be called *N-slices* in the following. This is achieved by rendering the material sample using Monte Carlo path tracing at a high sampling rate. The exact choice of this rate highly depends on the observed material and the rendering system – in all of our examples we use 256 spp. The directions have to be selected in such a way that they provide the discriminative power to robustly identify similar ABRDFs. Finding an optimal set of directions is challenging. On the one hand, these should include a low amount of redundant information. On the other hand, they should reveal the distinctive ABRDF features. Practically, we have selected five (figures 1 to 6) or nine (for the more challenging examples in figures 7 to 13) different direction pairs: The normal view with coaligned lighting, a pair of directions that capture scattering along the specular direction, e.g. hold $\theta = \theta^*$ fixed for both view and light direction and choose a pair of ϕ^* and $\phi^* + \pi$ for the other angle. Further pairs are distributed over the hemisphere capturing off-specular scattering. Generally, we choose more angles with small θ to obtain a better result for non-grazing angles.
This results in a feature vector V_p for each texel at a position p. Distances $d(V_p, V_{p'})$ between two feature vectors are simply computed as Euclidean distances. The overhead introduced by rendering the high-quality slices is low when compared to the computation time for the full BTF, consisting of several thousands of slices.

Next, taking these ABRDF samples of a BTF texel as a feature vector, we search for the k nearest neighbors using approximate nearest neigh-

bor search in a kd-tree. Using this information, we build a k-nearest neighbor graph connecting similar texels.
As known from probability density estimation, the k-nearest neighbor search we employ, automatically adapts to the density (in our case of ABRDFs in appearance space), whereas the classical non-local means computes a Parzen window estimate with a fixed kernel size, which may work poorly in areas of low density. This is why some research has gone into adaptive selection of the kernel width.

Note that the ABRDFs are influenced by near geometry via self occlusion effects, shadowing and inter-reflections. Therefore, although we do not explicitly make use of spatial neighborhoods, as patch based similarity measures do, our feature set does include some of that information implicitly. Once the neighborhood graph has been constructed, it can be used to reduce variance when rendering a full BTF, as described in the previous section.
In practice, we use the k-nearest neighbor graph during rendering and immediately update the pixels of all neighbors once a new radiance sample has been estimated. The computational overhead that is introduced by this modified image reconstruction algorithm is negligible, assuming the support of the associated local reconstruction filter f_i is not much larger than the area of a pixel and when using moderate neighborhood sizes such as e.g. 48 neighbors.

10.3 Results

We made both visual and numerical tests to evaluate the effectiveness of our approach.

Visual Analysis: In Figures 10.1, 10.10 and 10.5, we compare our non-local image reconstruction to unbiased path tracing results for selected, challenging examples. As can be seen, the technique is quite effective for variance reduction and shows a substantially better rate of convergence. A speed-up of about one order of magnitude can

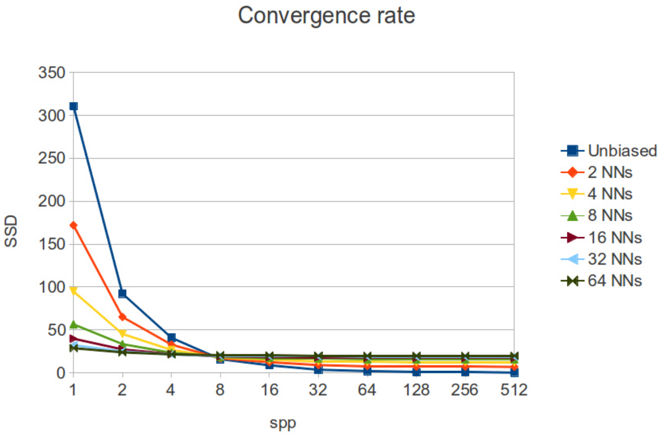


Figure 10.3: *Comparing rate of convergence: Sum of squared differences (SSD) error for our approach and unbiased path tracing (blue curve) against a 2048spp reference for the example shown in Figure 10.5 for different sampling rates and different numbers of nearest neighbors (NNs).*

be achieved for optically complex materials; disregarding the time needed for disk-IO, speed scales linear with the sampling rate.

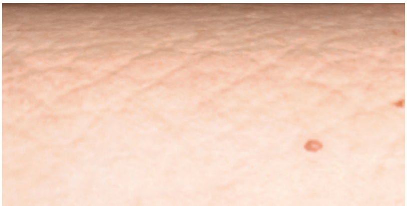
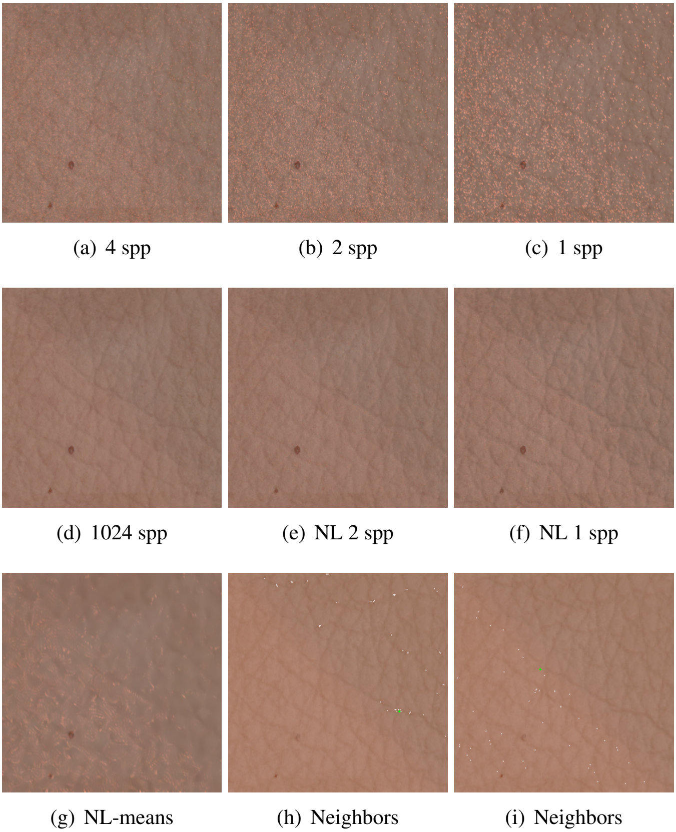


Figure 10.4: *Real-time rendering of a synthesized skin BTF on a cylinder under a point light after compression using full matrix factorization.*

For Figures 10.1 and 10.10, a BTF of size 128×128 pixels with an angular sampling of 70×70 viewing and lighting directions has been synthesized. For all other examples, BTFs of size 256×256 pixels with an angular sampling of 81×81 directions have been computed.

Figure 10.5 and 10.4 show a complex skin material, including volumetric subsurface scattering effects. It indicates that our approach can be effective even in case of non-exact repetitions and can handle a variety of different lighting effects such as highlights and light angle dependent color changes (see Figure 10.4 for a real-time rendering of the synthesized skin texture, illustrating the complexity of the computed and captured light transport). Please note how well local landmarks such as moles or wrinkles are preserved using non-local image reconstruction. Figure 10.5 also shows that naive application of non-local means filtering is not able to remove all noise, even in a configuration where most details are already blurred. Additionally, the figure shows two sets of neighborhoods that have been automatically chosen by the algorithm, indicating that the algorithm chooses sensible locations for the neighboring pixels.

A closeup is given in Figure 10.6, where different challenging com-


(a) 4 spp (b) 2 spp (c) 1 spp

(d) 1024 spp (e) NL 2 spp (f) NL 1 spp

(g) NL-means (h) Neighbors (i) Neighbors

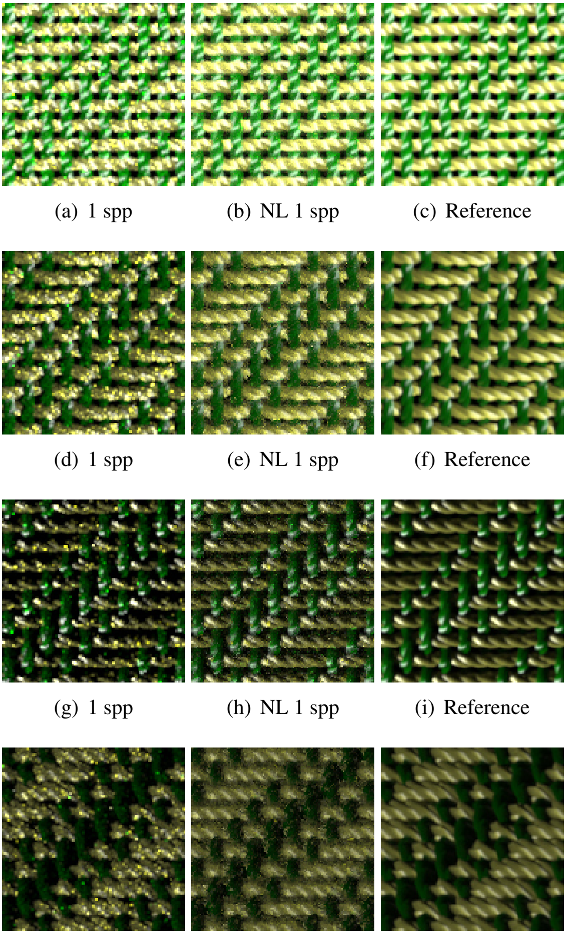
Figure 10.5: *Comparing a single spatial BTF slice synthesized using our approach with 48 neighbors per pixel (e-f) against unbiased path tracing (a-c) and reference image (d) for a complex skin material including volumetric subsurface scattering. Our result is already visually pleasing with 1 spp and looks very similar to a converged solution with 2 spp. (h) and (i) show two sets of neighbors (white pixels), belonging to the green cross in each image. (g) shows the result of the application of the classical non-local means image filtering algorithm to the tonemapped 1 spp image. All images in this figure show spatial slices that were not included in the set of slices, used for estimating neighborhoods.*

binations of viewing and lighting directions for a piece of cloth with highly specular fibers are shown. All fibers use a bidirectional curve scattering distribution function (BCSDF) decribed in [ZW07a] as material. The BTF slices illustrate the effects of different issues such as self-shadowing, highlights and self occlusion effects. Differences in color between different rows are largely the result of the cosine weighting based on the angle between surface normal and light direction.

For the top view in the first row of the figure, NL-reconstruction produces a result close to the reference even with only 1 spp. At 1 spp, the standard approach of path tracing with a Gaussian reconstruction filter cannot reproduce the shape of highlights of yellow yarns in the second row of the figure. They are reproduced well using our approach while only having a small amount of noise left. The reproduction of shadows and highlights is still acceptable even for grazing light angles θ_l in row three — however, the image starts to loose some contrast.

Self occlusion and shadows are not captured perfectly by NL-reconstruction with the chosen set of N-slices for the grazing angles in the last row; however the overall appearance is still relatively closely matching the reference. Please note that — when rendering with BTF — this extreme view projects to a relatively small area on screen.

Intuitively, increasing the number of neighbors used for non-local filtering should behave similarly to increasing the sampling rate. In reality, the convergence behaviour strongly depends on the quality of the neighborhoods. For an example we refer to Figure 10.7. Here, as the neighborhood grows, more details get recovered. However, at some point blurring is introduced by neighbors that do not match close enough. The same material is also depicted in Figure 10.11, showing again that materials containing only statistical repetitions and no exactly repeating pattern can be handled. Figure 10.9 shows a comparison for synthesized BTFs under image-based lighting – hardly any difference is visible between our result and a reference.



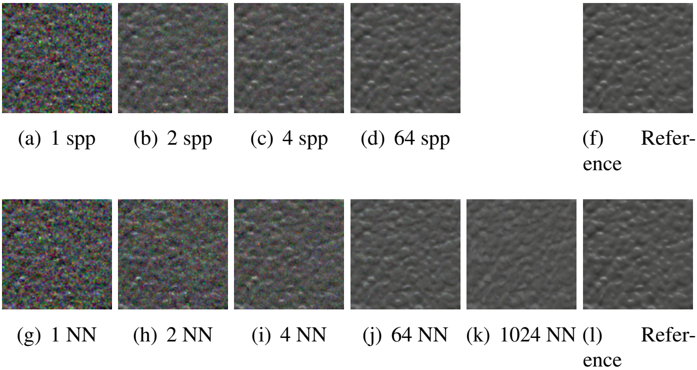
(a) 1 spp (b) NL 1 spp (c) Reference

(d) 1 spp (e) NL 1 spp (f) Reference

(g) 1 spp (h) NL 1 spp (i) Reference

(j) 1 spp (k) NL 1 spp (l) Reference

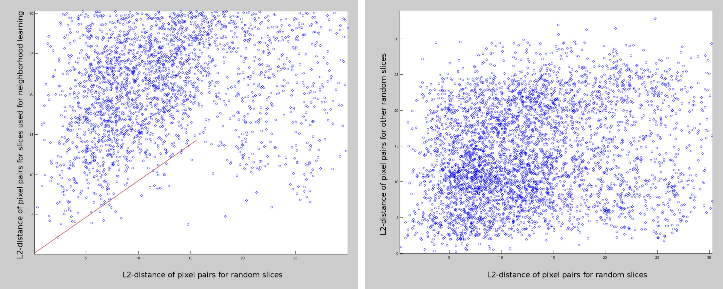
Figure 10.6: *Different challenging slices for a piece of cloth with highly specular fibers.* ***1st row****: viewing direction with inclination* $\theta_v = 0°$*, azimuth* $\phi_v = 0°$*, and light directions* $\theta_l = 0°, \phi_l = 0°$. ***2nd row****:* $\theta_v = 45°, \phi_v = 100°, \theta_l = 30°, \phi_l = 0°$. ***3rd row****:* $\theta_v = 30°, \phi_v = 90°, \theta_l = 75°, \phi_l = 240°$. ***4th row****:* $\theta_v = 75°, \phi_v = 315°, \theta_l = 75°, \phi_l = 240°$.



(a) 1 spp (b) 2 spp (c) 4 spp (d) 64 spp (f) Reference

(g) 1 NN (h) 2 NN (i) 4 NN (j) 64 NN (k) 1024 NN (l) Reference

Figure 10.7: *Comparison of the convergence for an increasing amount of samples per pixel without non-local reconstruction (upper row) and for an increasing number of neighbors (NN) while holding the sampling rate constant at only 1 spp (lower row). The example shows synthetic plastic with subsurface scattering for a viewing direction of inclination* $\theta_v = 30°$*, azimuth* $\phi_v = 290.00°$ *and lighting direction* $\theta_l = 75°, \phi_l = 95°$*. 1 NN equals 1 spp, hence (a) and (g) are identical. Differences in computation time for (g) to (j) are low. (j) has been created only from noisy samples in (g) and neigborhoods.*



(a) (b)

Figure 10.8: *Correlating similarity of pixel distances measured between two sets of BTF slices. Pairs of pixels have been selected and their L_2 distances (stacking color channels of slices) have been plotted for a set of random slices for the example in Figure 10.6 against a) the set of slices used for learning the neighborhood and b) another set of random slices.*

Numerical Analysis. The graph in Figure 10.3 demonstrates that our approach is very close to convergence even after 1 spp for the example shown in Figure 10.5. The more neighbors are used, the better is the result for low sampling rates. However, one can also see that for higher sampling rates, the result with more neighbors saturates.

Generally, quality strongly depends on parameter settings (i.e. the size k of the neighborhood and σ controlling neighborhood weights). However, as indicated by Figure 10.3, there is a wide sweet spot with respect to k. As can be seen, the rate of convergence does improve only gradually after $k = 16$ and the bias introduced by our filtering approaches a nearly constant level. It should be noted, however, that this bias manifests commonly as smoothing. Similarly, fixed large values for σ for a given number of N-slices gave satisfying results for all of our examples.

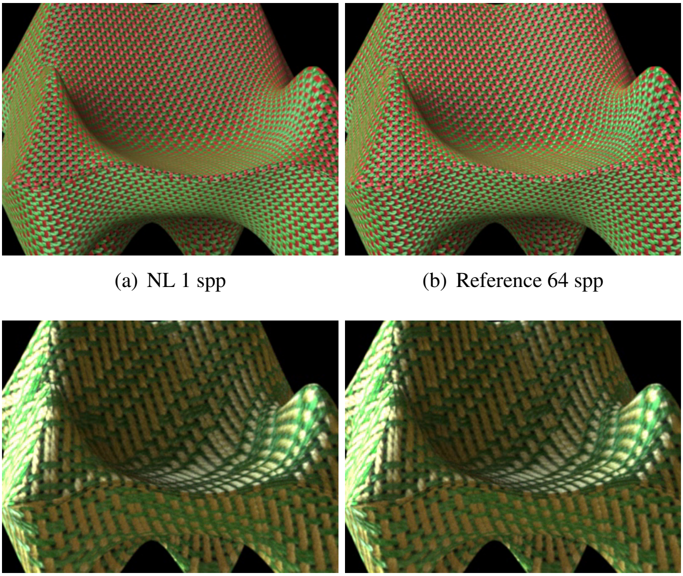
Essentially, by considering only a few N-slices, our assumption is that a correlation of ABRDFs in this subspace also induces correlation for the full BTF. To test the validity of this assumption, we carried out tests, relating normalized error with respect to the subspace to the error of randomly selected slices. A typical result is given in Figure 10.8 a. As can be seen, low distances for N-slices rarely result in high distances in the other slices. If, on the other hand, distances for N-slices become larger, the error in other slices increases as well. In a second test (Figure 10.8 b) we repeat the experiment, but with another set of N-slices not suitable for characterizing appearance well. In this case, as expected, no such correlation is visible, clearly indicating the importance of choosing N-slices right.

For materials with quasi-random structure, in particular in combination with substantial parallax, our method is likely to fail as our assumptions are no longer valid. For an example see Figure 10.12.

10.4 Limitations

The choice of the feature vector V_p is the most crucial step in our method. Failing to discriminate different ABRDFs, will most likely increase bias. For low sampling rates, this might be still acceptable if this results in a reduced variance of the Monte Carlo estimate.
BTFs can include completely different materials inside a single texel due to parallax effects. Especially for low camera viewing angles, the appearance of some texels may change drastically when compared to a view from above. For these texels, only very few good neighbors can be found globally. Depending on the value of σ in Equation 10.6, these texels either do not get denoised at all or they become blurred. Some blurriness may, however, be acceptable as texels are blurred anyway due to effects of camera projection for low viewing angles.

A major limitation of the current approach is that parallax effects reduce the amount of “good” neighbors one can find – however parallax effects are a general source of problems when handling BTFs also e.g. in the context of compression.



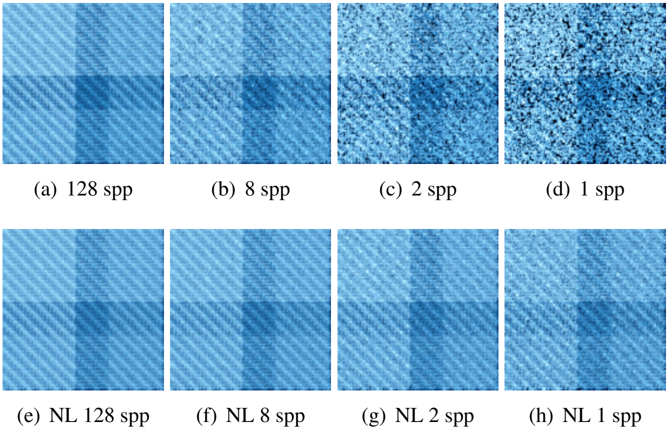
(a) NL 1 spp

(b) Reference 64 spp

(c) NL 1 spp

(d) Reference 64 spp

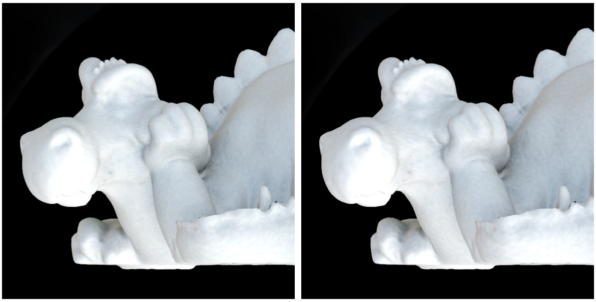
Figure 10.9: *Comparing our approach against a reference for two cloth BTFs using an image-based lighting setup.*



(a) 128 spp (b) 8 spp (c) 2 spp (d) 1 spp

(e) NL 128 spp (f) NL 8 spp (g) NL 2 spp (h) NL 1 spp

Figure 10.10: *The images show a synthetic piece of cloth in front of a black background. A comparison of a reference solution is given for a fixed viewing and lighting direction (a-d), to our solution (e-h) using non-local (NL) image reconstruction. Even using only 1 spp (h), we can achieve a reasonable quality – at 8 spp (f) hardly any difference is visible to the reference in (a). In (c) and (d) some rays have not hit any fiber; here the background shows through, resulting in black spots that the used Gaussian image reconstruction filter cannot remove (except for settings that result in excessive blurring).*



(a) NL 1 spp

(b) Reference 512 spp

Figure 10.11: *Renderings of full BTFs under image-based lighting showing the same material as the slices in Figure 10.7. Note that the color of the material is the result of a complex subsurface scattering process.*

10.5 Conclusion and Future Work

We have presented a method to efficiently synthesize BTFs from known micro-geometry. While our approach may leave room for improvement, we believe that its simplicity and effectiveness makes it attractive for practical application. However, further research is required to investigate the impact of user adjustable parameters, i.e. the size of a neighborhood vs. the number of samples per pixel. For results presented in this chapter, we have chosen σ large enough such that it has little effect. Because of this, choosing too many neighbors immediately leads to blurring. It would be interesting to further investigate the effects of different values of σ. We believe that information theoretic approaches could even render manual parameter tuning unnecessary. A next interesting step would be to apply the algorithm in the context of a real measurement device and see how one could lower measurement times by taking the photographs for different light directions with different exposure times.

In the future, we would like to extend the basic approach in several ways:
To reduce problems related to parallax effects, different solutions are thinkable. As long as a setup similar to a camera dome is simulated or if a real camera setup would be used, one can imagine to estimate warp fields compensating for shifting effects due to parallax. For purely virtual setups, however, other sampling schemes could completely avoid such artifacts.

We would also like to be able to control the bias caused by our technique. Moreover, we hope to make our method more effective by dynamically updating the neighborhood graph during the rendering process. Undersampling issues that may occur in case of small texel neighborhoods could be reduced by adapting the sampling rate locally. We could shoot more rays until a certain amount of virtual samples has been reached for all pixels, counting all samples that contribute to a certain pixel either directly or via neighbors.

In addition to adapting the sampling rate spatially, one could adapt the sampling rate angularly. It would be beneficial to have a finer angular sampling around highlights, to capture these faithfully. Combined with scattered data interpolation this could lead to efficient sampling schemes. Evaluating such BTFs during rendering would be more complex, though.

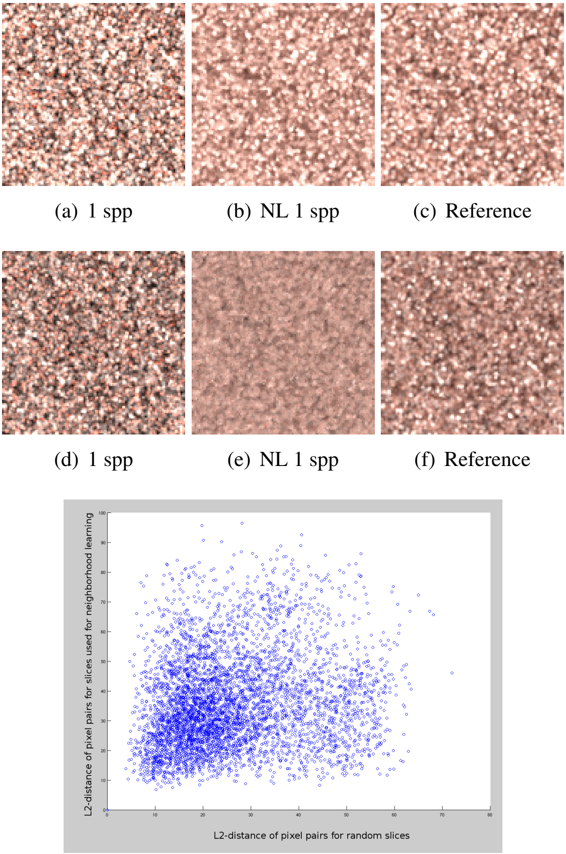
In our current system, the nearest neighbor graph is based on simple color similarities. In the future, more advanced sampling and similarity search algorithms in spatio-angular domain (6D) could be explored. Even more redundant information could be exploited, however, at the cost of a more complex algorithm.

For now, we have only handled repeating structures in the special context of BTF generation. However, we believe that using image-internal statistical information about repeating structures is a powerful way to reduce variance of Monte Carlo renderings in general. Especially for progressive rendering of scenes, early images could gain much perceived quality.

Finally, our method could be used for pre-filtering of surface shading in the presence of small scale detail [BN12].

10.6 Recently Published Related Work

In this section we summarize recent advances in research related to this chapter that have been made after the publication of our works [SMKZ11, SKZ13]. Rousselle et al. [RKZ12] propose a method to use non-local means filtering for accelerating adaptive rendering in general. To find neighborhoods, they first separate samples into two different buffers. They find neighborhoods in one buffer and use these to filter pixels in the other buffer using non-local means. They iterate this process several times, while switching the roles of both buffers after each iteration. While this approach works well and could also be explored in the context of BTF synthesis, performing non-local means several times on every image also comes with high computational cost and it would have to be evaluated if this strategy is effective, when compared to our approach. Rousselle et al. extend their approach [RMZ13] to combine color and feature buffers, based on e.g. textures or normals, using a SURE-based error estimate.



(a) 1 spp (b) NL 1 spp (c) Reference

(d) 1 spp (e) NL 1 spp (f) Reference

(g)

Figure 10.12: *Failure case example with randomly oriented particles (sand).* ***(a-f):*** *Each row shows a different BTF slice comparing our method rendered with 1 spp to both a reference and the standard approach. Due to the quasi-random characteristic of this material our results do not closely match the reference, in particular for grazing angles (2nd row).* ***(g):*** *Correlating error along the lines of Fig. 10.8.*

CHAPTER 11

VOLUMETRIC CLOTH RENDERING

While rendering cloth using synthetic BTFs, as discussed in the previous chapter, can be quite efficient, the approach also has drawbacks. Due to high memory requirements, BTFs can currently still only capture relatively small repeating patterns. Furthermore, BTFs also have

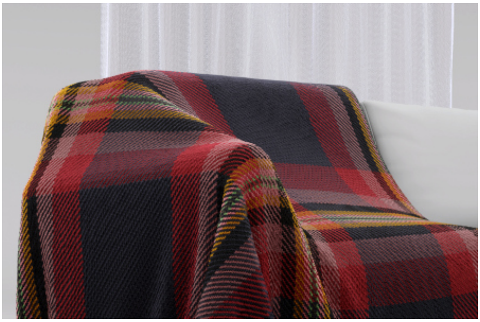


Figure 11.1: *Closeup view of Figure 1.1 with different lighting. A fiber-based model representing each yarn geometrically would become prohibitively costly as billions of line primitives are required to represent this model explicitly. Instead of needing terra-bytes of memory as required by an explicit representation, our proposed volumetric data structure easily fits into the RAM of current desktop hardware.*

several limitations by definition: Light diffusion is not modeled properly, especially at shadow boundaries; regions of high curvature are not represented correctly if the BTF is measured with regard to a flat sample and silhouette information is missing. Handling of transparency is difficult and rarely considered. Moreover, when creating synthetic BTFs, the interactive design process can be only performed based on the small sample that can be represented explicitly. However, to get a better impression of a cloth material, it is desirable to visualize it on a larger object already during the design phase. In this chapter, we describe a volumetric approach to cloth rendering, based on our work [SKZ11], that can be used for this task.

To overcome problems of exactly modeling the underlying microscale geometry, different solutions have been suggested, among them volumetric methods. We refer to Chapter 2.1.2 for an overview. In the context of hair rendering, Moon et al. [MWM08] have developed a hybrid approach to speed up multiple scattering computations. Instead of using the common volumetric rendering approach of using phase functions, they approximate geometry volumetrically using a statistical description and use the BCSDF of the individual fibers to model optical material properties. The hair geometry is first voxelized in a preprocessing step and relevant fiber properties are stored in a uniform grid data structure. These are the scattering coefficient, the average tangent of hair strands intersecting a voxel and the standard deviation of hair orientations with respect to this tangent. They are used in a subsequent light tracing phase to approximate multiply scattered light by spherical harmonics. This statistical model is only used to accelerate the computation of multiple scattering while for single scattering the complete micro-geometry is still needed. Therefore, the geometrical complexity remains high.

Since micro-geometry of cloth is much more complex than the one of human hair, the simple statistical model introduced by Moon et al. is not sufficient for predictive cloth rendering. The tangent direction of spun fibers varies significantly over small distances and nearby yarns may have completely different material parameters. While it is reasonable to assume that the distribution of fiber directions is homogeneous inside a voxel, when modeling hair and can be described by a normal distribution, this does not hold for cloth. Additionally, due to the homogeneous structure of hair, a loose approximation of the fiber density already leads to pleasant results. For cloth, however, this density has to be estimated much more accurately as fibers may be highly curved even inside a single voxel. A major challenge is to still be able to predict the correct color of yarns caused by multiple-scattering of light between fibers without representing these fibers explicitly.

Nevertheless, the idea of statistically modeling the micro-geometry while keeping the material properties of the individual fibers is very appealing. It first offers the possibility to use optical material properties of individual fibers, which are either provided by manufacturers or can be measured efficiently [ZLHA$^+$09], and second it delivers a possibility to deal with the inherent complexity of the micro-geometry of cloth. Therefore, we have incorporated this general idea into our approach. We have developed a statistical model for the micro-geometry of cloth that is used for both single and multiple scattering. Our main contributions in the context of volumetric cloth rendering are:

- the use of a Gaussian mixture model to model the fiber distribution in cloth.
- the simulation of single as well as multiple scattering using the statistical model.
- the introduction of a novel approach to compute a so called *effective fiber density*, allowing for highly accurate renderings.
- a solution to a general shadowing problem, inherent to any discrete volumetric representation rendered with path tracing, introducing the concept of *local visibility* for self-shadowing and modeling it as a new material parameter – the *Bidirectional Visibility Distribution Function* (BVDF).

Using our new approach, we can handle large pieces of cloth with orders of magnitude more fibers than methods representing fibers geometrically. Besides drastically reducing the memory requirements when comparing to an explicit representation, the rendering times decrease by a factor. We are able to render pieces of cloth with a visual quality comparable to fiber-based reference solutions while employing a completely generic method for modeling fibers without limitations. Our approach allows us to use state-of-the-art fiber scattering models that are essential to obtain accurate results. As our approach relies on virtual scattering events that account for multiple scattering effects inside a voxel, and because local shadowing is modeled separately by the BVDF, we can handle high frequency detail in both multiple scattering and shadowing. To demonstrate the effectiveness of our method, we compare renderings of the statistical model against ground truth reference solutions. For a discussion of related work in the context of cloth and fiber rendering, we refer to Chapter 2.

11.1 Overview

Figure 11.2 illustrates our pipeline for volumetric cloth rendering. The proposed method takes its advantage mainly from avoiding an explicit representation of individual yarn fibers, using less extensive volumetric statistical models.

To this end, the original cloth model is first voxelized. In a subsequent step, this volumetric representation is then rendered using Monte Carlo path tracing.

During voxelization, local statistics, modeling the distribution of yarns fibers, are stored in an octree, also serving as an efficient acceleration data structure for subsequent rendering. For each octree cell, we store information about the directional distribution of yarn fibers as a Gaussian mixture model. Each component of the mixture model represents the main direction and standard deviation of fibers of a single yarn inside a cell and is attributed with information about local fiber density and a material index. Moreover, the Bidirectional Visibility Distribution Function (BVDF), a property characterizing

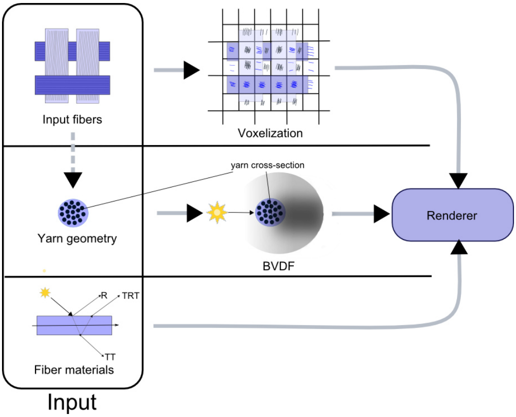


Figure 11.2: *Illustrating the pipeline, progressing from the input data on the left, over computation of voxelization and BVDF to the renderer on the right.*

the local visibility inside voxels near the surface, is precomputed. When rendering using Monte Carlo path tracing, *virtual scattering events* are generated. By sampling the statistical model, we obtain the positions of ray/cloth intersections as random variables. At each virtual intersection, a fiber with an associated material is hypothesized. Scattering at a fiber level is fully described by a BCSDF.
The whole process can be seen as sampling a voxels phase function, which stays implicit, “on the fly”. Shadowing artifacts in case of direct illumination are avoided by employing information captured by the BVDF.

11.2 Statistical Volumetric Modeling of Cloth

For comparison purposes, we first generate the cloth geometry explicitly in a representation our fiber based path tracing reference implementation can render. Then we directly infer the volumetric representation from this geometry. For each voxel cell, we describe the distribution of fibers using a statistical model.

11.2.1 Input Data

In theory, our volumetric representation for a piece of cloth can be generated from different kinds of input data. One can either use the procedural cloth model we have presented in Chapter 8, but one could also estimate volumetric statistics based on Micro CT data. The only requirements are that the distribution of fibers can somehow be measured inside a voxel and that local visibility information can be estimated (see Section 11.2.3).

11.2.2 Voxelization

The explicit cloth model is voxelized using an octree data structure that is also used for rendering. The tree is generated top down by propagating cylinder segments intersecting a voxel to its children. Finally, for each leaf voxel cell containing cloth, a Gaussian mixture model representing local fiber distribution is created according to Section 11.2.3. In our current implementation, the maximum octree-level is set manually and all leaf voxels have that same level. For timings regarding the construction we refer to Table 11.1. As can be seen, voxelization is extremely fast and does not constitute a bottleneck when compared to the generation of acceleration data structures for explicit geometry. Construction time could be greatly reduced if we would directly generate statistics from the procedural cloth model, instead of first constructing the geometry explicitly. However, in this work, we are interested in the explicit geometry as a reference anyway.

11.2.3 The Statistical Model

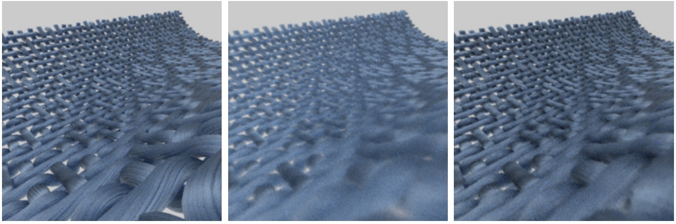
Given a set of n yarns, each consisting of several hundreds of fibers, we compute the statistical volumetric representation as follows: For the i-th yarn intersecting a voxel cell V, a tuple g_i representing Gaussian directionality, density and material properties is created:

$$g_i = \{\mathcal{N}_i(\boldsymbol{m}_i, s_i^2), \rho^i, \text{BCSDF}_i\}. \tag{11.1}$$

Besides the average tangent direction $\boldsymbol{m}_i$ and its associated standard deviation s_i, forming a Gaussian $\mathcal{N}_i(\boldsymbol{m}_i, s_i^2)$, a material index BCSDF$_i$ is stored. In addition, the mean free path length inside a voxel, represented by the *effective fiber density* ρ^i, is required to model fibers statistically. All $\mathcal{N}_i$ of a voxel cell, constitute components of a Gaussian mixture model. For the sake of simplicity, we will in the following refer to the whole tuple g_i as a component of a mixture model $G_V = \{g_1, ..., g_n\}$ where n equals the number of yarns inside voxel V. To minimize memory requirements, a compact representation was chosen to encode a Gaussian yarn model: Each mixture component is represented by as few as 13 bytes (16 bits for each of the 4 angles required for tangent direction and normal, 20 bits for the weight, 12 bits for sigma, 1 byte for a material index).

Fiber Density: The mean free path length is calculated based on the perpendicular attenuation coefficient $\sigma^\perp = 2r_f\rho$ with fiber radius r_f. Existing approaches developed for hair rendering that attempt to estimate the density ρ by ad-hoc methods, being based on counting fibers in a sphere volume, enclosing a voxel, are not suitable for cloth. As the length of lines covered inside a voxel is not properly taken into account, they fall short in case of highly curved fibers, as for spun yarn and in case of fibers that barely intersect the volume of interest.

In the following, we will assume that a set of n line segments $\{L_j | j = [1..n]\}$ is used to represent the cloth. To avoid the limitations of the



(a) Reference (b) Filtered fiber counting (c) Our method

Figure 11.3: *Comparison of the effects of our effective fiber density to the fiber counting method presented in Moon et al. [MWM08]. Note that this is a very challenging setup, showing a combination of directional and isotropic illumination with light falling through the cloth. Note that this example only compares different ways to obtain the required information for the effective scattering coefficient – rendering is perfomed using the exact same approach. In this setup our method not only shows less blurring but can also reproduce the color, resulting from multiple scattering of light, much better. Note the light blue in image (b) compared to the dark blue in images (a) and (c).*

fiber counting approach we introduce the concept of *effective fiber density* ρ.

Let's consider a spherical volume, similar to the search distance used by [MWM08], for density estimation. Then the average length of a line intersecting a sphere with radius r equals $\bar{l}_R = \bar{l}_{sphere} = \frac{4 \cdot r}{3}$. The projected area of the sphere along the tangent direction equals $A_R = A_{sphere} = \pi\, r^2$. Based on these properties, the effective fiber density for a spherical region then writes as

$$\rho^{sphere} = \frac{l_{total}}{\bar{l}_R\, A_R} = N/A_R \tag{11.2}$$

with $l_{total} = \sum_{j=1..n} l_j$ denoting the total summed up intersection

length l_j of all line segments with the sphere.
Intuitively, this can be seen as computing an effective fiber count N, the number of infinite spatially uniformly distributed fibers intersecting R, which would yield exactly the same density.

Of course the concept of effective fiber density may be generalized to arbitrarily shaped regions. As the cloth model is discretized by voxels, undesirable spatial blurring occurs if spheres were used to estimate ρ. This blurring not only smooths out details but also affects the amount of multiple scattered light, relevant for the appearance of cloth.
However, if using voxels, the angular variation of $\bar{l}_R$ and A_R need to be considered. The effective fiber density can be computed by summing over all line segments intersecting a given voxel:

$$\rho^{voxel} = \sum_j \frac{l_j}{\bar{l}_j(\boldsymbol{t}_j)\, A_{voxel}(\boldsymbol{t}_j)}. \tag{11.3}$$

Here, $A_{voxel}(\boldsymbol{t}_j)$ denotes the area of a voxel projected onto a plane perpendicular to the tangent direction $\boldsymbol{t}_j$ of a given line segment and $\bar{l}_j$ is the expectation of intersection length (the average) for $\boldsymbol{t}_j$.
To speed up computations, tabulated values of $A_{voxel}(\boldsymbol{t}_j)$ and $\bar{l}_j$ are used during voxelization. In Figure 11.3, a comparison between our method and the method proposed by Moon et al. [MWM08] is given for a challenging closeup example. An alternative to dividing by $\bar{l}_j(\boldsymbol{t}_j)$ would be to divide by the intersection length of a line obtained by extending the segment l_j. While the computational cost would be higher, as this length cannot be precomputed, the resulting density would be more exact.

The Effective Scattering Coefficient: As very briefly discussed by Moon et al. [MWM08], the effective scattering coefficient not only depends on fiber density but also on tangent direction t_j. For a single fiber indexed by j, the effective scattering coefficient σ^j_{single} with respect to a given direction $\boldsymbol{d}$ computes as:

$$\sigma^j_{single}(\alpha_j) = 2r_f\, \rho^j_{single}\, \sin(\alpha_j) \tag{11.4}$$

where ρ^j_{single} means the fiber density induced by the (single) fiber of radius r_f and $\alpha_j = \angle(\boldsymbol{d}, \boldsymbol{t}_j)$.

Now let g_i be the i-th Gaussian mixture component, representing a bundle of n_i fibers. Then the total scattering coefficient σ^i for the Gaussian $\mathcal{N}_i$ may be computed by summing over the contributions of the individual fibers:

$$\sigma^i = \sum_{j=1..n_i} \sigma^j_{single}(\alpha_j). \tag{11.5}$$

Finally, assuming $\mathcal{N}_i$ with average tangent direction $\boldsymbol{m}_i$ and a standard deviation s_i, σ^i may be approximated based on the fiber density ρ^i:

$$\sigma^i(\beta_i, s_i) \quad \approx \quad 2\, r_f\, \rho^i \int \sin(\alpha)\, \mathcal{N}_i(\boldsymbol{m}_i, s_i^2)(\alpha)\, d\alpha$$

where $\beta_i = \angle(\boldsymbol{d}, \boldsymbol{m}_i)$. To avoid computationally costly integration during rendering, we follow Moon et al. [MWM08] and precompute the integral. The time required for such pre-computation is negligible.

11.3 Monte Carlo Path Tracing with Virtual Scattering Events

We take a Monte Carlo path tracing approach to render the voxelized cloth. In contrast to participating media or highly scattering materials, such as skin, path tracing is effective in our case as, due to absorption inside the fibers, energy quickly decays to zero after a few scattering events.

Because of the volumetric representation of geometry, discrete scattering events along ray paths need to be synthesized stochastically based on the statistical information stored in the octree.

Let $\boldsymbol{x}$ be the position of the last vertex of a ray path and $\boldsymbol{d}$ denote the direction of the associated outgoing ray R. Then the next vertex of the light path is computed based on virtual scattering events by the following steps (see Figure 11.4 Left):

- **Step 1:** Compute the voxel V that includes $\boldsymbol{x}$.
- **Step 2:** Compute the position of a new potential virtual scattering event along $\boldsymbol{d}$ according to the total scattering coefficient (related to mean free path length) associated to V. This virtual scattering event is rejected if it lies outside the boundaries of V. In this case no scattering occurs and R advances to the next intersecting voxel.
- **Step 3:** Otherwise, a Gaussian mixture component of V is chosen and the ray gets scattered and attenuated according to the BCSDF.

11.3.1 Virtual Scattering Events

Once the current voxel V, including the last scattering event, has been computed, a new virtual scattering event needs to be created based on the Gaussian mixture components $G = \{g_1, .., g_n\}$ associated to V. Let δ_V be the distance at which the associated outgoing ray R is leaving (intersecting the boundary of) V. Then the probability p for being scattered inside V is given by Beer-Lambert's law as:

$$p(\delta_V) = 1 - T(\delta_V) = 1 - e^{\delta_V \sigma_s}, \tag{11.6}$$

where T is the transmittance and σ_s is denoting the total scattering coefficient of V. To obtain the combined scattering probability for all mixture components, σ_s is computed by summing over all scattering coefficients: $\sigma_s = \sum_i \sigma_s^i$. Based on the above, the virtual scattering event is stochastically computed using four uniformly distributed random numbers $\xi_{1..4} \in (0, 1]$.
First, the distance δ of the virtual scattering event along the ray R is

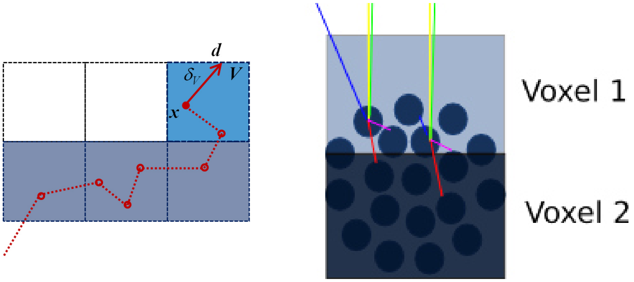


Figure 11.4: ***Left:*** *Path tracing with virtual scattering events. The last virtual scattering event was created in voxel V at position $\boldsymbol{x}$. The scattered ray is leaving $\boldsymbol{x}$ to direction $\mathbf{d}$. The probability of being scattered in V along the ray is computed based on δ_V and total scattering coefficient σ_s.* ***Right:*** *Illustrating the shadowing issues when representing a yarn's cross section volumetrically – each circle represents a single fiber. Yellow lines represent eye-rays – for each eye ray, 4 exemplary configurations of light-rays are shown. Green light-rays are never blocked, blue ones occasionally – pink and red ones illustrate cases of light coming from behind the cloth – the former are blocked locally in Voxel 1, whereas the latter are only blocked globally in Voxel 2 and are, therefore, regarded as not being blocked for the BVDF. In the volumetric case, all light-rays have a probability > 0 of being blocked inside voxel 1. To approximate shadowing correctly, these directional effect are modeled by the BVDF. Note that the voxels shown have twice the size (per dimension) of the voxels used for the statistical model (see Section 11.3.2).*

found by inverting transmittance T:

$$\delta = -\frac{\log(\xi_1)}{\sigma_s}. \tag{11.7}$$

If $\delta > \delta_V$ this scattering event is rejected, as it lies outside the voxel V.

Second, if scattering takes place, the new i^*-th Gaussian mixture component g_{i^*} is randomly selected with a probability proportional to its density:

$$i^* = \operatorname*{argmin}_{k \in \{1,..,n\}} (\xi_2 \leq \sum_{j=1}^{k} w_j) \tag{11.8}$$

with $w_j = \frac{\rho^j}{\sum_i \rho^i}$. Finally, based on ξ_3 and ξ_4, we choose a fiber direction determined by average tangent direction and standard deviation of g_{i^*} and eventually scatter according to the fiber scattering model BCSDF_{i^*}.

11.3.2 Direct Lighting and Self-Shadowing

Taking a volumetric approach to model cloth statistically, self-shadowing details below the scale of a single voxel cannot be captured.

In particular the directionality of local self-shadowing gets lost unless a prohibitively fine voxelization is employed. However, this effect is critical for the appearance, especially for direct illumination. Consider the following example: A head light is located at the same position as the camera – all points visible from the camera should also be lit by the light. As there is a non-vanishing probability for virtual scattering events along the shadow ray for any voxel containing fibers, shadowing is significantly overestimated. Moreover, as for any volumetric model, discontinuities at interfaces between optically dense and sparse regions are not well approximated (see Figure 11.4 Right). This is in particular true for densely woven cloth (the common case) that behaves optically very similar to a solid model with a well-defined

boundary.
To compensate for the above limitation of purely density-based representations, statistics for angularly dependent occlusion in case of direct illumination are used for approximating self-shadowing at a local level. Noting that these two kinds of bias are affecting shadowing within at least two adjacent voxels, statistics — we denote the *Bidirectional Visibility Distribution Function* — are computed at a resolution twice the desired size of a leaf voxel.

Bidirectional Visibility Distribution Function (BVDF): We propose to model the correlation between eye-rays from direction ω_o and shadow-rays into direction ω_i by introducing the concept of *local visibility*.
The fiber scattering equation for direct lighting is given by:

$$L_{o,d}(x,\omega_o) = \int_{S^2} f(x,\omega_i,\omega_o) L_d(x,\omega_i) V(x,\omega_i,\omega_o) sin(\alpha) d\omega_i$$

with BCSDF f, scattered radiance $L_{o,d}$, incident radiance distribution L_d due to direct lighting, visibility V and $\alpha = \angle(\omega_i, t)$ for fiber tangent t. V is split into a local part V_l, which accounts for shadowing due to occlusion inside a voxel cell and a global term V_g, which models occlusion outside that cell:

$$V(x,\omega_i,\omega_o) = V_l(x,\omega_i,\omega_o) V_g(x,\omega_i,\omega_o)$$

Generally, V_l is a spatially varying quantity. However, we will see that it can be reasonable to consider it as a material property of yarns. Assuming voxels with an extent in the order of the size of a yarn's cross section, local visibility is modeled by the *Bidirectional Visibility Distribution Function* V_{bvdf} given by averaging V_l inside a voxel cell:

$$V(x,\omega_i,\omega_o) = V_{bvdf}(\omega_i,\omega_o) V_g(x,\omega_i,\omega_o)$$

Note that V_l and V_g are binary, whereas $V_{bvdf} \in [0,1]$. In contrast to V_g, that is estimated using conventional shadow rays with virtual

scattering events (see Section 11.3.1), tabulated values are used for V_{bvdf}. To this end the average visibility inside a voxel is computed for multiple viewing and lighting direction pairs in a pre-processing pass: A small planar patch of cloth is generated, using the given yarn properties. We divide the space into cells of twice the size we plan to use for octree leaf voxels of the whole piece of cloth. The proportion of local shadowing is then computed using a fiber based rendering system for all cells hit by eye-rays. The resulting smooth function is stored by a 4D table (with $32 * 64^3$ bins in our case). Due to the small size of the sample, containing just a few yarns, this pre-computation is no bottleneck of our method and typically takes less than two minutes to compute for each yarn type.
To transfer the visibility information to a bend piece of cloth, we store inside each octree leaf cell a surface-normal of the underlying locally planar base mesh used for cloth synthesis. w_o and w_i are expressed in a local coordinate frame spun by this normal and the tangent direction of the last virtual scattering event. The general structure of BVDFs always looks similar: When light and camera position are next to each other (green rays in Figure 11.4), $V_{bvdf} \approx 1$, conversely when light and camera are on opposing sides of the cloth (black rays), V_{bvdf} is small. Moderate deviations in cross section shapes and densities of yarns only result in subtle changes. This observation is supported by the testing results summarized in Figure 11.5. Figure 11.6 shows several examples of BVDFs. Each small square shows the effect of varying direction θ_o of the eye-ray and direction θ_i of shadow-rays for a fixed combination of ϕ_o (always $= 0$ for the presented subset of the BVDF) and ϕ_i. When both light direction and viewing direction are parallel to the surface-normal pointing towards the sample ($\theta_o = \theta_i = 0$), no shadowing occurs as expected – conversely, when the light comes from behind the cloth ($\theta_i > \pi/2$) shadowing occurs.

The BVDF is used only in case of direct illumination to avoid distracting shadowing artifacts that otherwise would occur. Its effect is illustrated in Figure 11.7, comparing renderings with and without BVDF. For all other shadow-rays, calculated during multiple-

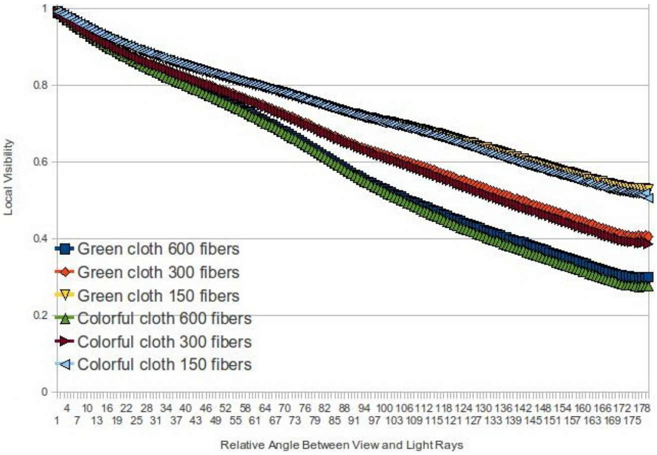


Figure 11.5: *Local visibility for two different cloth samples (~400K voxel examples / middle column of Figures 11.8 and 11.10) with varying numbers of fibers per yarn: For illustrational purposes only the relative angle between light and view direction is considered. Visibility almost only depends on yarn properties, independent of the weave pattern (at this scale). Despite the large amount of change in fiber density, visibility only changes moderately. Even for solid pieces of cloth, it will not reach* 0 *(fully occluded) for* 180°*, where light and view direction oppose each other, as a significant amount of occlusion is modeled by the global factor.*

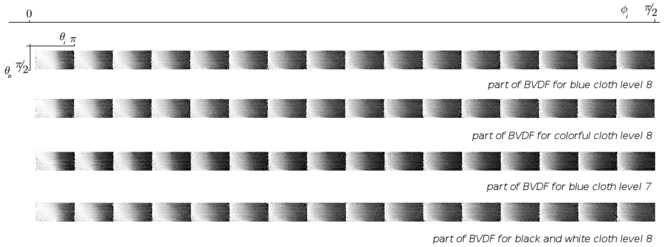
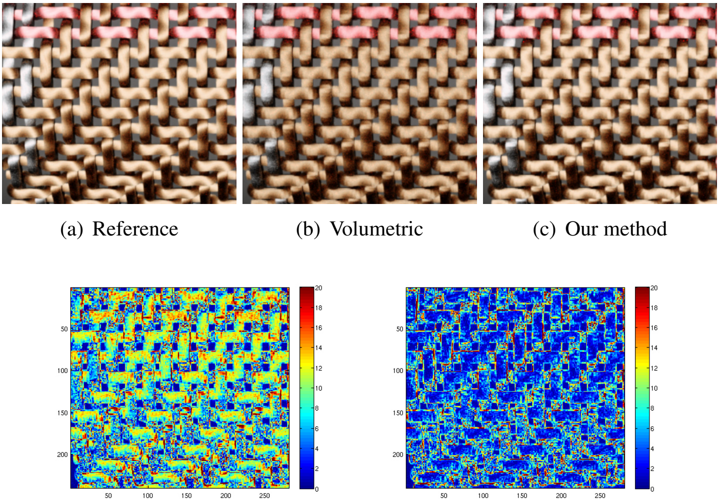


Figure 11.6: *Comparison of a subset of a BVDF with fixed azimuthal camera direction $\phi_o = 0$ for several cloth examples. White means no shadowing and black means that every shadow-ray shot into that direction hits a fiber. One can see that the BVDF is mainly a material property of the yarn. "Blue cloth" and "colorful cloth" have almost the same BVDF as they use the same yarn geometry, whereas "black and white cloth" with its fewer fibers is somewhat lighter (for low angles of ϕ_i) – although still very similar as the same spinning technique has been used. The BVDF for lower octree levels mainly shows more shadowing but is in its essence comparable to the next higher level.*



(a) Reference

(b) Volumetric

(c) Our method

(d) Difference: Volumetric

(e) Difference: Our method

Figure 11.7: *Closeup comparison of the reference solution (a) to naive handling of shadowing by using the volumetric information only (b) and to our method using the BVDF for local visibility estimation (c). The scene is lit using a single point light – the fibers in this image exert a significant amount of absorption and specular highlights dominate the appearance. Incorrect handling of shadowing results in darkening and an incorrect output color for the purely volumetric technique. (d) and (e) show Delta-E difference images for the volumetric technique and our method. Apart from general variance, our method shows most differences near yarn borders (caused by the discrete voxelization), whereas the purely volumetric method overestimates shadowing everywhere. Note that the method is not intended for such a closeup view.*

METHOD	MEM	IBL	POINT	CONST.
reference (20M lines)	10GB	07:32m	03:16m	7:20m
level 9 (2.5M voxels)	177MB	02:27m	01:19m	1:28m
level 8 (400k voxels)	30MB	02:18m	01:15m	0:54m
level 7 (75k voxels)	5.6MB	02:12m	01:12m	0:45m
level 6 (13k voxels)	1.1MB	02:09m	01:10m	0:36m

Table 11.1: *Comparing average rendering times for image based lighting (IBL) using an environment map and for a single point light located at the camera position and comparing memory consumption for different octree resolutions. Times for constructing the kD-tree and for voxelization are noted under the name "CONSTR.". All times were taken on a Core i7 CPU operating at 3.07 Ghz, rendering images of size 300x210 with 512 samples per pixel. Note that although the kD-tree implementation is generally very efficient, its construction process has not been parallelized.*

scattering, occlusion is computed based on virtual scattering events along the ray according to Section 11.3.1. Once rays have entered the cloth, the actual shadowing is less critical and may be computed using the statistical model. A valuable side effect of using the BVDF is that rendering times reduce since less costly shadow-rays need to be evaluated.

11.4 Results

We have tested our approach with synthetic cloth models with varying weave patterns (see Figures 11.8 to 11.12), different types of yarn geometry and realistic and at the same time challenging BCSDFs matching glossy dielectric fibers. To better identify potential weaknesses, cloth was coiled around a cylinder. For the same reason, we have selected examples with a very regular fiber structure: a single-ply coaxial helix model with a uniform fiber distribution within the cross-section with no fiber migration.

The geometric complexity of the models was taken such that a fiber based reference rendering could be created on a computer with 12GB RAM. Fibers of all presented cloth samples have a fixed index of refraction (1.5) and varying absorption coefficients using the BCSDF model of [ZW07a].

All images have been simulated with full global illumination based on Monte Carlo path tracing. As can be seen from the closeups in Figure 11.10, the colors of i.e. the red yarns are largely determined by the multiple scattering component – therefore, correct reproduction of this color is far from being trivial. For other yarns like the green ones with higher absorption values, multiple scattering is less prominent.

The results given in Figures 11.8 to 11.12 indicate that for the highest octree level, with 2.5 million voxels, the volumetric approach delivers results that are visually next to identical to the reference solution, while already needing much less memory. Levels 8 (~400k voxels) and 7 (~100k voxels) only show some (expected) blurring caused by the discretization, but the overall appearance is captured faithfully and consistently across different voxel resolutions. For level 7, the cross section of a yarn is covered by only slightly more than a single voxel. In this figure, voxels are projected to several pixels to make the artifacts introduced by the discretization visible – in practice one would choose a level where the voxel size is selected in such a way that voxels are projected to no more than (for example) a single pixel. At Level 6 (~10k voxels), the length of a voxel edge is slightly larger than the yarn diameter. Artifacts are introduced because the separation of yarns cannot be resolved correctly: for example yarns below another one show through and their distributions are mixed.
Although we have mainly included this level for illustrational purposes in the figure, even this coarse representation might sometimes be sufficient for images of distant clothes as the overall color impression is still reproduced. In addition to the cylinder images, we have simulated BRDFs for the reference technique as well as for our approximation based on a small patch of cloth. Two exemplary results for fixed lighting directions are illustrated in Figure 11.13: Reference and approximation match well, regardless of the viewing direction.

Note that the difference in memory and computational costs is even more dramatic for more complex dense models. Assuming a fixed spatial resolution, the costs for the novel method are bound by the number of voxel cells.,The explicit model, instead, requires memory at least linear in the size of the fiber primitives and costly acceleration data structures (such as a kD-tree) for rendering. For example for staple yarns, made from several plies, the memory savings could still be much higher. Please note that the comparison of rendering times may be biased in favor of the explicit technique: it already becomes completely unusable, due to excessive memory requirements, for geometries that can still be regarded as trivial for the volumetric case. It should also be noted that the octree has been optimized for size (i.e. no voxel coordinates are stored explicitly), whereas the kD-tree has been optimized for speed (and e.g. stores bounding box coordinates). Several examples of large pieces of cloth are presented in Figure 11.1. These are — in contrast to the volumetric method — prohibitively costly to render with the fiber based approach. The curtain in the background also serves as an example for a transparent piece of cloth.

11.5 Limitations

Due to the assumptions made, our approach has limitations when compared to methods that explicitly model individual yarn fibers. First of all, only effects on a scale bigger than the size of a voxel can be resolved properly. Thus, if the voxel's extent is not chosen appropriately, bias occurs, which causes spatial and angular blurring, as well as shadowing artifacts. Naturally, artifacts become more evident in case of extremely curved yarns, very directional lighting and for very specular fibers. However, it is important to recognize that these are the practical limitations of any approach not explicitly modeling fine scale geometry.

Our method is not effective in case of extreme closeups where individual yarn fibers become visible, as a prohibitively high spatial resolution would be required to resolve such fine geometric detail. The BVDF, as formulated above, only describes visibility for points hit by eye-rays – although this is not unreasonable, as mainly effects at the surface are described, a bias, which mostly results in darkening, is introduced in case of multiple scattering. Shadow-rays shot during multiple scattering that have an origin near the surface might profit from a BVDF representation as well.

11.6 Conclusion and Future Work

In this chapter, we have presented a physically-based alternative to fiber based rendering of fabrics. As for methods using an explicit representation of fiber geometry, our approach allows to simulate light scattering based on small scale optical properties of yarn fibers. By taking a statistical model, the memory as well as computational costs were greatly reduced. We are convinced that our approach will be valuable for applications such as virtual prototyping of cloth, where a high degree of realism is desirable but fiber-based simulations are not practical.

We believe that there is still potential for increasing the efficiency of the method in several different ways: First of all, as some types of cloth exhibit a repetitive weave pattern, Gaussian mixture components could be computed only once per pattern and referenced accordingly. Moreover, radiance caching techniques could be applied, that first store average radiance values in the voxel and then attempt to decrease variance by combining results across similar repetitions of a weave pattern. Two straight forward extensions would be the use of an adaptive octree, which tries to keep yarns in separate voxels and at the same time approximates the silhouette to some specified degree, using as few voxels as possible. One could also utilize the hierarchical structure for level of detail by not only storing Gaussian mixtures in leaf voxels but also for lower levels. It would be interesting to investigate what other acceleration data-structures could be applied: The reference technique actually does not perform that much slower in our examples, although it has to perform intersection test calculations. These are a lot more costly in comparison to sampling our statistical model – this is mainly due to the efficiency of the kD-tree when compared to the octree.

Cloth can include other effects like dirt, brighteners or a piece of cloth could simply be wet. While these can be modeled to some degree by altering the BCSDF, future work could go into modeling these effects volumetrically as well.

Due to modeling constraints, most presented images contain repeating structures. However, this is no limitation of our volumetric representation, which can handle arbitrary fiber based input data. For some types of *fancy yarns*, a spatially varying BVDF could actually be required. As the BVDF varies smoothly with respect to yarn properties (see Figure 11.6), interpolation between a few basis BVDFs could be applied.

We have shown in Chapter 8 that we can model the appearance of real cloth samples well with fiber-based models. So far, we have demonstrated the effectiveness of our method only by comparing it to synthetic reference images. While we believe that this comparison is fair as it reflects the current state-of-the-art in the field, measuring real cloth samples under different viewing and lighting directions and using this information for further validation, will be an interesting topic of future research. One might also consider using the technique for other materials for which a similar statistical model of micro-geometry could be appropriate.

11.7 Recently Published Related Work

Recently, Zhao et al. [ZHRB13] have proposed an interesting approach, utilizing the fact that cloth commonly consists of repeating structures, to accelerate rendering. This shares some conceptual similarity with the approach presented in the previous chapter. Instead of making use of repeating structures in the image domain, as we do, they use repeating structures in the volume. Cloth is synthesized based on a number of exemplar blocks and light transport is pre-computed for these blocks, allowing for efficient rendering.

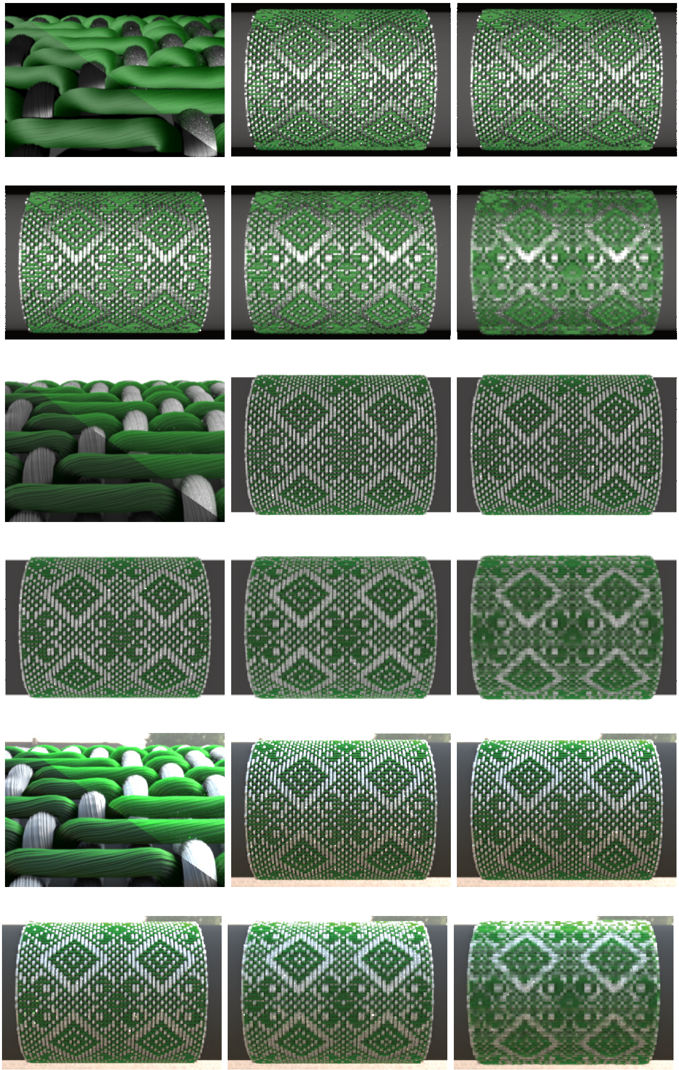


Figure 11.8: *Effect of changing octree resolution: The sample is lit using a point light from camera direction (1st two rows), a white dome light (middle) and a sunny outdoor environment (bottom). For each condition from upper left to lower right: closeup (upper right within image: multiple scattering, lower left: single scattering), reference (19M line segments), octree 2M Voxels, and below ~400K, ~80K, ~10K. For a better comparison, the image resolution is kept constant across different resolutions. Variance is comparable for reference images and the volumetric method.*

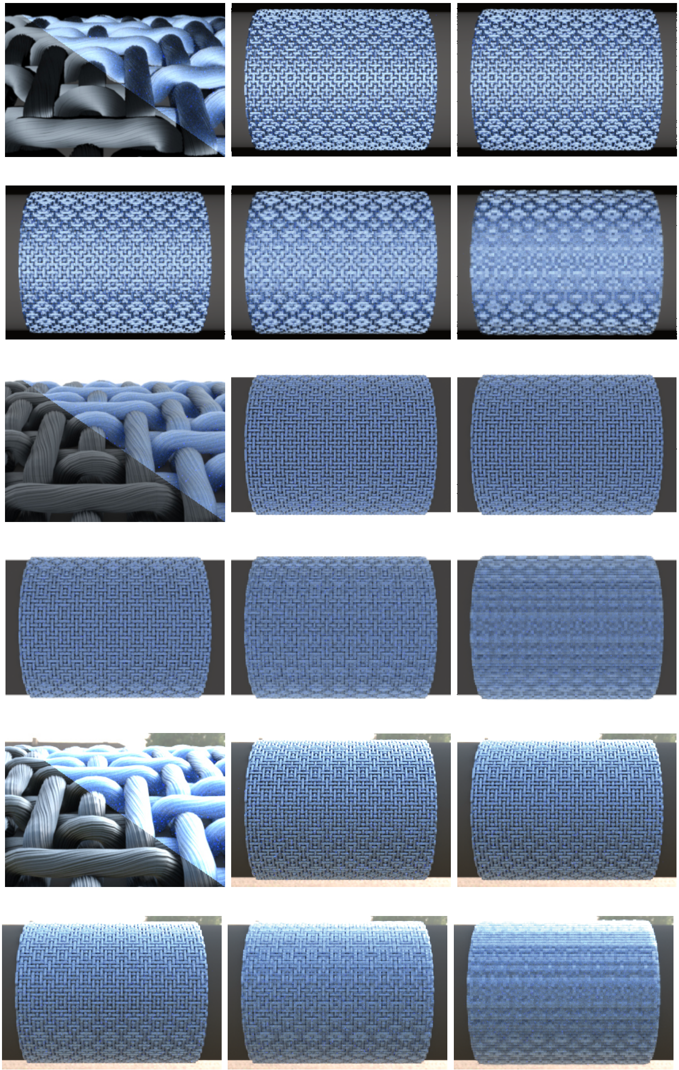


Figure 11.9: *Comparison for Blue Cloth along the lines of figure 11.8*

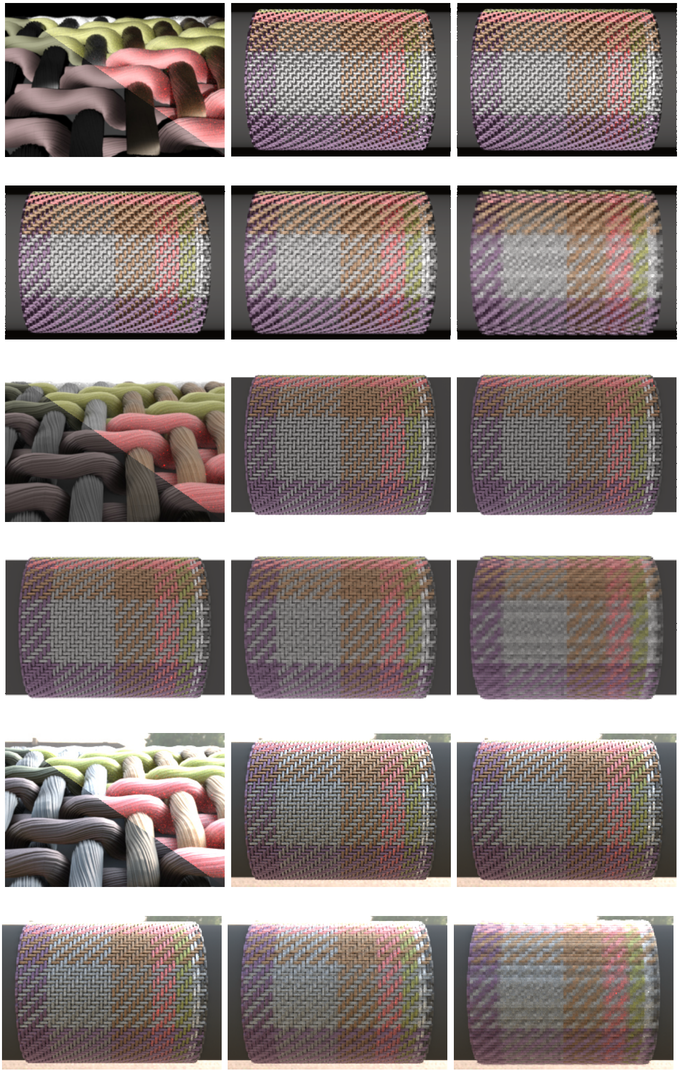


Figure 11.10: *Comparison for Colorful Cloth along the lines of figure 11.8*

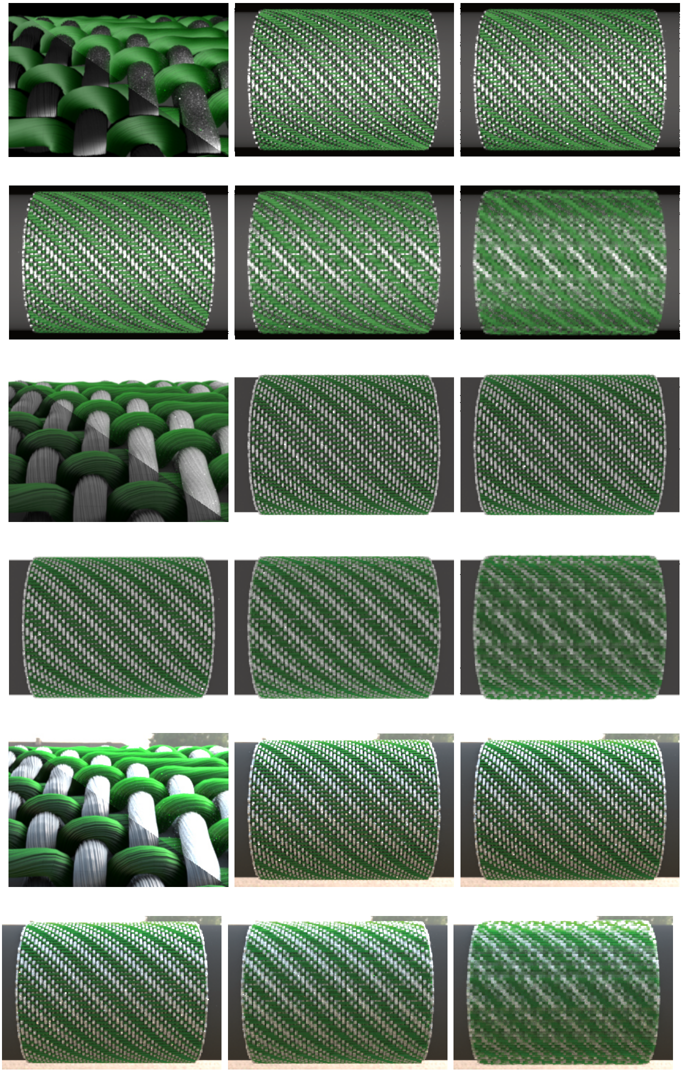


Figure 11.11: *Comparison for Twill Cloth along the lines of figure 11.8*

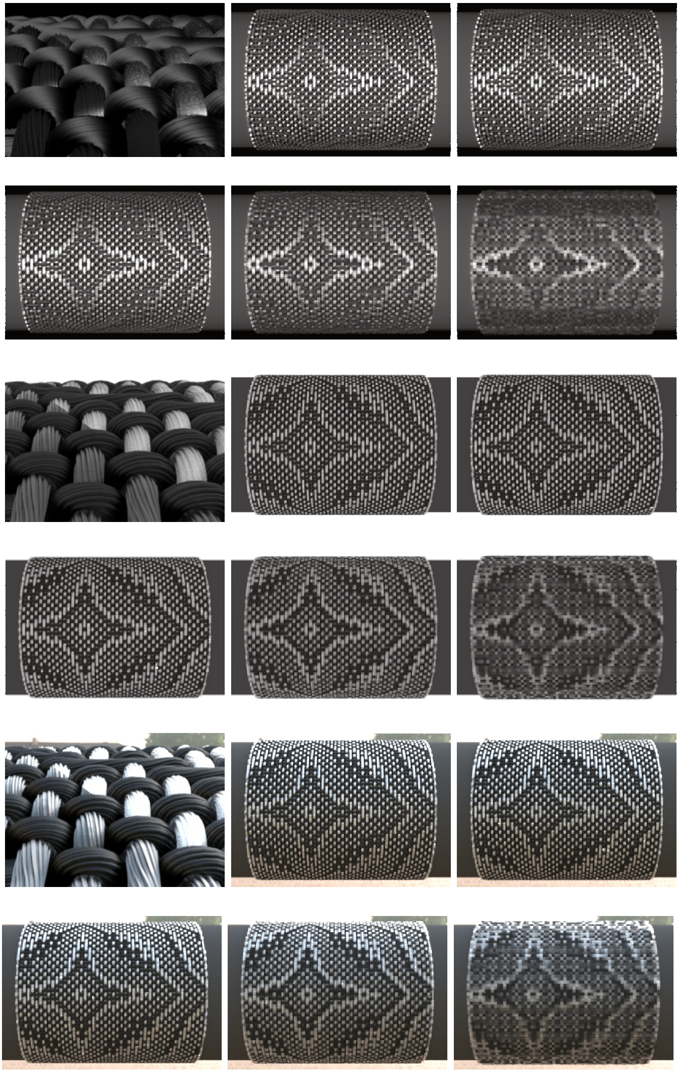


Figure 11.12: *Comparison for Black and White Cloth along the lines of figure 11.8*

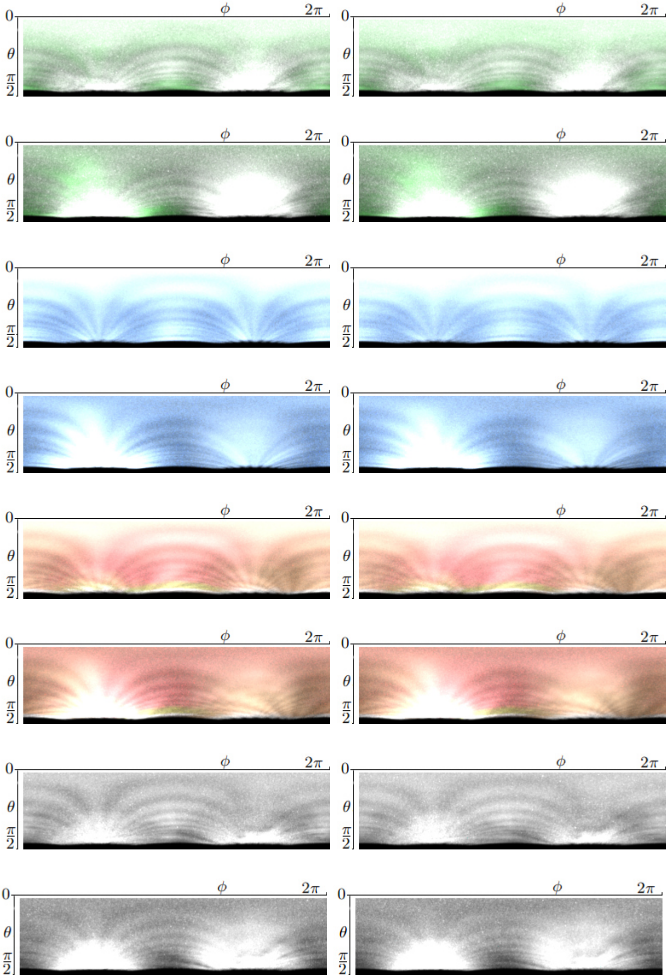


Figure 11.13: *Simulated reflectance fields (slices of the BRDF) of a quadratic piece of cloth, consisting of 30 warp and 30 weft yarns for two fixed lighting directions showing "green cloth", "blue cloth", "colorful cloth" and "black and white cloth". Direction is parametrized by spherical angles* $0 < \phi < 2\pi$ *and* $0 < \theta < \frac{\pi}{2}$. *Incident light direction is* $\phi = \theta = 0$ *for (a),(b),(e),(f) and* $\phi = \theta = \frac{\pi}{4}$ *for (c),(d),(g),(h). Reference (~7M line segments) on the left of each pair / Volumetric (~400K voxels) on the right.*

Part V

Closure

CHAPTER 12

CONCLUSION

We have presented a pipeline for visual prototyping of cloth consisting of three main aspects:

First, to be able to design a virtual cloth representation, we need an appropriate appearance model. For intuitive design it should contain the three major elements of cloth as editable parameters: optical properties of fibers, geometrical properties of yarns and compositional elements such as a weave pattern. We have presented a procedural fiber-based yarn model, incorporating principles of textile research, combined with a state-of-the-art fiber scattering model.

Second, to be able to model existing samples, methods are required to obtain parameters of the model. We have presented an approach to semi-automatically obtain a procedural cloth model using a simple image-based capturing setup. Our approach automatically finds the repeating pattern, regularizes non-rigid deformations and detects the paths of yarns, the weave pattern and variations of local deformations at the yarn level. Renderings of our model match the appearance of an input image.

Third, we have presented methods to efficiently compute predictive renderings of cloth based on these elements. The choice of the most efficient representation for rendering, in the context of computer aided design of cloth appearance, largely depends on the scale we are interested in and the editing flexibility we want to have:

A general concept used to efficiently represent the appearance of materials, we also employ within this thesis, is to describe structure at some scale explicitly while storing finer structure in another, probably aggregated, form.

The cloth model can be represented by a fiber assembly with associated BCSDFs, and Monte Carlo path tracing can be used to estimate the appearance. The basic shape of fibers can be described by curves. However, details of the internal structure of individual fibers and their exact shape are only represented indirectly by modeling the effects they have on the appearance as seen from a distance. This is a reasonable simplification for most applications as the geometric details of fibers used for cloth can only be seen under a microscope. During rendering, light transport within the fiber does not have to be computed explicitly, as it can be efficiently represented by a BCSDF. In the context of CAD, fibers are commonly selected from a given set of types. Therefore, it seems practical to create a database of fibers with associated BCSDFs. Combined with our procedural model, this allows for high flexibility. However, modeling all fibers explicitly does not scale, as memory requirements become prohibitive for cloth samples larger than a few squared centimeters.

Assuming a viewing distance at a yarn scale, we can e.g. use BTFs with a spatial resolution not resolving individual fibers. Similar to how the BCSDF aggregates light scattering within a fiber, the BTF additionally aggregates light scattering among fibers and between yarns. To use the benefits of BTFs in terms of efficient rendering, while still allowing for editing based on the elements of cloth, we have presented an approach to create synthetic bidirectional texture functions by means of path tracing. Cloth design is performed on a small explicitly modeled repeatable sample and a novel technique, we call non-local image reconstruction, has allowed us to perform a fast computation of a BTF showing the sample.

Finally, we have presented a method using a statistical volumetric representation of the fiber-geometry, allowing us to perform visual prototyping of even large samples, consisting of billions of fibers. Optical properties are directly linked to BCSDFs of fibers, while their arrangement is stored as a volumetric representation created using e.g. a fiber-based model. A comparison with reference renderings has shown that our approach is able to simulate the same color appearance, resulting from multiple scattering of light, we obtain for a direct rendering of an explicitly modeled fiber-based representation.

CHAPTER 13

FUTURE WORK

Our pipeline for visual prototyping of cloth could be improved in different ways. A potential direction of future research would be to make the estimation of parameters for the procedural model fully automatic. This could include the estimation of BCSDF parameters directly from the cloth input images. A technique along the line of Zinke et al. [ZLHA$^+$09] is thinkable. Also parameters such as fiber density and optical fiber parameters could be estimated in an *analysis by synthesis* framework. However, the task of estimating both geometrical and optical parameters at the same time will be challenging, as both of them affect the observed color.

So far, we have mainly concentrated on woven cloth models. It would be interesting to extend the pipeline to also support other types of fabrics. While our rendering methods are relatively independent of the specific type of fabric, our analysis uses the domain specific properties of single layer woven cloth.

The creation of synthetic BTFs is a generic method, which works fine for any material as long as a repeatable structure can be stored in a rectangular texture and the material can be well represented using a BTF.
Our volumetric rendering method can be used for any yarn-based cloth representation. For non-wovens, such as felt, where fibers are more randomly oriented, a different statistical model may be needed.

Many of the ideas presented in this work, related to an automatic analysis of cloth, could probably be adapted to the task of analyzing knitwear as well. Knitwear also contains translational symmetries and one could find regions of homogeneous stitch types and apply our bi-scale regularization framework.

Instead of using the symmetry-aware cross correlation operator one could apply a matching of templates that characterize the features of stitch types. This could then be used to estimate initial yarn locations, which could be further refined using an active yarn model.

An important future step is a further validation of our pipeline, when compared to real samples so that it can be used for predictive rendering. So far, we have only compared renderings of our procedural cloth model with the single input image directly. In the future, one should validate the results and estimate the appearance based on more view and light directions.

The employed fiber scattering model has been evaluated for hair and synthetic fibers [ZW07b]. Animal hair (wool) behaves reasonably similar. For certain fiber types, however, appearance cannot be effectively modeled without changing the BCSDF model.
BCSDFs are only accurate up to a certain curvature of fibers, especially in case of low absorption. Also the far field assumption is not fully accurate in case of multiple scattering for dense fiber assemblies. While major effects of forward and backward scattering, as well as locations of highlights, can be assumed to be modeled reasonably well, a further evaluation is needed. An interesting direction of future research would be to evaluate how well the proposed model can match the appearance of real yarns. To validate the accuracy of the fiber scattering model, one could for example track fibers for a Micro-CT measurement of a yarn. Then one could independently measure a BCSDF for fibers, for example using the method of Zinke et al. [ZLHA+09] and finally, compare the appearance of the real yarn with a simulated one.

Micro-CT scans could also be used to directly estimate parameters of the procedural yarn model. Additionally, one could think of adding physically-based simulation at the yarn and fiber level to aid the modeling and analysis process. Moreover, dyes, dirt or water could change the interface properties and therefore the optical appearance. To some extent, these effects can be modeled by adapting the properties of BCSDFs, however other, for example volumetric, representations could be investigated as well.

While we have concentrated on applications relevant to computer graphics, we believe that methods developed in this thesis have applications in textile research as well. The analysis procedure can be useful for automatic quality control, defect detection or reverse engineering of woven cloth.

A successful strategy used within this thesis has been to use domain specific knowledge about structures:
By combining information about translational symmetries with an effective way to handle non-rigid deformations, we were able to design a structure-aware regularization and filtering approach. We have used this to separate structures of yarns from structures at the fiber level. A similar approach could be useful for any kind of material with symmetrical parts.

By separating cloth into its basic elements, we were able to use different abstractions at different scales: BCSDFs for the optical properties of fibers, a procedural model as well as a volumetric statistical description for the geometry of yarns and finally a method to combine yarns using a description of the weave pattern. While we are not able to measure a full BSSRDF of a cloth material with current measurement setups, the properties of all individual elements can be measured in isolation. In combination with predictive rendering approaches, this allows us in principle to obtain a representation with the expressiveness of a BSSRDF, while being more practical.

Although further validation is needed, also in case of cloth, this general approach could be useful for other materials as well. One example for this could be sand, where we could model individual particles using a representation similar to a BCSDF, parameterized with respect to a sphere around the particle instead of a local cylinder, while modeling the distribution of particles in volumetric statistical ways.

List of Publications

[SKZ11a] Kai Schröder, Reinhard Klein, and Arno Zinke. A volumetric approach to physically-based rendering of fabrics. Technical Report CG-2011-1, Universität Bonn, January 2011.

[SKZ11b] Kai Schröder, Reinhard Klein, and Arno Zinke. A volumetric approach to predictive rendering of fabrics. *Computer Graphics Forum (Proceedings of EGSR 2011)*, 30(4):1277–1286, July 2011. 8, 124

[SKZ13] Kai Schröder, Reinhard Klein, and Arno Zinke. Non-local image reconstruction for efficient computation of synthetic bidirectional texture functions. In *Computer Graphics Forum (presented at EGSR 2013)*, volume 32, pages 61–71. Wiley Online Library, 2013. 8, 99, 120

[SMKZ11] Kai Schröder, David Möller, Reinhard Klein, and Arno Zinke. Non-local image reconstruction for efficient btf synthesis. In *ACM SIGGRAPH ASIA 2011: Sketches*. ACM, December 2011.

[SZ13] Kai Schröder and Arno Zinke. Interactive appearance design in the presence of optically complex materials. In *Eurographics Workshop on Material Appearance Modeling: Issues and Acquisition*, pages 13–16. Eurographics Association, June 2013.

[SZK13] Kai Schröder, Arno Zinke, and Reinhard Klein. Image-based reverse engineering and visual prototyping of woven cloth. In *Visualization and Computer Graphics, IEEE Transactions on*, volume 21(2), pages 188–200, February 2015. 8, 37, 74

[SZZ12] Kai Schröder, Shuang Zhao, and Arno Zinke. Recent advances in physically-based appearance modeling of cloth. In *ACM SIGGRAPH Asia 2012: Course Notes*. ACM, November 2012.

BIBLIOGRAPHY

[ABW94] Masaki Aono, David E Breen, and Michael J Wozny. Fitting a woven-cloth model to a curved surface: mapping algorithms. *Computer-Aided Design*, 26(4):278–292, 1994. 89

[ABW01] Masaki Aono, David E Breen, and Michael J Wozny. Modeling methods for the design of 3d broadcloth composite parts. *Computer-Aided Design*, 33(13):989–1007, 2001. 89

[ADBW96] Masaki Aono, Paolo Denti, David E Breen, and Michael J Wozny. Fitting a woven cloth model to a curved surface: dart insertion. *Computer Graphics and Applications, IEEE*, 16(5):60–70, 1996. 89

[AMTF03] N. Adabala, N. Magnenat-Thalmann, and G. Fei. Real-time visualization of woven textiles. In J. C. Guerri, Pajares A, and C. A. Palau, editors, *Publication of EUROSIS*, pages 502–508, 2003. 16, 17

[BC07] T. Brox and D. Cremers. On the statistical interpretation of the piecewise smooth mumford-shah functional. *Scale Space and Variational Methods in Computer Vision*, pages 203–213, 2007. 34, 65

[BCM05] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. *CVPR IEEE Conference*, pages 60–65, 2005. 49, 99, 101

[BEM11] Pablo Bauszat, Martin Eisemann, and Marcus Magnor. Guided image filtering for interactive high-quality global illumination. *Computer Graphics Forum (Proc. of Eurographics Symposium on Rendering (EGSR))*, 30(4):1361–1368, 2011. 100

[Bli77] James F Blinn. Models of light reflection for computer synthesized pictures. In *ACM SIGGRAPH Computer Graphics*, volume 11, pages 192–198. ACM, 1977. 12

[BN09] Y. Ben Salem and S. Nasri. Automatic recognition of woven fabrics based on texture and using SVM. *Signal, Image and Video Processing*, pages 1–6, 2009. 54

[BN12] Eric Bruneton and Fabrice Neyret. A survey of nonlinear prefiltering methods for efficient and accurate surface shading. *Visualization and Computer Graphics, IEEE Transactions on*, 18(2):242–260, 2012. 120

[BRW05] T. Brox, B. Rosenhahn, and J. Weickert. Three-dimensional shape knowledge for joint image segmentation and pose estimation. *Pattern Recognition*, pages 109–116, 2005. 34, 65

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001. 34

[CHKB08] P. Coupé, P. Hellier, C. Kervrann, and C. Barillot. Bayesian non local means-based speckle filtering. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 1291–1294. IEEE, 2008. 102

[CKS97] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997. 63

[CS10] F. Castro and M. Sbert. Incremental reuse of paths in random walk radiosity. *Large-Scale Scientific Computing*, pages 379–386, 2010. 99

[CT82] Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (TOG)*, 1(1):7–24, 1982. 12

[CTWS02] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional. *International journal of computer vision*, 50(3):295–313, 2002. 33

[CV01] T.F. Chan and L.A. Vese. Active contours without edges. *Image Processing, IEEE Trans. on*, 10(2):266–277, 2001. 34, 55

[DFKE06] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising with block-matching and 3 d filtering. In *Proceedings of SPIE*, volume 6064, pages 354–365, 2006. 102

[DLH01] K. Daubert, H.P.A. Lensch, and W. Heidrich. Efficient cloth modeling and rendering. In *Rendering techniques 2001: proceedings of the Eurographics workshop in London, United Kingdom, June 25-27, 2001*, page 63. Springer Verlag Wien, 2001. 15, 17

[DSHL10] H. Dammertz, D. Sewtz, J. Hanika, and H. Lensch. Edge-avoiding a-trous wavelet transform for fast global illumination filtering. In *Proceedings of High Performance Graphics*, pages 67–75. Eurographics Association, 2010. 100

[DvGNK97] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 151–157, 1997. 5

[DVGNK99] K.J. Dana, B. Van Ginneken, S.K. Nayar, and J.J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics (TOG)*, 18(1):1–34, 1999. 12

[ED08] E. Eisemann and X. Décoret. Occlusion textures for plausible soft shadows*. In *Computer Graphics Forum*, volume 27, pages 13–23. Wiley Online Library, 2008. 100

[EL99] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision-Volume 2*, page 1033. IEEE Computer Society, 1999. 102

[GG84] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *PAMI*, (6):721–741, 1984. 33

[GLH$^+$97] SA Grishanov, Stepan Vladimirovitch Lomov, RJ Harwood, T Cassidy, and C Farrer. The simulation of the geometry of two-component yarns. part i: The mechanics of strand compression: simulating yarn cross-section shape. *Journal of the Textile Institute*, 88(2):118–131, 1997. 77, 78, 79, 82

[GLPP08] B. Goossens, H. Luong, A. Pizurica, and W. Philips. An improved non-local denoising algorithm. In *Proceedings of the 2008 International Workshop on Local and Non-Local Approximation in Image Processing*, 2008. 102

[GOS09] RH Gong, B Ozgen, and Manuchehr Soleimani. Modeling of yarn cross-section in plain woven fabric. *Textile Research Journal*, 79(11):1014–1020, 2009. 84

[GRS95] Eduard Groller, Rene T. Rau, and Wolfgang Strasser. Modeling and visualization of knitwear. *IEEE Transaction on Visualization and Computer Graphics*, 1(4):302–310, 1995. 17, 18

[GSC11a] SA Grishanov, Francois Siewe, and T Cassidy. Advances in cad simulation of textile yarns. 2011. 77

[GSC11b] Sergei Grishanov, Francois Siewe, and Tom Cassidy. An application of queuing theory to modelling of melange yarns. part ii: A method of estimating the fibre migration probabilities and a yarn structure simulation algorithm. *Textile Research Journal*, 81(8):798–818, 2011. 79

[HDMS03] Vlastimil Havran, Cyrille Damez, Karol Myszkowski, and Hans-Peter Seidel. An efficient spatio-temporal architecture for animation rendering. In *Proceedings of the 14th Eurographics workshop on Rendering*, EGRW '03, pages 106–117, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 99

[HF11] M. Haindl and J. Filip. Advanced textural representation of materials appearance. In *SIGGRAPH Asia 2011 Courses*, page 1. ACM, 2011. 13

[HJW$^+$08] T. Hachisuka, W. Jarosz, R.P. Weistroffer, K. Dale, G. Humphreys, M. Zwicker, and H.W. Jensen. Multidimensional adaptive sampling and reconstruction for ray tracing. In *ACM Trans. Graph.*, volume 27, page 33. ACM, 2008. 101

[HKR01] You Huh, Young Ryul Kim, and Woon Young Ryu. Three-dimensional analysis of migration and staple yarn structure. *Textile Research Journal*, 71(1):81–89, 2001. 79

[HLY00] C.C. Huang, S.C. Liu, and W.H. Yu. Woven Fabric Analysis by Image Processing: Part I: Identification of Weave Patterns. *Textile Research Journal*, 70(6):481, 2000. 54, 55

[HPB07] Miloš Hašan, Fabio Pellacini, and Kavita Bala. Matrix row-column sampling for the many-light problem. *ACM Trans. Graph.*, 26(3), July 2007. 100

[HS81] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981. 44

[HS05] M. Heiler and C. Schnörr. Natural image statistics for natural image segmentation. *International Journal of Computer Vision*, 63(1):5–19, 2005. 34, 65

[HSE11] A. Hilsmann, D.C. Schneider, and P. Eisert. Warp-based near-regular texture analysis for image-based texture overlay. In *Vision, Modeling, and Visualization (2011)*, pages 73–80. The Eurographics Association, 2011. 42

[IM06] P. Irawan and S. Marschner. A simple, accurate texture model for woven cotton cloth. Technical report, Tech. Rep. PCG-06-01, Cornell University, Department of Computer Science, 2006. 26

[Ira07] P. Irawan. *Appearance of Woven Cloth*. PhD thesis, Cornell University, 2007. 16, 17

[JAM$^+$10] Wenzel Jakob, Adam Arbree, Jonathan T Moon, Kavita Bala, and Steve Marschner. A radiative transfer framework for rendering materials with anisotropic structure. *ACM Transactions on Graphics (TOG)*, 29(4):53, 2010. 18, 19

[JBS03] B.S. Jeon, J.H. Bae, and M.W. Suh. Automatic recognition of woven fabric patterns by an artificial neural network. *Textile Research Journal*, 73(7):645, 2003. 54

[JC95] H.W. Jensen and N. Christensen. Optimizing path tracing using noise reduction filters. In *Proceedings of WSCG*, volume 95, pages 134–142, 1995. 99

[Jeo08] Y. Jeong. Novel technique to align fabric in image analysis. *Textile Research Journal*, 78(4):304–310, 2008. 42

[JLO90] FV Jenson, SL Lauritzen, and KG Oleson. Bayesian updating in causal probabilistic networks by local computation. *Computational Statistics Quarterly*, 4(4):269–282, 1990. 65

[JMLH01] Henrik Wann Jensen, Stephen R Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 511–518. ACM, 2001. 15

[KBC07] C. Kervrann, J. Boulanger, and P. Coupé. Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal. In *Proceedings of the 1st international conference on Scale space and variational methods in computer vision*, pages 520–532. Springer-Verlag, 2007. 102

[Kee94] M Keefe. Solid modeling applied to fibrous assemblies. part i: Twisted yarns. *Journal of the Textile Institute*, 85(3):338–349, 1994. 77, 79

[KFIY$^+$05] J. Kim, J.W. Fisher III, A. Yezzi, M. Çetin, and A.S. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *Image Processing, IEEE Transactions on*, 14(10):1486–1502, 2005. 34, 65

[Kis12] Ruoss Kistler. *Kleines Zwirn Lexikon*, 2011 (accessed February 2, 2012). 23

[KJM08] J.M. Kaldor, D.L. James, and S. Marschner. Simulating knitted cloth at the yarn level. In *ACM Transactions on Graphics (TOG)*, volume 27, page 65. ACM, 2008. 29, 89

[KKO99] T.J. Kang, C.H. Kim, and K.W. Oh. Automatic recognition of fabric weave patterns by digital image analysis. *Textile Research Journal*, 69(2):77, 1999. 43, 55

[KSL04] C.F.J. Kuo, C.Y. Shih, and J.Y. Lee. Automatic recognition of fabric weave patterns by a fuzzy c-means clustering method. *Textile Research Journal*, 74(2):107–111, 2004. 55

[KSL05] J. Kautz, P.P. Sloan, and J. Lehtinen. Precomputed radiance transfer: theory and practice. In *ACM SIGGRAPH 2005 Courses*, page 1. ACM, 2005. 101

[KT06] C.F.J. Kuo and C.C. Tsai. Automatic recognition of fabric nature by using the approach of texture analysis. *Textile Research Journal*, 76(5):375, 2006. 54

[Kum08] Ajay Kumar. Computer-vision-based fabric defect detection: a survey. *Industrial Electronics, IEEE Transactions on*, 55(1):348–363, 2008. 49

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988. 63

[LAC$^+$11] J. Lehtinen, T. Aila, J. Chen, S. Laine, and F. Durand. Temporal light field reconstruction for rendering distribution effects. In *ACM Trans. Graph.*, volume 30, page 55. ACM, 2011. 100

[Law03] Carl A Lawrence. *Fundamentals of spun yarn technology*. CRC Press, 2003. 78

[LC00] Y. Liu and R.T. Collins. A computational model for repeated pattern perception using frieze and wallpaper groups. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 537–544. IEEE, 2000. 42

[LC01] Yanxi Liu and R.T. Collins. Skewed symmetry groups. In *CVPR*, volume 1, pages I–872 – I–879 vol.1, 2001. 42

[LFTG97] Eric PF Lafortune, Sing-Choong Foo, Kenneth E Torrance, and Donald P Greenberg. Non-linear approximation of reflectance functions. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 117–126. ACM Press/Addison-Wesley Publishing Co., 1997. 12

[LLH04] Y. Liu, W.C. Lin, and J. Hays. Near-regular texture analysis and manipulation. *ACM Transactions on Graphics (TOG)*, 23(3):368–376, 2004. 42

[LM96] T. Leung and J. Malik. Detecting, localizing and grouping repeated scene elements from an image. *Computer Vision – ECCV'96*, pages 546–555, 1996. 42

[Lor71] PR Lord. The structure of open-end spun yarn. *Textile Research Journal*, 41(9):778–784, 1971. 24

[LS88] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224, 1988. 65

[LSK$^+$07] S. Laine, H. Saransaari, J. Kontkanen, J. Lehtinen, and T. Aila. Incremental instant radiosity for real-time indirect illumination. In *Proceedings of Eurographics Symposium on Rendering*, pages 277–286. Citeseer, 2007. 100

[LT08] S. Lankton and A. Tannenbaum. Localizing region-based active contours. *Image Processing, IEEE Transactions on*, 17(11):2029–2039, 2008. 34, 65

[LZS$^+$12] Hua Lin, Xiesheng Zeng, Martin Sherburn, Andrew C Long, and Mike J Clifford. Automated geometric modelling of textile structures. *Textile Research Jour.*, 82(16):1689–1702, 2012. 89

[Mac02] C.K. Machens. Adaptive sampling by information maximization. *Physical review letters*, 88(22):228104, 2002. 101

[Mat03] Wojciech Matusik. *A data-driven reflectance model*. PhD thesis, Citeseer, 2003. 12

[MAT11] MATLAB. *version 7.13.0 (R2011b)*. The MathWorks Inc., Natick, Massachusetts, 2011. 64

[McC99] M.D. McCool. Anisotropic diffusion for monte carlo noise reduction. *ACM Trans. Graph.*, 18(2):171–194, 1999. 99

[MFSS06] À. Méndez-Feliu, M. Sbert, and L. SZIRMAYKALOS. Reusing frames in camera animation. *Journal of WSCG*, 14(3), 2006. 99

[MH62] W. E. Morton and J. W. S. Hearle. *Physical properties of textile fibres / W.E. Morton and J.W.S. Hearle*. Textile Institute, Manchester, England :, 1962. 20

[Mit87] D.P. Mitchell. Generating antialiased images at low sampling densities. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 65–72. ACM, 1987. 101

[MJC$^+$03] Stephen R. Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. Light scattering from human hair fibers. *ACM TOG*, 22(3):780–791, 2003. SIGGRAPH 2003. 20, 21

[MM06] Jonathan T. Moon and Stephen R. Marschner. Simulating multiple scattering in hair using a photon mapping approach. *ACM Trans. Graph.*, 25(3):1067–1074, 2006. SIGGRAPH 2006. 21

[MMR99] P. J. Morris, J. H. Merkin, and R. W. Rennell. Modeling of yarn properties from fiber properties. *Journal of Textile Institute*, 90:322–335, 1999. 78

[MMS+05] G. Müller, J. Meseth, M. Sattler, R. Sarlette, and R. Klein. Acquisition, synthesis, and rendering of bidirectional texture functions. In *Computer Graphics Forum*, volume 24, pages 83–109. Wiley Online Library, 2005. 13

[MMSK03] Jan Meseth, Gero Müller, Mirko Sattler, and Reinhard Klein. Btf rendering for virtual environments. In *Virtual Concepts 2003*, pages 356–363, November 2003. 14

[MN88] D.P. Mitchell and A.N. Netravali. Reconstruction filters in computer-graphics. In *ACM Siggraph Computer Graphics*, volume 22, pages 221–228. ACM, 1988. 105

[MP94] S Mann and RW Picard. *Being undigital with digital cameras*. MIT Media Lab Perceptual, 1994. 35

[MS88] D. Mumford and J. Shah. Boundary detection by minimizing functionals. *Image understanding*, pages 19–43, 1988. 34

[MS89] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989. 34

[MTCK+04] N. Magnenat-Thalmann, F. Cordier, M. Keckeisen, S. Kimmerle, Reinhard Klein, and Jan Meseth. Simulation of clothes for real-time applications. In *Eurographics 2004, Tutorials*. INRIA and the Eurographics Association, 2004. 17

[MWM08] Jonathan T. Moon, Bruce Walter, and Steve Marschner. Efficient multiple scattering in hair using spherical harmonics. *ACM Transactions on Graphics*, 27(3), 2008. SIGGRAPH 2008. 21, 124, 130, 131, 132

[NDM05] Addy Ngan, Frédo Durand, and Wojciech Matusik. Experimental analysis of brdf models. In *Proceedings of the Sixteenth Eurographics conference on Rendering Techniques*, pages 117–126. Eurographics Association, 2005. 12

[NPY11] Henry YT Ngan, Grantham KH Pang, and Nelson HC Yung. Automated fabric defect detection – a review. *Image and Vision Computing*, 29(7):442–458, 2011. 49

[ODR09] R.S. Overbeck, C. Donner, and R. Ramamoorthi. Adaptive wavelet rendering. *ACM Trans. Graph*, 28(5):1–12, 2009. 101

[oSN77] United States. National Bureau of Standards and Fred Edwin Nicodemus. *Geometrical considerations and nomenclature for reflectance*, volume 160. US Department of Commerce, National Bureau of Standards Washington, D. C, 1977. 12

[PBCL09] M. Park, K. Brocklehurst, R.T. Collins, and Y. Liu. Deformed lattice detection in real-world images using mean-shift belief propagation. *PAMI*, 31(10):1804–1816, 2009. 42

[Per85] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, 1985. 80

[PG09] R. Pan and W. Gao. Automatic skew rectification of image of high density woven fabric. *Jour. of Textile Research*, 2009. 42

[PGL09] R. Pan, W. Gao, and J. Liu. Color Clustering Analysis of Yarn-dyed Fabric in HSL Color Space. In *Proceedings of the 2009 WRI World Congress on Software Engineering-Volume 02*, pages 273–278. IEEE Computer Society, 2009. 54

[PGL$^+$10] R. Pan, W. Gao, J. Liu, H. Wang, and X. Zhang. Automatic Detection of Structure Parameters of Yarn-dyed Fabric. *Textile Research Journal*, 2010. 54

[PGLW10a] R. Pan, W. Gao, J. Liu, and H. Wang. Automatic Detection of the Layout of Color Yarns for Yarn-dyed Fabric via a FCM Algorithm. *Textile Research Journal*, 2010. 54

[PGLW10b] R. Pan, W. Gao, J. Liu, and H. Wang. Automatic recognition of woven fabric patterns based on pattern database. *Fibers and Polymers*, 11(2):303–308, 2010. 54

[PGM10] D.J. Peter, VK Govindan, and A.T. Mathew. Nonlocal-means image denoising technique using robust m-estimator. *Journal of computer science and technology*, 25(3):623–631, 2010. 102

[PGZ08] R. Pan, W. Gao, and X. Zhang. High-precision skew rectification and validation for fabric image. *Journal of Textile Research*, 29(4):51, 2008. 42

[Pho75] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975. 12

[PSA$^+$04] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 664–672. ACM, 2004. 100

[Pur87] W. Purgathofer. A statistical method for adaptive stochastic sampling. *Computers & Graphics*, 11(2):157–162, 1987. 101

[PUZ$^+$07] Thomas Pock, Martin Urschler, Christopher Zach, Reinhard Beichel, and Horst Bischof. A duality based algorithm for tv-l 1-optical-flow image registration. In *MICCAI*, pages 511–518. Springer, 2007. 44

[RFS03] J. Rigau, M. Feixas, and M. Sbert. Entropy-based adaptive sampling. In *Graphics Interface*, pages 149–157, 2003. 101

[RKB04] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314. ACM, 2004. 34

[RKZ11] F. Rousselle, C. Knaus, and M. Zwicker. Adaptive sampling and reconstruction using greedy error minimization. In *ACM Trans. Graph.*, volume 30, page 159. ACM, 2011. 101

[RKZ12] Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. Adaptive rendering with non-local means filtering. *ACM Transactions on Graphics (TOG)*, 31(6):195, 2012. 120

[RMZ13] Fabrice Rousselle, Marco Manzi, and Matthias Zwicker. Robust denoising using feature and color information. In *Computer Graphics Forum*, volume 32, pages 121–130. Wiley Online Library, 2013. 120

[Rob94] C.R. Robbins. Chemical and physical behavior of human hair. *third ed. Springer-Verlag, New York*, 1994. 20

[RSK12] Roland Ruiters, Christopher Schwartz, and Reinhard Klein. Data driven surface reflectance from sparse and irregular samples. *Computer Graphics Forum (Proc. of Eurographics)*, 31(2):315–324, May 2012. 12

[RW94] H.E. Rushmeier and G.J. Ward. Energy preserving non-linear filters. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 131–138. ACM, 1994. 99

[RWJ09] Pan Ruru, G. Weidong, and L. Jihong. Skew rectification for yarn-dyed fabric via fft in hsl color space. In *Software Engineering, 2009. WCSE'09. WRI World Congress on*, volume 2, pages 481–485. IEEE, 2009. 42

[S.04] USDA E. R. S. http://www.ers.usda.gov/briefing/cotton/, 2004. 22

[SB06] K. Sreprateep and E.L.J. Bohez. Computer Aided Modelling of Fiber Assemblies. *Comput. Aided Des. Appl*, 3(1-4):367–376, 2006. 22, 77, 78, 79

[SBDDJ13] Iman Sadeghi, Oleg Bisker, Joachim De Deken, and Henrik Wann Jensen. A practical microcylinder appearance model for cloth rendering. *ACM Transactions on Graphics (TOG)*, 32(2):14, 2013. 16

[SC04] M. Sbert and F. Castro. Reuse of paths in final gathering step with moving light sources. *Computational Science-ICCS 2004*, pages 189–196, 2004. 99

[SD12] Pradeep Sen and Soheil Darabi. On filtering the noise from the random parameters in monte carlo rendering. *ACM Trans. Graph.*, 31(3):18, 2012. 100

[SGCB09] Francois Siewe, Sergei Grishanov, Thomas Cassidy, and Geoffrey Banyard. An application of queuing theory to modeling of melange yarns part i: A queuing model of melange yarn structure. *Textile Research Journal*, 79(16):1467–1485, 2009. 78, 79

[Sha02] Irfan Ahmed Shaikh. *Pocket Textile Expert: Mini Textile Encyclopedia*. Textile Info Society, 2002. 23, 24

[Sha07] Irfan Ahmed Shaikh. *Pocket Weaving Expert: A Practical Handbook on Textile Weaving*. Textile Info Society, 2007. 27, 87

[Shi07] T. Shinohara. Iterative extraction of yarn positional information from three-dimensional image of textile fabric. In *SICE, 2007 Annual Conference*, pages 2842–2848, 2007. 56

[SIMP06] B. Segovia, J.C. Iehl, R. Mitanchey, and B. Péroche. Non-interleaved deferred shading of interleaved sample patterns. In *Proceedings of the 21st ACM SIGGRAPH/EUROGRAPHICS* *symposium on Graphics hardware*, pages 53–60. ACM, 2006. 100

[SKZ11] Kai Schröder, Reinhard Klein, and Arno Zinke. A volumetric approach to predictive rendering of fabrics. *CGF*, 30(4):1277–1286, July 2011. 8, 124

[SKZ13] Kai Schröder, Reinhard Klein, and Arno Zinke. Non-local image reconstruction for efficient computation of synthetic bidirectional texture functions. *Computer Graphics Forum (accepted for publication)*, 2013. 8, 99, 120

[SMKZ11] K. Schröder, D. Möller, R. Klein, and A. Zinke. Non-local image reconstruction for efficient btf synthesis. In *SIGGRAPH Asia 2011 Sketches*, page 30. ACM, 2011. 8, 99, 120

[SRB10] D. Sun, S. Roth, and M.J. Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439. IEEE, 2010. 43, 44

[SRWK11] Christopher Schwartz, Roland Ruiters, Michael Weinmann, and Reinhard Klein. Webgl-based streaming and presentation framework for bidirectional texture functions. In *The 12th International Symposium on Virtual Reality, Archeology and Cultural Heritage VAST 2011*, pages 113–120. Eurographics Association, Eurographics Association, October 2011. Best Paper Award. 14

[SSK03] M. Sattler, R. Sarlette, and R. Klein. Efficient and realistic visualization of cloth. In *Proceedings of the 14th Eurographics workshop on Rendering*, pages 167–177. Eurographics Association Aire-la-Ville, Switzerland, Switzerland, 2003. 17

[STOK04] T. Shinohara, J. Takayama, S. Ohyama, and A. Kobayashi. Analysis of textile fabric structure based on three-dimensional fiber model matching method. In *SICE, 2004 Annual Conference*, volume 1, page 399. SICE; 1999, 2004. 56

[STOK06] T. Shinohara, J. Takayama, S. Ohyama, and A. Kobayashi. Yarn Segmentation Based on Filament Direction in 3-D CT Images for Structure Analysis of Textile Fabric. In *IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*, pages 3338–3343, 2006. 56

[STOK09] T. Shinohara, J. Takayama, S. Ohyama, and A. Kobayashi. Extraction of Yarn Positional Information From Three-Dimensional ct Image of Textile Fabric Using Yarn Tracing With Filament Model for Structure Analysis. *Textile Research Journal*, 2009. 56

[SvBLD03] Frank Suykens, Karl vom Berge, Ares Lagae, and Philip Dutré. Interactive Rendering of Bidirectional Texture Functions. In *Eurographics 2003*, pages 463–472, September 2003. 100

[SZ99] F. Schaffalitzky and A. Zisserman. Geometric grouping of repeated elements within images. *Shape, Contour and Grouping in Computer Vision*, pages 81–81, 1999. 42

[SZK15] K. Schroder, A. Zinke, and R. Klein. Image-based reverse engineering and visual prototyping of woven cloth. *Visualization and Computer Graphics, IEEE Transactions on*, 21(2):188–200, Feb 2015. 8, 37, 74

[Tao96] Xiaoming Tao. Mechanical properties of a migrating fiber. *Textile research journal*, 66(12):754–762, 1996. 79

[TJ97] R. Tamstorf and H.W. Jensen. Adaptive sampling and bias estimation in path tracing. In *Eurographics Workshop on Rendering Techniques*, volume 97, pages 285–296, 1997. 101

[TM98] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998. 99

[Tre65] LRG Treloar. A migrating-filament theory of yarn properties. *Journal of the Textile Institute Transactions*, 56(7):T359–T380, 1965. 78

[VBP10] CW Verberne, LAA Beex, and RHJ Peerlings. Mgd geers mechanical modelling of textiles. *Literature survey, Eindhoven University of Technology*, 2010. 84

[VGNI04] Jana Voborova, A Garg, B Neckar, and S Ibrahim. Yarn properties measurement: an optical approach. In *International textile, clothing and design conference*, 2004. 77, 80

[VKKK97] Vladimir L. Volevich, Edward A. Kopylov, Andrei B. Khodulev, and Olga A. Karpenko. An approach to cloth synthesis and visualization. In *The 7-th International Conference on Computer Graphics and Visualization*, 1997. 16

[WAT92] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance. Predicting reflectance functions from complex surfaces. *SIGGRAPH Comput. Graph.*, 26(2):255–264, 1992. 16

[WBBP05] Joachim Weickert, Andrés Bruhn, Thomas Brox, and Nils Papenberg. A survey on variational optic flow methods for small displacements, 2005. 43

[Wer23] Max Wertheimer. Untersuchungen zur lehre von der gestalt. ii. *Psychological Research*, 4(1):301–350, 1923. 39

[WGP11] Xin Wang, Nicolas D Georganas, and Emil M Petriu. Fabric texture analysis using computer vision techniques. *Instrumentation and Measurement, IEEE Transactions on*, 60(1):44–56, 2011. 56

[WHON97a] Tien-Tsin Wong, Pheng-Ann Heng, Siu-Hang Or, and Wai-Yin Ng. Image-based rendering with controllable illumination. In *Proceedings of EGRW '97*, pages 13–22. Springer-Verlag, 1997. 12

[WHON97b] Tien-Tsin Wong, Pheng-Ann Heng, Siu-Hang Or, and Wai-Yin Ng. Image-based rendering with controllable illumination. In *Rendering Techniques 97*, pages 13–22. Springer, 1997. 99

[WKB$^+$02] I. Wald, T. Kollig, C. Benthin, A. Keller, and P. Slusallek. Interactive global illumination using fast ray tracing. In *Proceedings of the 13th Eurographics workshop on Rendering*, pages 15–24. Eurographics Association, 2002. 100

[WSI07] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(3):463–476, 2007. 102

[WT07] Charlie CL Wang and Kai Tang. Woven model based geometric design of elastic medical braces. *Computer-Aided Design*, 39(1):69–79, 2007. 89

[WTY05] Charlie CL Wang, Kai Tang, and Benjamin ML Yeung. Freeform surface flattening based on fitting a woven mesh model. *Computer-Aided Design*, 37(8):799–814, 2005. 89

[WZP06] H. Wu, M. Zhang, and Z. Pan. Skew detection and weft density identification for fabric images [j]. *Journal of Image and Graphics*, 5, 2006. 42

[WZPY05] H. Wu, M. Zhang, Z. Pan, and H. Yin. Automatic identifying weave patterns for double-layer weft woven fabric. In *Computer Graphics, Imaging and Vision: New Trends, 2005. International Conference on*, pages 117–122, 2005. 55

[WZT$^+$08] Jiaping Wang, Shuang Zhao, Xin Tong, John Snyder, and Baining Guo. Modeling anisotropic surface reflectance with example-based microfacet synthesis. *ACM Trans. Graph. (to appear in SIGGRAPH 2008)*, 27(3):41:1–41:9, 2008. 17

[XCL$^+$01] Ying-Qing Xu, Yanyun Chen, Stephen Lin, Hua Zhong, Enhua Wu, Baining Guo, and Heung-Yeung Shum. Photorealistic rendering of knitwear using the lumislice. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 391–398, New York, NY, USA, 2001. ACM Press/Addison-Wesley Publishing Co. 17, 18

[XH09] B. Xin and J. Hu. Investigation on the Classification of Weave Pattern Based on an Active Grid Model. *Textile Research Journal*, 79(12):1123, 2009. 55

[XP05] R. Xu and S.N. Pattanaik. Non-iterative, robust monte carlo noise reduction. *IEEE Computer Graphics and Applications*, 25(2):31–35, 2005. 99

[YKJM12] Cem Yuksel, Jonathan M. Kaldor, Doug L. James, and Steve Marschner. Stitch meshes for modeling knitted clothing with yarn-level detail. *ACM Trans. Graph.*, 31(4):37:1–37:12, July 2012. 89

[ZHRB13] Shuang Zhao, Miloš Hašan, Ravi Ramamoorthi, and Kavita Bala. Modular flux transfer: Efficient rendering of high-resolution volumes with repeated structures. *ACM Transactions on Graphics (TOG)*, 32(4):131, 2013. 145

[ZI10] M. Zontak and M. Irani. Internal statistics of a single natural image. *Machine Vision and Applications*, 21(2):129–141, 2010. 99

[ZJMB11] Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. Building volumetric appearance models of fabric using micro ct imaging. *ACM Trans. Graph*, 30(44):1–44, 2011. 19, 73, 89

[ZJMB12] Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. Structure-aware synthesis for predictive woven fabric appearance. *ACM Trans. on Graphics (TOG)*, 31(4):75, 2012. 19, 56, 73

[ZLHA$^+$09] Arno Zinke, Tomas Lay Herrrera, Anton Andriyenko, Martin Rump, Andreas Weber, and Reinhard Klein. A practical approach for photometric acquisition of hair color. December 2009. 125, 157, 158

[ZREB06] Todd Zickler, Ravi Ramamoorthi, Sebastian Enrique, and Peter N Belhumeur. Reflectance sharing: Predicting appearance from a sparse set of images of a known shape. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(8):1287–1302, 2006. 100

[ZSW04] A. Zinke, G. Sobottka, and A. Weber. Photo-realistic rendering of blond hair. In B. Girod, M. Magnor, and H.-P. Seidel, editors, *Vision, Modeling, and Visualization (VMV) 2004*, pages 191–198, Stanford, U.S.A., November 2004. IOS Press. 21

[ZW07a] Arno Zinke and Andreas Weber. Light scattering from filaments. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):342–356, 2007. 20, 21, 111, 142

[ZW07b] Arno Zinke and Andreas Weber. Light scattering from filaments. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):342–356, 2007. 77, 83, 158

[ZXW13] Jie Zhang, Binjie Xin, and Xiangji Wu. A review of fabric identification based on image analysis technology. *Textiles and Light Industrial Science and Technology*, 2(3), 2013. 53

[ZY96] S.C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE PAMI*, 18(9):884–900, 1996. 34

[ZYWK08] Arno Zinke, Cem Yuksel, Andreas Weber, and John Keyser. Dual scattering approximation for fast multiple scattering in hair. *ACM TOG*, 27(3), 2008. SIGGRAPH 2008. 21

LIST OF FIGURES

LIST OF TABLES