STAG: Smart Tools and Apps in computer Graphics (2016)
Giovanni Pintore and Filippo Stanco (Editors)

Practical Offline Rendering of Woven Cloth

Vidar Nelson† Peter M. McEvoy† Marco Fratarcangeli

Chalmers University of Technology

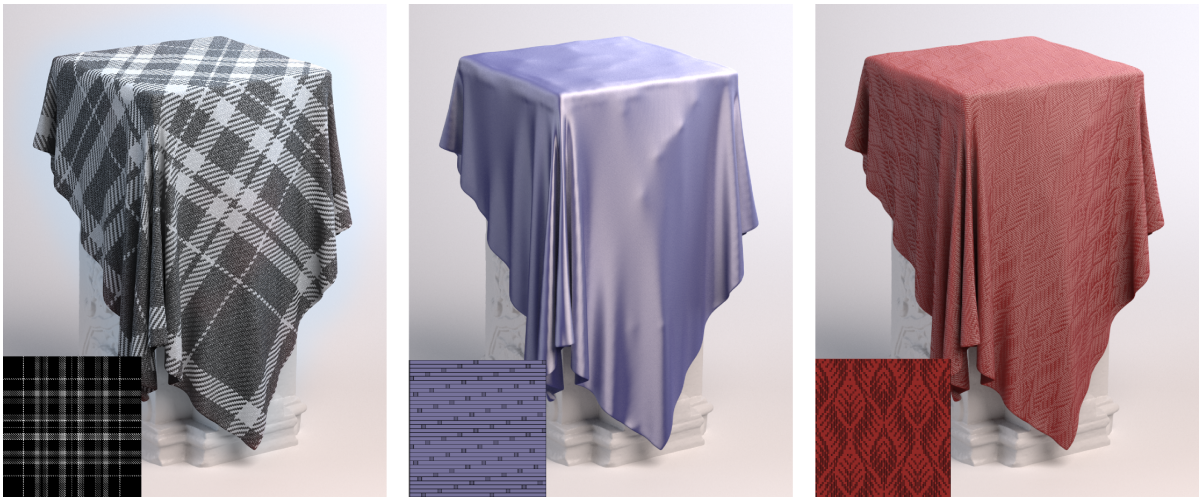


Figure 1: A set of fabrics rendered using our shader, and their corresponding weaving patterns that are stored in publicly available WIF files.

Abstract

The techniques for rendering woven cloth employed in a production environment often neglect many of the structural features of the fabric, as well as light-scattering processes that occur in the yarn. Research in this area has progressed, and new promising methods have recently been proposed; however, many of these are not applied in practice due to their inherent complexities. In this paper, we develop and implement a shader for woven cloth that fulfills some of the needs of a production environment using an existing model for simulating the interaction with light. The shader delivers highly realistic results that are comparable, and in some cases superior, to current methods used in a real production environment. We enriched and validated the proposed framework by using direct feedback from a large company that produces images of furniture using computer graphics. The results demonstrate that our shader accurately simulates the appearance of certain types of woven cloth, showing reflections that are not present in other methods in current use. Our shader is easy to integrate in existing pipelines, flexible because it provides the artist with enough parameters to recreate many different types of fabric, and generic for the domain of woven cloth because it is able to accept as input widely used WIF files, which describe weave patterns as well as yarn specific parameters.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

Creating accurate and appealing product images is a craft that involves a substantial amount of work, especially when considering that there often are multiple versions with subtle variations of these products. There is an ongoing trend to produce product images through computer graphics (e.g. Fig. 2); a trend that significantly reduces time and cost, but still requires manual craftsmanship.

The complex surface and structure of woven textiles gives rise to interesting interactions with light. Both the type of yarn and the pat-

† Joint first author

DOI: 10.2312/stag.20161365

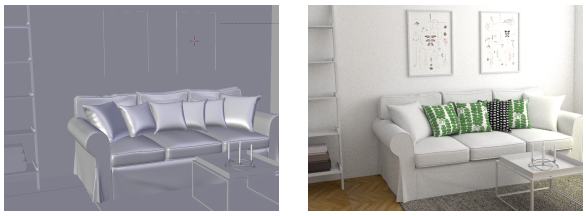


Figure 2: *Left.* 3D furniture, as seen in a 3D editor. *Right.* The same scene rendered with realistic lighting.

tern with which it is woven contribute to the final appearance of the fabric, often resulting in highly anisotropic reflections, a prominent feature of fabrics such as satin or velvet. In order to generate photorealistic images of woven cloth, it is critical to take into account the way light interacts with the yarn and the pattern that make up these materials.

In many production environments the approach is to use texture maps and a general purpose shader to reproduce the appearance of woven cloth. Each texture is individually crafted by skilled artists using high resolution photographs as reference. However, while it can generate high quality results, there are also several drawbacks: it involves a significant amount of artistic work and the final quality of the results depends heavily on the skills of the artists. Furthermore, these high resolution textures require a large amount of memory during rendering and still often fail to capture the anisotropic light interaction with the individual yarns. Fig. 3 is an example of this texture based approach. The image shows the textures used to simulate the look of a fabric, together with the rendered result.

Figure 3: A set of textures designed to simulate the look of a fabric, and a rendered piece of cloth using these textures.

This paper proposes a solution to these issues by implementing a production ready shader for woven cloth that can produce high quality images while giving artists enough control that any common type of woven fabric can be recreated. This means that the physical properties of how light interacts with the yarn need to be taken into account when deciding on the model to be used for the shader. The shader and the underlying model for light-yarn interaction have to allow for effective control to allow artists to quickly and flexibly recreate many different types of fabric. It is also important that that the resulting shader is easy to implement and can be integrated with different rendering engines, since there are many different tools used in a production environment.

From studying existing and well-regarded shading models for yarn in the context of woven cloth, a suitable model was chosen as a base for the work, specifically the works by Irawan and Marschner [IS06, Ira08, IM12]. From this model, we developed a shader that is physically plausible and has enough controls and flexibility to be useful in production.

Our shader has three characteristics that make it particularly well suited for production-ready realistic rendering:

Realistic. The shader uses a physically based model for simulating the scattering of light inside the yarn. In addition, the characteristic anisotropic reflection arising from the twisting nature of most yarn is simulated. An accurate simulation of these phenomena makes the images produced by the shader more realistic compared to current techniques employed in production environments.

Flexible. Basic parameters are presented to the user in a graphical interface to allow control of the underlying model. In order to allow the artist to recreate more types of materials, most parameters can be altered for each yarn in the fabric, thus allowing for tighter control over how the shader is to behave. In addition, some of the shader parameters can be controlled through textures, giving the shader further versatility.

Generic. In order to fit into the production environment at ICOM, the shader was implemented as a plug-in for the rendering engine V-Ray and the 3D authoring software Autodesk 3ds Max. The core part of the shading model, however, was implemented in a portable fashion as to allow for easy integration into other production pipelines. In addition, we implemented a method for parsing and incorporating digital representations of weaving patterns (contained in WIF files), which are commonly used in the textile industry. By reading the same file format as is used during the design and weaving of the fabrics, the manual work involved in adding a new type of fabric is minimized. The structure of the weave is automatically inferred from the weaving pattern and the artist only needs to tweak the reflectance properties of the yarn in order to match a physical sample.

The complete source code for the shader is available as open source software: see `https://github.com/vidarn/ThunderLoom`.

2. Related Work

As mentioned above, a common method of simulating the appearance of woven cloth relies on texture maps and a general purpose shading model. An alternative to crafting textures by hand is to generate them automatically by illuminating a physical sample of the surface with spherical gradient patterns [GCP*09]. A large amount of information regarding the surface, such as glossiness and anisotropy, can be captured through this measurement technique. This method still has drawbacks, however: It requires a complicated hardware setup and still involves a significant amount of artistic work and time, since the resulting texture maps still need to be edited in order to avoid tiling issues. These issues, together with the fundamental difficulty of accurately recreating the complex light interaction of cloth using only texture maps limits the usefulness of the texture based approach.

In order to achieve higher quality results, more specialized methods need to be considered. There is a significant body of literature addressing the problem of rendering of fabrics. A good overview of current models for simulating the appearance of cloth can be found in some of this literature [SZZ12, KSZ*15]. A set of methods which achieve high visual detail are those that represent the volumetric geometry of a fabric at the yarn level. Examples of this can be seen in work aimed at simulating the appearance and geometry of knitwear [Shu03]; a different but closely related problem. A volumetric representation of the knitted yarn geometry is generated, which can then be used to simulate the light yarn interaction. Another example of this method, applied to woven cloth, uses volumetric data from micro-CT scans of real fabrics [ZJMB12] and is able to reproduce the appearance of the cloth at a very high magnification. The volumetric data coupled with the micro-flake model for anisotropic light scattering [JAM*10, HDCD15] can give very impressive results. This method successfully captures the fine details of woven cloth and explicitly simulates the scattering that occurs at the surface and within the yarn. However, these methods share some of the issues with the texture based approach. Representing the volumetric geometry of individual yarns requires a large amount of data and becomes resource-intensive when considering larger pieces of textiles. Furthermore, each new fabric incurs a great cost since it requires substantial manual labour or complicated measurements to get the required data sets. For production environments that need to create hundreds of shaders each year, this is a large drawback; however, recent advances in procedurally generating volumetric geometry show promise in alleviating some of these issues [ZLB16].

Another approach is to use a custom bidirectional reflection distribution function (BRDF) [IM12]; a method that treats the textile as a 2D surface and uses a mathematical model of light-yarn interaction to simulate the surface appearance of woven cloth, without relying on high resolution input data or needing to generate a volumetric representation. While this means that the visual quality at high magnification is reduced in comparison to the volumetric approach, it can provide excellent results from a distance. Furthermore, the BRDF based approach generally requires far less memory and is easier for an artist to control.

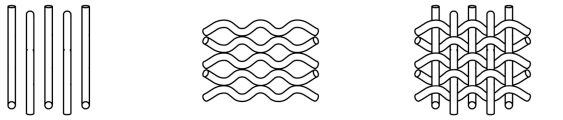
The use of digital weaving drafts for rendering woven cloth has been introduced in earlier work [AMTF03], where the information stored in WIF files was used to visualize arbitrary weaving patterns. While the proposed shading model in this work is less sophisticated than the ones mentioned above, the weaving patterns were faithfully reconstructed and extra information including transparency was recovered from the weaving drafts.

3. Background

In this section, some fundamental concepts of woven cloth are introduced, as well as some shading-related terms and concepts. In an effort to make the paper more readable and self-contained, some of the key theoretical concepts introduced in the work by Irawan and Marschner [IM12] are also summarized here.

3.1. Fundamentals of Woven Fabric

The basic process of weaving consists of interlacing yarns in different patterns using a loom. Two sets of yarn are used; the *weft* running along the loom and the *warp* running orthogonal to it (Fig. 4). The loom lifts a selection of the warp threads, allowing the weft to be slid through, passing below the raised warp threads and above the others. This process is repeated until the fabric is complete [Ada01].



(a) Warp. (b) Weft. (c) Complete weave.

Figure 4: The warp and weft in a woven fabric. The weft runs parallel to the loom and the warp runs orthogonal to it.

The sequence in which the warp and weft threads are interwoven is defined by a *weaving pattern*. Fabrics that are woven with different patterns often have different physical and visual characteristics, even when the same type of yarn is used.

3.2. Modeling Yarn Reflections

A yarn is made up of many smaller filaments that are packed together. These filaments are often somewhat translucent, and once the light enters the yarn, it is scattered by the reflection from the filaments. Some of the light will continue along the path of incidence, but at each intersection with a filament, part of the light will be reflected.

The model proposed by Irawan and Marschner [IM12] does not calculate the intersection for each individual filament. Instead, it considers the average number of intersections that would occur when the light travels a certain distance in to the yarn. It is assumed that when the light intersects a filament, some of it is reflected specularly, and some of it is transmitted through the filament. Furthermore, a portion of the transmitted light will be absorbed by the filament, which means that less light will leave the filament than that which entered it. When all these factors are considered, the resulting BRDF can be presented as the product of three factors,

$$f_r(x,\vec{\omega}_i,\vec{\omega}_o) = \underbrace{f_p(\vec{\omega}_i,\vec{\omega}_o)}_{\textit{Phase function}} \overbrace{A(x,\vec{\omega}_i,\vec{\omega}_o)}^{\textit{Attenuation function}} \underbrace{G(x,\vec{\omega}_i,\vec{\omega}_o)}_{\textit{Geometry factor}}, \quad (1)$$

where x is the current shading point, and $\vec{\omega}_i$ and $\vec{\omega}_o$ are the incident and outgoing directions of the light, respectively.

The phase function, f_p, describes the scattering inside the yarn and simulates the way light interacts with the filament fibers. It determines the amount of light that is scattered from the direction $\vec{\omega}_i$ to the direction $\vec{\omega}_o$ at any point. Irawan and Marschner [IM12] introduce two parameter variables α and β and use the phase function

$$f_p(\vec{\omega}_i,\vec{\omega}_o,\alpha,\beta) = \alpha + g(\vec{\omega}_i\cdot\vec{\omega}_o,0,\beta),$$

which is the sum of a constant term, α, and the *von Mises* probability distribution $g(\cos\theta,a,b)$, which includes β as a parameter.

The von Mises function is analogous to the normal distribution but defined in terms of an angle, θ. Together, the parameters α and β of the phase function factor can be used to change the behavior of how the light is scattered in the yarn.

The attenuation function, $A(x, \vec{\omega}_i, \vec{\omega}_o)$, is used to model that some of the light is absorbed by the filaments of the yarn. Irawan and Marschner [IM12] take advantage of a formulation based on *Seeliger's law* [HK93], which describes the absorption inside a homogeneous medium under a flat surface. The assumption of a flat surface is not true of a yarn cylinder, but the error introduced by this approximation is considered minimal.

The geometry factor, G, does not model any specific phenomenon. Instead, it is the result of the parametrization of the yarn cylinders. It can be found by considering an integral over an entire yarn segment, as shown in [Ira08, p. 64]. However, the derivation is quite involved and not relevant to the current discussion.

4. Design and Implementation

This section describes the implementation of our production ready shader based on the BRDF introduced by [IM12]. This includes the reading of WIF files, integrating the shading model with professional rendering software and adding functionalities that make the shader more flexible for production use.

4.1. Reading and Interpreting Weaving Patterns

A standard file format for weaving drafts was used in order to streamline the process of adding new fabrics. The *WIF*, or *Weaving Information File*, file format is a text based format supported by most weaving software [Nie97]. The contents of a WIF file describes the layout of the weaving pattern, as well as the the color of each yarn. We parse this information and represent it as a matrix where each entry describes one intersection of warp and weft and contains a boolean value indicating whether the warp or weft thread is on top of the other. It also includes the color of the topmost yarn.

Furthermore, the WIF file contains information regarding the spacing of the warp and weft threads, which we use to automatically scale the pattern to the correct size when applied to a 3D model.

4.2. Representing the Yarn Geometry

In a weave pattern, each thread might pass over several orthogonal threads at a time. We treat each such continuous run of thread as a single *yarn segment*. A yarn segment starts and ends when it is being covered by another thread. Fig. 5 shows an example of such a segment, where the weft passes over two warp threads. In real cloth, a visible yarn segment has hidden yarns running perpendicularly under it. This gives the yarn segment a bent shape. To approximate this behavior, the yarn segments are treated as bent cylinders. The local coordinates $(x, y) \in [-1, 1]$ of a yarn segment are transformed to the surface of such a cylinder using the mapping

$$u = \arcsin\left(2\frac{y}{l}\sin(u_{max})\right)$$
$$v = \arcsin\left(2\frac{x}{w}\right),$$

where l and w are the length and width of a yarn segment, respectively. The parameter u_{max} describes the radius of the bend. Fig. 6 illustrates this transformation.

A key part of the model proposed by Irawan, Marschner [IM12] is that it takes the twisting nature of the filaments into account. The parameter ψ determines the *twist* of the yarn. This describes the angle the filaments make with the yarn cylinder, see Fig. 6. The tangent vector, $\vec{t}$ (not shown in the figure), is defined to be orthogonal to the normal and parallel to the direction of the filaments.

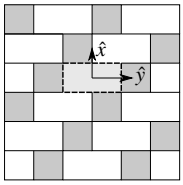


Figure 5: A pattern matrix with a highlighted yarn segment in the center. The corresponding yarn local coordinate system is also illustrated. The y-axis is defined such that it always runs along the yarn, while the x-axis runs across it.

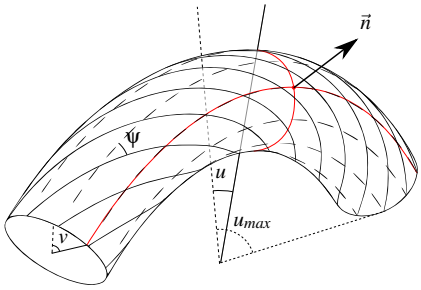


Figure 6: The coordinate u is the angle to the current position along the length of the yarn and the coordinate v is the angle along the short side of the yarn. The vector $\vec{n}$ is the normal of the bent cylinder surface. The twist parameter ψ determines the angle the filaments make with the yarn.

4.3. Normalizing the Reflection

The *Energy Conservation* condition for physically plausible BRDFs states that the integral of the BRDF over the hemisphere is less than or equal to 1 [PH10]. In order to ensure this, we introduce a scaling factor

$$a = \left(\max_{x,\vec{\omega}_i} \int_\Omega f_r(x, \vec{\omega}_i, \vec{\omega}_o)(\vec{\omega}_i \cdot \vec{n}) \mathrm{d}\vec{\omega}_o\right)^{-1}. \quad (2)$$

The maximization is applied for all points $x = (u, v)$ such that $u \in [0, 1]$ and $v \in [0, 1]$ to make sure that the shader is normalized over the entire piece of cloth, and for all values of $\vec{\omega}_i \in \Omega$, where Ω is the hemisphere of directions above the surface, to make sure it is normalized for all incident angles. By normalizing the BRDF

with the maximum value over these dimensions, the condition is fulfilled.

We evaluated Eq. (2) prior to each rendering session using Monte Carlo integration at a number of random points for u, v and $\vec{\omega}_o$, and recording the maximum value encountered. In order to get a good estimate of the normalization factor a in Eq. (2), and to avoid flicker in animations, we use the Halton low discrepancy sequence instead of purely random points.

4.4. Integrating the Shader with Different Rendering Engines

We integrated the cloth shader into two different rendering engines, Mitsuba and V-Ray. These renderers are designed with different goals in mind. Mitsuba is developed for academic purposes, aiming to be easy to extend with new functionalities. The main goal of V-Ray, on the other hand, is to be fast and flexible to use for 3D artists, but not necessarily for a developer. For this reason, Mitsuba was used for the initial development, since it allowed for faster iteration time and a provided a more user-friendly API. V-Ray was used for the final implementation.

Since the implementation of the cloth shader was meant to support multiple renderers, the core of the shader had to be kept independent of any specific code base. We implemented, in a portable fashion, a set of functions that made up the core of the shading model. Functions related to storing and representing weaving patterns, as well as functions for evaluating the diffuse and specular reflections compose a simple API. Using this API, the plug-ins for different renderers can call the generalized shader code. This allowed the shader to be more easily portable to other renderers. The user interface, and interaction with the native material system of each renderer still had to be developed separately for each rendererings-solution, however.

4.5. Flexible Artist Control

The user interacts with the shader through a graphical interface inside 3ds Max. This interface exposes the parameters of the model and provides the option to control some of the parameters by using textures.

The shader is able to recreate complex fabrics which consist of several different types of yarn. This is achieved by allowing the reflection parameters to be overridden for a subset of the threads. We treated each color specified in the weaving draft as a grouping of a separate ”yarn type”, with common settings. In this way, it is easy to control the reflection properties of certain threads, without affecting the others.

5. Results

In this section, we evaluate the ability of the shader to recreate the look of different types of fabrics. A comparison is made by considering two different physical cloth samples and comparing the results of the shader both with photographs of the samples, and with texture based V-Ray shaders for these fabrics that are used in production. We also make a small demonstration that shows a range of materials that can be reproduced using the shader, and finally, we make some basic measurements of the performance impact of our shader.

5.1. Appearance at High Magnification

Even though the shader is not meant to be used at close ranges, it is nonetheless useful to inspect it at high magnification in order to see its behaviour. Fig. 7 shows a schematic illustration of a piece of yarn. In the image, the individual filament threads are visible as they twist around the yarn. The distinct highlights that these filaments give are also visible. In the shader, however, the number of filaments is assumed to be infinite, which would give the cylinder a smooth surface. Fig. 8 shows a comparison between the specular highlight from the yarn segment in Fig. 7 and the same segment as represented by the shader.

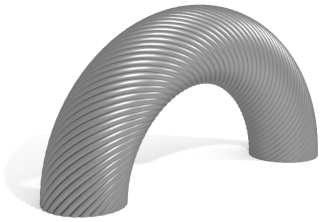
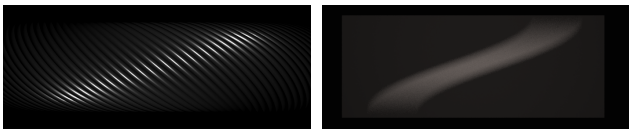


Figure 7: The geometry of a yarn segment, which is made up of twisted filament threads. Note the characteristic streak of specular reflections from the surface.



(a) Rendering of a yarn segment using explicit geometry.

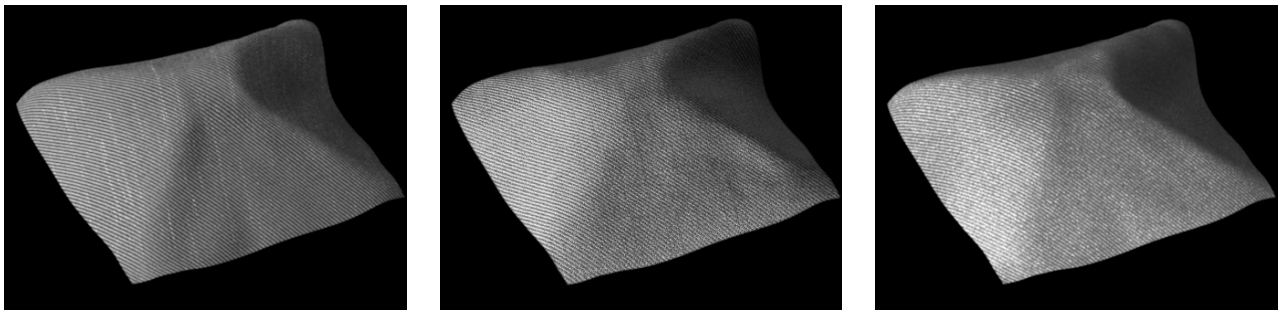
(b) Rendering of a yarn segment using the cloth shader

Figure 8: A comparison between *(a)* a rendered image using a geometrical representation of a yarn segment and *(b)* the cloth shader.

5.2. Comparison with Physical Cloth Samples

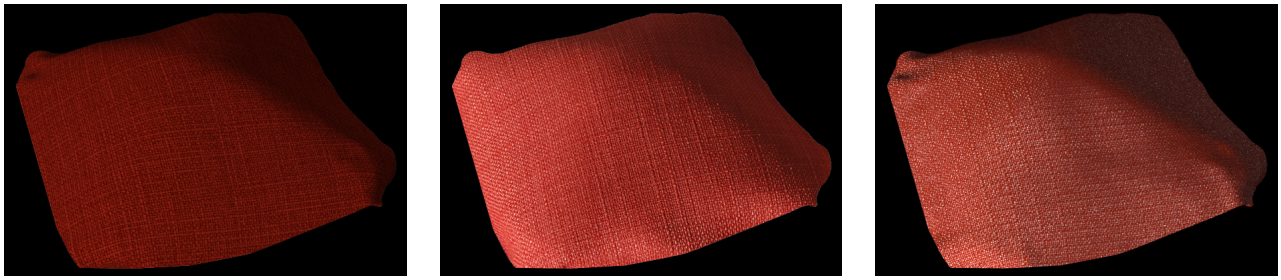
The work described in this paper was done in collaboration with the computer graphics department at IKEA Communications AB (ICOM), which is responsible for a large portion of the images in the IKEA catalogue. Since the images appear next to real photographs, the requirements for realism are high. Cloth is one of the more difficult surfaces to recreate in these images and a robust method for rendering textiles is needed. As mentioned before, many production environments use texture maps and a general purpose shader to reproduce the appearance of woven cloth, this is the method currently used in production at ICOM.

In order to perform comparisons, two physical samples of commercial fabrics were provided by ICOM, together with their corresponding texture based V-Ray shaders. Two sets of tests were performed to evaluate the visual results of our shader.



(a) Texture based shader. (b) Reference photograph. (c) Cloth shader.

Figure 9: A comparison of Fabric A with *(a)* the texture based shader currently used in production, *(b)* a reference photograph and *(c)* the cloth shader presented in this report. The highlight on the left side of the cloth is correctly reproduced by our shader, while it is completely neglected by the current production shader.



(a) Texture based shader. (b) Reference photograph. (c) Cloth shader.

Figure 10: A comparison of Fabric B with *(a)* the texture based shader currently used in production, *(b)* a reference photograph and *(c)* the cloth shader presented in this report. It is clear that the current production shader fails to correctly capture the reflections that appear in the reference photograph and in the cloth shader.

In the first test, the cloth sample was draped over a 3D printed template and photographed. This scene was then recreated as a 3D model. Using this model, images were rendered with both the texture based shaders and our cloth shader. Comparisons of the texture based shaders currently used in production, the physical sample, and the cloth shader developed in this paper are shown in Fig. 9 and Fig. 10.

Fabric A, shown in Fig. 9, is a twill weave fabric. The photograph shows a specular highlight on the left side of the fabric. This reflection is completely missing from the current production shader Fig. 9a. However, our cloth shader shown in Fig. 9c accurately simulates this phenomenon.

Fabric B, shown in Fig. 10, consists of a plain weave pattern that shows an interesting structure by having random variations in yarn sizes. An attempt at recreating these irregularities has been made for the production shader, and in our cloth shader (Fig. 10c). The variations in our cloth shader were accomplished by utilizing noise functions, and by manually altering the weaving pattern to add varying yarn sizes. Here our shader simulates specular reflections from the yarns, but not in a manner which closely matches the reference. The production shader almost exclusively shows diffuse behavior and neglects much of the specular behavior.

The second test consisted of photographing a flat sample of Fabric A with the camera positioned directly above the fabric. The sample was lit from 6 different angles. From this set of images a normal map was automatically generated which was used to displace a flat 3D mesh. This resulted in a 3D model of the cloth with the wrinkles of the fabric. The model was rendered using both our and the texture based shader. A comparison of the results can be seen in Fig. 11. Both our shader and the reference photographs show a much darker image at grazing angles, while the texture based shader has a much more uniform brightness. Furthermore, the texture based shader has less pronounced wrinkles, despite both sets of rendered images using the same geometry.

Range of Reproducible Materials. In order to test the expressiveness of the cloth shader, a few materials with different visual characteristics were recreated. In Fig. 1 on the first page, weaving drafts used as input to our shader are shown together with their final renders. These images are the result of different weaving patterns and combinations of parameters that can be changed through the exposed controls. While there are no physical samples to compare these results to, they still serve to illustrate the range of materials the shader is able to reproduce to a reasonable degree of realism.

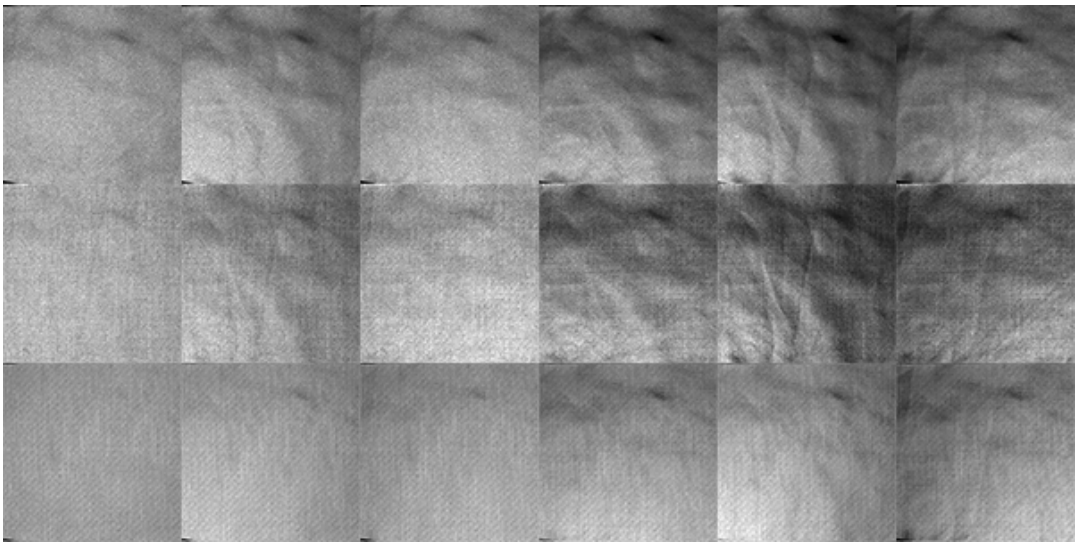


Figure 11: A comparison between our shader *(top row)*, reference photographs *(middle row)*, and the current production shader *(bottom row)*. The images show a sample of Fabric A lit from different angles.

5.3. Performance

We compared render times and memory usage for our shader and the texture based shader used in Fig. 11. The comparison is presented in Table 1. Rendering times and memory usage will naturally vary for both shaders depending on parameters and texture maps, but from these results we can expect ours to use much less memory and result in rendering times that comparable to, or even faster than, a texture based shader.

	Our	Production
Time (Adaptive Sampling)	44 s	1 min 17 s
Time (Fixed Sampling)	1 min 23 s	2 min 16 s
Memory (Total)	101 MB	470 MB
Memory (Shader)	1 MB	370 MB

Table 1: Render times and memory usage for our shader and the texture based production shader. Fixed sampling uses a predefined number of samples per pixel while adaptive sampling automatically adds more samples until the rendered image has converged sufficiently. The total memory includes all the memory used by V-Ray, while the shader memory is corrected to only include the memory used by the shader. In essence, it is the increase in memory usage compared to a default, untextured surface.

6. Discussion and Conclusions

From the results shown in Section 5, it can be seen that for some types of woven fabric, the cloth shader can produce results very close to the reference photograph. The shader seems best suited for reproducing fabrics with a small and regular pattern. An example of this is Fabric A, shown in Fig. 9. Here, the rendered result using our cloth shader is very similar to the reference photograph, showing highlighting features that appear to be completely absent in the shader currently being employed.

In Fig. 10 the results of trying to mimic the appearance of Fabric B are shown. This fabric features more large scale variation than Fabric A, both in terms of color and structure. The rendered result successfully captures reflections, highlights and other similarities to the reference photograph, but is still lacking in some areas. Fabric B suffers from the fact that the yarns in the model are defined on an evenly spaced rigid grid. In the result, some structural variation was added by manually altering the pattern to simulate a thicker yarn. However, as can be seen in Fig. 10c, this variation repeats too regularly.

The comparison made in Fig. 11, shows that our shader more accurately recreates the reflection properties of the cloth than the texture based approach. The reason is that our shader takes the significant specularity and anisotropic behavior of yarn into account. The texture based shader relies heavily on a diffuse term that gives it a behavior that is too uniform over the different directions of illumination.

Additionally, some attempts were made to recreate fabrics with different visual appearances. Fig. 1 shows examples of using the shader to simulate the appearance of tartan cloth, satin cloth and a jacquard fabric, respectively. This is by no means a rigorous evaluation of the realism of the result, but it illustrates the range of fabrics that the shader is capable of producing.

6.1. Further Work

The shader developed in this paper serves its purpose as a production ready tool, but there are many improvements that can be made. This section will present some interesting work that can be pursued in order to make the shader more expressive and easier to use.

Making the Shader More Flexible. From the experience of trying to match the look of Fabric B, it was concluded that the ability to modulate the width of the yarn randomly along each thread

would be an important addition. This is an important phenomenon that occurs in many fabrics and which currently is very difficult to recreate using the shader. Furthermore, the current implementation does not support translucency of the fabric. This means that the shader cannot be used in some scenarios, such as for rendering curtains. Implementing translucency would be a natural extension of the current model, and a logical next step in the development of the shader.

Finding a More Accurate Phase Function. The von Mises phase function used in the model for light scattering inside the yarn is a general model for forward scattering and while it gives results that look satisfactory, it has a vague physical grounding. It would be interesting to perform measurements on different types of yarn fibers to determine the shape of their phase functions. These types of measurements have been made for light scattering inside hair [MJC*03], which by now is a fairly well understood problem. However, the microscopic geometry of a yarn filament can vary significantly from that of hair, which means that the measurements made for hair are not directly applicable to cloth rendering.

Even more complexity is introduced when considering dyeing of yarn, which affects its scattering properties, or the fact that it is common to print patterns on the cloth, thereby adding a layer of paint to the filament and making the scattering even more complex.

6.2. Final Remarks

The rendered images created using the shader show realistic results for the right types of fabric and can be used in conjunction with professional tools commonly found in a production environment. It alleviates the hurdle of storing and managing a large library of high-resolution textures by instead having the shader generate the appearance of patterns from a simple text description. This also has the benefit of freeing up memory during rendering that would otherwise be used to store high-resolution textures.

Many of these benefits also carry through to the artists. The lighter data load gives them more immediate control over the appearance of the cloth, allowing them to quickly change parameters or patterns without having to modify large textures. While the shader can simulate and more accurately capture the look of woven fabrics, finding the combinations of parameters that achieve the desired look still requires artistic skill and gives the artists the final say.

An interesting use case for the shader outside the domain of image production is during development of new fabrics and weaving patterns. A preview of a pattern being designed can be rendered without having to create a prototype weave or having to wait for a sample to be sent from a factory, thereby greatly reducing the feedback loop during early textile development.

References

[Ada01] ADANUR S.: *Handbook of Weaving*. CRC Press, 2001. 3

[AMTF03] ADABALA N., MAGNENAT-THALMANN N., FEI G.: Visualization of woven cloth. In *Proceedings of the 14th Eurographics Symposium on Rendering* (2003), The Eurographics Association, pp. 178–186. 3

[GCP*09] GHOSH A., CHEN T., PEERS P., WILSON C. A., DEBEVEC P.: Estimating specular roughness and anisotropy from second order spherical gradient illumination. In *Proceedings of the Twentieth Eurographics Conference on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2009), EGSR'09, Eurographics Association, pp. 1161–1170. doi:10.1111/j.1467-8659.2009.01493.x. 2

[HDCD15] HEITZ E., DUPUY J., CRASSIN C., DACHSBACHER C.: The SGGX microflake distribution. *ACM Transactions on Graphics 34*, 4 (2015), 48:1–48:11. doi:10.1145/2766988. 3

[HK93] HANRAHAN P., KRUEGER W.: Reflection from layered surfaces due to subsurface scattering. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1993), SIGGRAPH '93, ACM, pp. 165–174. doi:10.1145/166117.166139. 4

[IM12] IRAWAN P., MARSCHNER S.: Specular reflection from woven cloth. *ACM Transactions on Graphics 31*, 20 (2012), 1–20. doi:10.1145/2077341.2077352. 2, 3, 4

[Ira08] IRAWAN P.: *Appearance of woven cloth*. PhD thesis, Cornell University, 2008. URL: http://core.kmi.open.ac.uk/download/pdf/4907382.pdf. 2, 4

[IS06] IRAWAN P., STEPHEN R M.: *A Simple , Accurate Texture Model for Woven Cotton Cloth*. Tech. Rep. June, Cornell University Program of Computer Graphics, 2006. 2

[JAM*10] JAKOB W., ARBREE A., MOON J. T., BALA K., MARSCHNER S.: A radiative transfer framework for rendering materials with anisotropic structure. *ACM Transactions on Graphics 29*, 4 (2010), 1. doi:10.1145/1833351.1778790. 3

[KSZ*15] KHUNGURN P., SCHROEDER D., ZHAO S., BALA K., MARSCHNER S.: Matching Real Fabrics with Micro-Appearance Models. *ACM Trans. Graph. 35*, 1 (2015), 1:1—-26. 3

[MJC*03] MARSCHNER S. R., JENSEN H. W., CAMMARANO M., WORLEY S., HANRAHAN P.: Light scattering from human hair fibers. *ACM Trans. Graph. 22*, 3 (July 2003), 780–791. doi:10.1145/882262.882345. 8

[Nie97] NIELSEN R.: Wif weaving information file version 1., 1997. URL: http://www.mhsoft.com/wif/wif.html. 4

[PH10] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory To Implementation*, 2 ed. Morgan Kaufmann, Burlington, Massachusetts, 2010. 4

[Shu03] SHUM Y. C. . S. L. . H. Z. . Y.-Q. X. . B. G. . H.-Y.: Realistic rendering and animation of knitwear. *IEEE Transaction on Visualization and Computer Graphics* (2003). 3

[SZZ12] SCHRÖDER K., ZHAO S., ZINKE A.: Recent Advances in Physically-Based Appearance Modeling of Cloth. *SIGGRAPH Asia 2012 Courses* (2012), 12:1–12:52. doi:10.1145/2407783.2407795. 3

[ZJMB12] ZHAO S., JAKOB W., MARSCHNER S., BALA K.: Structure-aware synthesis for predictive woven fabric appearance. *ACM Trans. Graph. 31*, 4 (July 2012), 75:1–75:10. URL: http://doi.acm.org/10.1145/2185520.2185571, doi:10.1145/2185520.2185571. 3

[ZLB16] ZHAO S., LUAN F., BALA K.: Fitting procedural yarn models for realistic cloth rendering. *ACM Trans. Graph. 35*, 4 (July 2016), 51:1–51:11. doi:10.1145/2897824.2925932. 3