Task 4. Cloth Rendering

Garoe Dorta-Perez
CM50245: Computer Animation and Games II

May 15, 2015

1 Introduction

Rendering realistic images is a challenging task, specially if there are memory or time constrains for the computation. Cloth is a complex material composed of interwoven threads of different types. Moreover, its appearance can vary from diffuse to highly specular.

2 Previous work

Several methods have been proposed to render cloth fabrics efficiently and realistically. One of the earliest approaches was based on simple empirical shading models [7]. The main objective was to accomplish believe shading, disregarding physical accuracy.

The methods can be broadly divided into three groups, data based models, geometric models and volumetric models.

The data based approach focuses on collecting reflectance information, that will be later used to model the cloth. Bidirectional Texture Function (BTF) [1] is a function that is often used to sampled in the data based techniques.

Geometric models focus on simulation the micro-geometry of the cloth in conjunction with global illumination. The light scattering is simulated for each fibre in the thread, where the fibres are modelled as perfect cylinders. To be able to model the the complete scattering effects on a surface, Bidirectional Scattering-Surface Reflectance Distribution Function (BSSRDF) [3] have been used. With this function, complex light phenomena like subsurface scattering are modelled.

3 Methodology

We have chosen to implement a recent paper [4] which presents a microcylinder based model for fast and realistic cloth rendering. The authors propose a model of fabric based on two microcylinders oriented in two orthogonal directions, as shown in Figure 1. Before we begin lets define a normalized vector $\mathbf{s} = \mathbf{n} \times \mathbf{t}$, where $\times$ is the vector cross product operator. With the vectors $\{\mathbf{t}, \mathbf{n}, \mathbf{s}\}$ we can define a Cartesian coordinate system, whose centre will be a triangle ray-intersection point $\mathbf{p}$. Having all vectors defined with respect to this coordinate system will allow to compute more easily the quantities that will be introduced below. The transformation matrix that will transform from world coordinates to this system using column vectors with homogeneous coordinates is

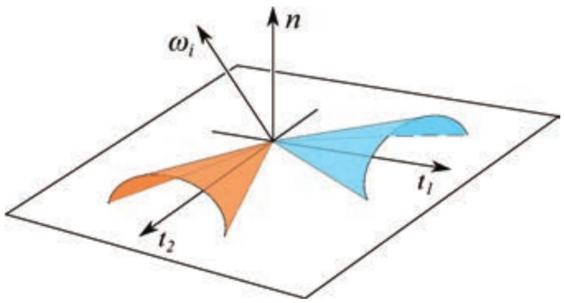

Figure 1: Sadeghi et al. shading model [4], where $\boldsymbol{\omega_i}$ is the incident light direction, $\mathbf{n}$ is the surface normal and $\mathbf{t_1}, \mathbf{t_2}$ are the orthogonal thread directions.

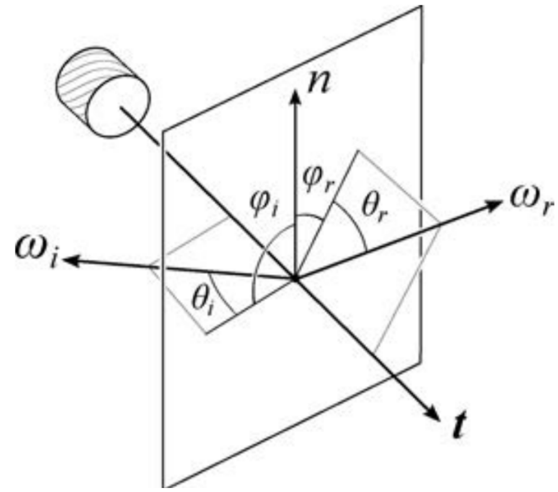

Figure 2: Sadeghi et al. shading model [4], showing θ and ϕ angles given a pair of directions $\boldsymbol{\omega_i}$ and $\boldsymbol{\omega_r}$.

$$M = M_{trans} M_{rot} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -p_x & -p_y & -p_z & 1 \end{pmatrix} \begin{pmatrix} t_x & n_x & s_x & 0 \\ t_y & n_y & s_y & 0 \\ t_z & n_z & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

The reflectance of a single cylinder is defined as

$$L_r = \int \frac{\left(f_{r,s}(\mathbf{t}, \boldsymbol{\omega_i}, \boldsymbol{\omega_r}) + f_{r,v}(\mathbf{t}, \boldsymbol{\omega_i}, \boldsymbol{\omega_r})\right) L_i(\boldsymbol{\omega_i}) cos(\theta_i) \delta \boldsymbol{\omega_i}}{cos^2(\theta_d)}, \tag{2}$$

where $\mathbf{t}$ is the thread direction, $\boldsymbol{\omega_i}$ is the ray incoming direction, $\boldsymbol{\omega_r}$ is the ray outgoing direction, θ_i, θ_r, ϕ_i and ϕ_r are angles as shown in Figure 2, $\theta_d = \theta_i - \theta_r$ and L_i is the incoming irradiance in the evaluated point. Note that radiometric notation [2] is used to define L_r, which represents the outgoing radiance over a infinitesimal arc length of the cylinder and how the integral extends over the entire sphere instead of the typical hemisphere.

The surface reflection term in Equation 2 is defined as

$$f_{r,s}(\mathbf{t}, \boldsymbol{\omega_i}, \boldsymbol{\omega_r}) = F_r(\eta, \boldsymbol{\omega_i}) cos(\phi_d/2) g(\gamma_s, \theta_h), \tag{3}$$

where $\theta_h = (\theta_i + \theta_r)/2$, $\phi_d = \phi_i - \phi_r$, F_r is a Fresnel reflection term that is computed using Schlick's approximation [5] $F_r(\eta, \boldsymbol{\omega_i}) = \eta + (1-\eta)(1 - \mathbf{h} \cdot \boldsymbol{\omega_i})^5$ where $\cdot$ is the vector dot product operator, $\mathbf{h} = (\boldsymbol{\omega_i} + \boldsymbol{\omega_r}) / \left|\boldsymbol{\omega_i} + \boldsymbol{\omega_r}\right|$ is the normalized halfway vector, η is the reflectance for $\mathbf{h} \cdot \boldsymbol{\omega_i} = 1$, $g(\gamma, \theta) = \gamma e^{\mathbf{v} \cdot \mathbf{p} - 1}$ is a Gaussian lobe [6] where $\mathbf{p}$ is the lobe axis, the direction $\mathbf{v}$ is the spherical parameter in the resulting function and γ is the amplitude.

The volume scattering term in Equation 2 is defined as

$$f_{r,v}(\mathbf{t}, \boldsymbol{\omega_i}, \boldsymbol{\omega_r}) = F_t(\eta, \boldsymbol{\omega_i}) F_t(\eta', \boldsymbol{\omega_r'}) \frac{(1-k_d)g(\gamma_v, \theta_h) + k_d}{cos(\theta_i) + cos(\theta_r)} \mathbf{k_a}, \quad (4)$$

where k_d is a scattering constant, $\mathbf{k_a}$ is an rgb albedo constant vector, $F_t = 1 - F_r$ is a Fresnel transmission term, $\boldsymbol{\omega_r'}$ is a projection of $\boldsymbol{\omega_r}$ into a plane that contains the normal $\mathbf{n}$ and η' is computed using the Bravais index [2] as $\eta'(m) = \sqrt{\eta^2 - sin^2(m)}/cos(m)$ where m is the angle between $\boldsymbol{\omega_r}$ and its projection $\boldsymbol{\omega_r'}$.

The outgoing radiance in the path shown in Figure 2 is

$$L_r(\boldsymbol{\omega_r}) = a_1 L_{r,1}(\boldsymbol{\omega_r}) + a_2 L_{r,2}(\boldsymbol{\omega_r}), \quad (5)$$

where a_1 and a_2 are the area coverage ratio for the first and second thread within the patch, in our case we assume a watertight pattern of equally size threads, leading to $a_1 = a_2 = 0.5$, L_1 and l_2 are the outgoing radiances for the first and second thread computed as shown in Equation 2.

In order to compute a thread direction $\mathbf{t}$ without using external data structures, we follow a fixed texture axis as shown in Figure 3. Given an intersection point in a triangle $\mathbf{p} = [x_p, y_p, z_p]$ and its texture coordinates $\mathbf{u} = [u_p, v_p]$. We define $\mathbf{t}$ vector as the vector $\mathbf{t} = (\hat{\mathbf{p}} - \mathbf{p})/\,|\hat{\mathbf{p}} + \mathbf{p}|$ such that $\hat{\mathbf{p}}$ texture coordinates are $\hat{\mathbf{u}} = [u_p + 1, v_p]$. In our shader we can easily compute the texture coordinates of the triangle vertices, however we can not immediately get a world position from a new texture coordinate. Assuming that the 3d to texture transformation is an affine matrix T,

$$\mathbf{u}T = \mathbf{p} \rightarrow \begin{pmatrix} u & v \end{pmatrix} \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} = \begin{pmatrix} x & y & z \end{pmatrix} \quad (6)$$

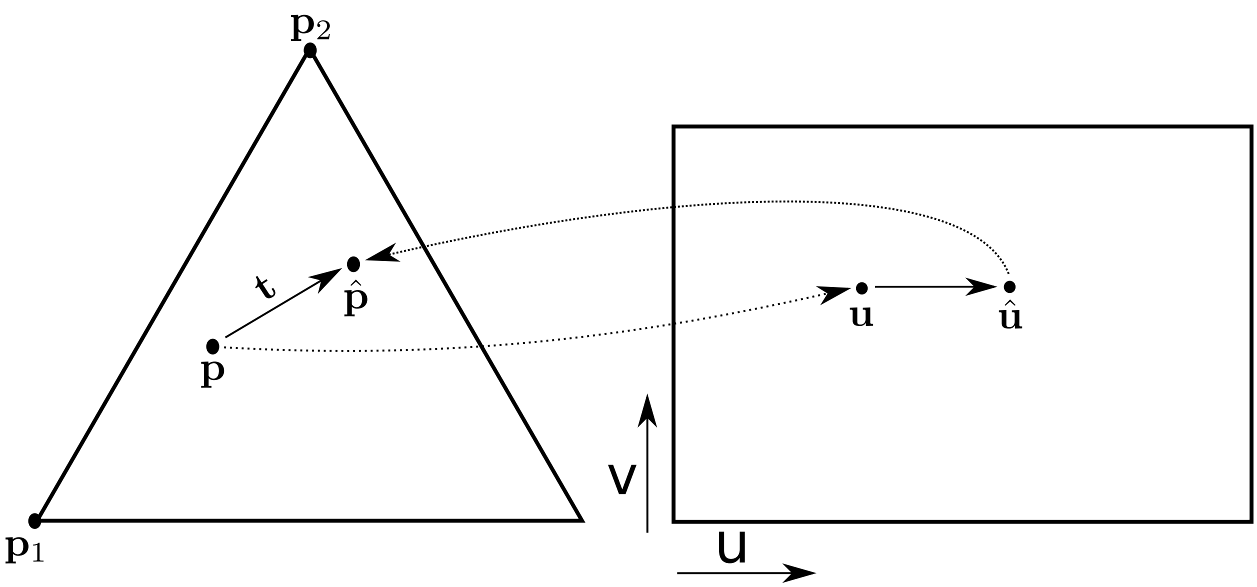

Figure 3: To compute the vector $\mathbf{t}$, we follow the texture coordinates of the intersection point $\mathbf{p}$.

The system in Equation 6 has six unknowns and three equations, therefore we need two pairs of world points - texture coordinates to solve T. Writing out the terms for two known points $\mathbf{p}_1$ and $\mathbf{p}_2$ in the triangle,

$$\begin{aligned} x_1 &= u_1 a + v_1 d, \quad y_1 = u_1 b + v_1 e, \quad z_1 = u_1 c + v_1 f, \\ x_2 &= u_2 a + v_2 d, \quad y_2 = u_2 b + v_2 e, \quad z_2 = u_2 c + v_2 f. \end{aligned} \tag{7}$$

Solving for each term analytically,

$$d = \frac{u_2 x_1 - u_1 x_2}{u_2 v_1 - u_1 v_2}, \qquad e = \frac{u_2 y_1 - u_1 y_2}{u_2 v_1 - u_1 v_2}, \qquad f = \frac{u_2 z_1 - u_1 z_2}{u_2 v_1 - u_1 v_2}, \tag{8}$$

$$a = \frac{x_1 - v_1 d}{u_1}, \qquad b = \frac{y_1 - v_1 e}{u_1}, \qquad c = \frac{z_1 - v_1 f}{u_1}. \tag{9}$$

Notice that the term of $1/(u_2 v_1 - u_1 v_2)$ is shared, so it can be precomputed in order to increase performance. The aforementioned method will give $\mathbf{t}_1$, for $\mathbf{t}_2$ the process is equivalent with the exception that $\hat{\mathbf{u}}$ will be incremented on the v direction, $\hat{\mathbf{u}} = [u_p, v_p + 1]$.

4 Results

5 Conclusion and Future Work

References

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

[2] Stephen R. Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. Light scattering from human hair fibers. *ACM Transactions on Graphics*, 22 (3):780–791, 2003.

[3] Fred E. Nicodemus, Joseph C. Richmond, Jack J. Hsia, Irving W. Ginsberg, and Thomas Limperis. *Geometrical considerations and nomenclature for reflectance*. National Bureau of Standards Washington, DC, USA, 1977.

[4] Iman Sadeghi, Oleg Bisker, Joachim de Deken, and Henrik Wann Jensen. A practical microcylinder appearance model for cloth rendering. *ACM Transactions on Graphics*, 32 (2):1–12, 2013.

[5] Christophe Schlick. An Inexpensive BRDF Model for Phsyically-based Rendering. In *Computer Graphics Forum*, volume 13, pages 233–246, 1994.

[6] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-Frequency Rendering of Dynamic, Spatially-Varying Reflectance. *ACM Transactions on Graphics*, 28(5):133:1–133:10, 2009.

[7] Jerry Weft and Murray Hill. The Synthesis of Cloth Objects. *SIGGRAPH Comput. Graph.*, 20(4):49–54, 1986.