

IDSS: A Novel Representation for Woven Fabrics

Jiahua Zhang, George Baciu, Dejun Zheng, Cheng Liang, Guiqing Li, and Jinlian Hu

Abstract—The appearance of woven fabrics is intrinsically determined by the geometric details of their meso/micro scale structure. In this paper, we propose a multiscale representation and tessellation approach for woven fabrics. We extend the Displaced Subdivision Surface (DSS) to a representation named Interlaced/Intertwisted Displacement Subdivision Surface (IDSS). IDSS maps the geometric detail, scale by scale, onto a ternary interpolatory subdivision surface that is approximated by Bezier patches. This approach is designed for woven fabric rendering on DX11 GPUs. We introduce the Woven Patch, a structure based on DirectX's new primitive, patch, to describe an area of a woven fabric so that it can be easily implemented in the graphics pipeline using a hull shader, a tessellator and a domain shader. We can render a woven piece of fabric at 25 frames per second on a low-performance NVIDIA 8400 MG mobile GPU. This allows for large-scale representations of woven fabrics that maintain the geometric variances of real yarn and fiber.

Index Terms—Woven fabric, subdivision surface, interlaced displacement, intertwined displacement, tessellation, GPU

1 INTRODUCTION

A wide and diverse range of applications, from 3D games to material analysis and textile design, require interactive rendering of complex geometric details of woven fabrics. In particular, woven structures and fiber distributions are better visualized at a variety of scales while incurring the smallest possible computational burden. This can be accomplished by using Level-of-Detail (LoD) management, a process of simplification in which the levels of detail are rendered throughout multiple viewing ranges such as macro, meso, and micro scales. The LoD management makes use of synthetic representations, which can be employed simultaneously, essentially balancing workload and visual quality.

1.1 Motivation and Challenges

The main objective of this work is to explore and identify more accurate geometric representations for highly complex structures found in woven fabrics. In our work, we focus on fiber and yarn representations and offer a new framework for multiscale rendering of complex fabric surfaces.

Woven fabrics constitute the primary materials of casual and formal garments from common shirts, suits, and dresses to silk ties, attachments and accessories as well as decorative covers for interior design. As such, woven

fabrics are a source of rich and complex texture that is often defined by simple elements found at the level of fiber or yarn interlacing structure.

Furthermore, the technology and performance of weaving machinery have drastically improved the quality and density of woven fabrics. Hence, the requirements for greater verisimilitude in the woven details of fabrics for all kinds of applications, e.g., virtual design, structural simulation, and computer games. The design of woven fabrics for garment panels, generating realistic woven patterns in a 3D environment, is a functionality that is currently missing in all single scale fabric texture design systems used in the textile industry. This creates the need to visualize the scale-varying appearance of fabrics for virtual clothing and multiscale rendering at interactive rates.

Fig. 1 shows three scales at which woven fabrics are commonly viewed. At the macro or panel scale, Fig. 1a, the cloth appears smooth. At the meso or yarn scale, Fig. 1b, we can see the weave of the yarn. At the micro or fiber scale, Fig. 1c, we can see the construction of the yarn itself, made up of intertwined strands.

Displaced Subdivision Surfaces (DSS) [1] provide an effective way to represent detailed surfaces. A DSS patch defines both the surface topology and parameterization by using a coarse control mesh and also captures the fine geometric detail using a scalar-valued function. The DSS representation also permits geometry separation by providing smooth domain surface and detail displacement fields. However, this is not sufficient to represent a detailed surface with scale-varying appearance such as woven fabrics.

1.2 Contributions

In this paper, we propose a new representation called the Intertwisted Displacement Subdivision Surface (IDSS), a unified multiscale representation for representing the scale-varying appearances of woven fabrics at varying tessellation levels. Our contributions are summarized as follows:

- The “interlaced intertwined displacement” operator in conjunction with the “Bezier approximation of

• J. Zhang, G. Baciu, D. Zheng, and C. Liang are with the GAMA Lab, PQ724, Dept. Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China. E-mail: {newzjh, edwardchina, alieliang85}@126.com, csgorge@comp.polyu.edu.hk.
 • G. Li is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou Higher Education Mega Center, Guangzhou, 510006, China. E-mail: ligq@scut.edu.cn.
 • J. Hu is with QT807a, Institute of Textiles and Clothing, Q Core, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. E-mail: tchujl@inet.polyu.edu.hk.

Manuscript received 8 Feb. 2011; revised 22 Sept. 2011; accepted 8 Oct. 2011; published online 17 Feb. 2012.

Recommended for acceptance by H. Pottmann.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2011-02-0026. Digital Object Identifier no. 10.1109/TVCG.2012.66.

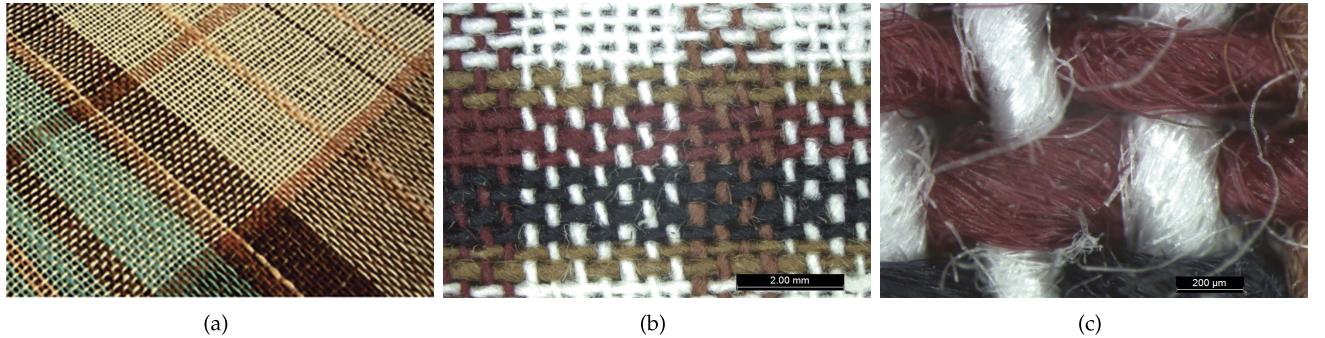


Fig. 1. The natural appearance of woven fabric at three scales. (a) Panel scale. (b) Yarn scale. (c) Fiber scale.

ternary interpolatory subdivision surface.” These new mechanisms allow the realistic reproduction and rendering of the interlaced and intertwined effects of woven fabrics.

- A synthesized expression with geomorphing (19) for a complete scale-varying geometric representation. This makes it possible to view woven fabrics, their weaving patterns and actual fiber distributions simultaneously at different scales.
- Customized geometric models at multiple scales. We can thus choose to employ bicubic Bezier approximation of ternary subdivision at the fabric panel scale, swept surfaces at the yarn scale, and intertwined surfaces at the fiber scale. This allows the independent editing of each scale.
- More elaborate topology and structure. Unlike multiresolution or progressive mesh techniques, in which different resolutions share the same topology by varying only the number of triangles and vertices, the proposed IDSS representation allows coarse geometry to be replaced with finer geometry.

1.3 Organization

The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 describes IDSS Formulation. Section 4 describes the customized geometric models that the IDSS uses to represent woven fabrics. Section 5 describes GPU tessellation and geomorphing for woven fabrics. Section 6 describes the implementation of the proposed representation on current GPUs. Section 7 describes our experiments. Section 8 draws conclusion.

2 RELATED WORK

The main requirement for multiple-view scalability is that it should allow smooth, continuous changes between views for scale-varying appearances of woven fabrics. Ultimately, the quality of rendering is determined by the method used to synthesize the scale-varying geometries and the computational efficiency of the tessellation method.

Well-known image-based techniques, such as 2D textures [4], bump mapping [5], and parallax mapping [6] emphasize the pixel color rather than the geometric position of the surface. Popular geometry-based approaches to the modeling of details at varying scales include displacement mapping [7], volume-textured surfaces [8] and shell maps [9]. Lee et al. [1] provide a solution that maps displacement

onto subdivision surfaces. More recent approaches to obtain synthetic geometries include per-pixel displacement mapping [10], relief mapping [11], and view dependent displacement mapping [12], [13]. The drawback of these mapping methods is that they cannot dynamically adapt to physical parameters such as yarn thickness, gap width between yarn and the twist configuration of fibers. This is because they cannot regenerate the displacement map, bump map or geometry detail map in real time. Musalski et al. [14] addressed this by proposing multiresolution geometric detail on subdivision surfaces as a way to render different resolution displacement maps in an image pyramid. However, it is difficult to represent the geometric detail associated with scale-varying topologies.

A popular way to render high order surfaces is to apply adaptive tessellation for workload balancing [15], [16], [17], [18], [19], [20]. The most popular way to achieve this on DX11 GPUs is to apply Bezier approximation in the tessellator. Loop and Schaefer proposed the ACC (bicubic Bezier approximation of Catmull-Clark subdivision surfaces) [22], Li et al. proposed the ALSS (Bezier approximation of Loop subdivision surfaces) [23], and Dudash proposed the approximation of displaced subdivision surfaces [3].

3 IDSS FORMULATION

The proposed IDSS is a multiscale representation of scale-varying geometries and scale-varying topologies for woven fabrics. It extends the DSS in that it facilitates interlaced or intertwined vector displacements and multiscale transformation synthesis.

3.1 IDSS Derivation

The conventional DSS [1] maps the normal displacement onto a domain surface to produce a high level of detail surface while maintaining the coarse control mesh. However, the DSS is displaced along the normal direction. Therefore, it cannot represent the complex interlaced, intertwined, and screwed effects of woven fabrics. An IDSS facilitates this by extending the DSS with 3D displacement vectors. The formulation takes the form of a matrix transformation that supports vector displacement mapping at the i th scale. Therefore, we first transform the detail displacement, a 3D offset vector d_i by using the Darboux frame M_i , and then add it to the local position l_i in

the domain surface at i th scale to achieve a detailed surface position p_i

$$p_i = l_i + M_i d_i, \quad (1)$$

where l_i is a local position in the domain surface at i th scale. M_i is the Darboux frame at i th scale, a 3×3 unit base matrix constructed at the local position l_i . The 3D displacement vector d_i indicates the three displacement offsets along the three directions of the Darboux frame, respectively.

As the displaced position p_{i+1} at the detailed scale, the $(i+1)$ th scale, is defined in the coordinate space of the Darboux frame corresponding to l_i at i th scale, the displacement vector d_i at i th scale is equal to the displaced position p_{i+1} at $(i+1)$ th scale. The displacement vector d_{i+1} at $(i+1)$ scale is equal to the displaced position p_{i+2} at a higher detail scale, the $(i+2)$ th scale. These vectors link the i th scale and $(i+1)$ th scale

$$d_i = p_{i+1}. \quad (2)$$

By merging (2) into (1) and setting the displaced position p_n as the local position l_n at the final scale of detail, the n th scale, we have a synthesized multiscale expression as follows:

$$p_i = l_i + M_i l_{i+1} + M_i M_{i+1} l_{i+2} + \dots. \quad (3)$$

Equation (3) represents the multiscale expression by synthesizing two of the multiple scales. We transform the local position of all the other scales to the i th scale. The local positions from all the other scales are transformed by (3). Equation (3) can be refined as an expression for all the $n-i+1$ scales

$$\begin{aligned} p_i &= \beta_i l_i + \beta_{i+1} \left(\prod_{k=i}^i M_k \right) l_{i+1} + \dots \\ &+ \beta_j \left(\prod_{k=i}^{j-1} M_k \right) l_j + \dots + \beta_n \left(\prod_{k=i}^{n-1} M_k \right) l_n. \end{aligned} \quad (4)$$

We introduce the factors β_i for smooth transition between frames at i th scale. They also represent the measurements of a number of details from each scale. These factors will be addressed in Section 5.2.

3.2 IDSS Application

In the representation of textile fabrics, the IDSS produces a complete range of representations by synthesizing scale-varying geometries of any two of multiple scales. The IDSS is applied to map interlaced displacements from the yarn scale onto a subdivision surface at the panel scale and map intertwined displacements from the fiber scale onto a swept surface at the yarn scale. The domain surface at the panel scale is a subdivision surface. Therefore, in the weft case for woven fabrics, (4) is parameterized as follows:

$$\begin{aligned} p_p &= \beta_p l_p(u, v) + \beta_y M_p(u, v) l_y(u) \\ &+ \beta_f M_p(u, v) M_y(u) l_f(u) \\ &+ \beta_s M_p(u, v) M_y(u) M_f(u) l_s(v). \end{aligned} \quad (5)$$

Equation (5) consists of four parts: the panel-scale term l_p is the domain surface, the ternary subdivision surface; the

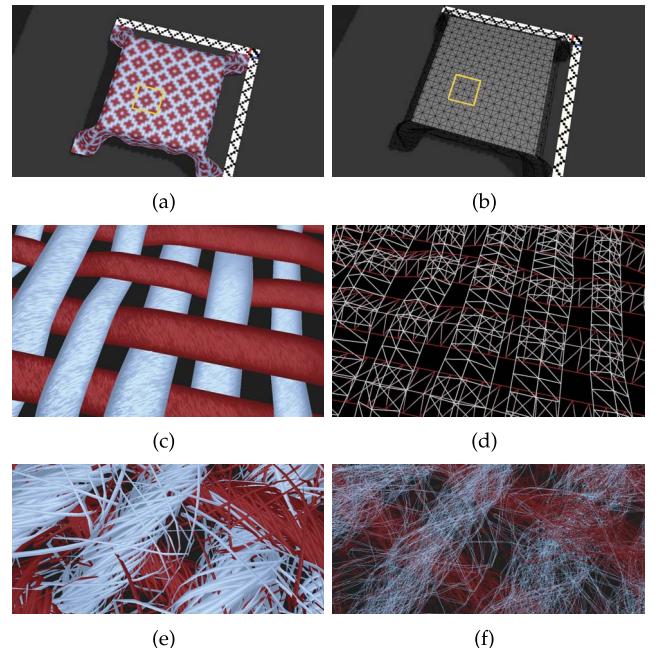


Fig. 2. A woven fabric and a woven-patch (region within the yellow boundary) rendered at three scales: (a) and (b) At the panel scale. (c) and (d) At the yarn scale. (e) and (f) At the fiber scale. (a), (c), and (e) solid rendering. (b), (d), and (f) wireframe rendering.

yarn-scale term $M_p l_y$ is the interlaced displacement transformed into the panel scale space; the fiber scale term $M_p M_y l_f$ is the intertwined displacement transformed into the panel scale space; the shell scale term $M_p M_y M_f l_s$ is the shell position of a fiber transformed into the panel scale space.

At the panel scale, the local position $l_p(u, v)$ is the limit position of a subdivision surface. The transformation $M_p(u, v)$ for the next scale is a 3×3 unit base matrix constructed using a current-scale Darboux frame (τ_p, n_p, b_p) . At the yarn scale, the local position $l_y(u)$ is the parametrization of the yarn path using variable u in the prescale Darboux frame. The transformation $M_y(u)$ for the next scale is a 3×3 unit base matrix constructed using a current-scale Darboux frame (τ_y, n_y, b_y) . At the fiber scale, the local position $l_f(u)$ is the parametrization of migrating fiber path using variable u in the prescale Darboux frame. The transformation $M_f(u)$ for the next scale is a 3×3 unit base matrix constructed using a current-scale Darboux frame (τ_f, n_f, b_f) . At the final scale, the local position $l_s(v)$ is the parametrization of the shell using variable v in the prescale Darboux frame.

Fig. 2 shows a woven fabric rendered using IDSS at three scales. We rendered the left column using solid shading and the right column in wireframe. Figs. 2a and 2b show a woven fabric rendered as a subdivision surface distant from the viewpoint. As the camera approaches the woven fabric, Figs. 2c and 2d swept geometries of the yarn are generated to illustrate the mesostructures of the woven fabric. Up close, Figs. 2e and 2f it is possible to clearly see intertwined fibers in a yarn.

3.3 IDSS Partition

The surface constructed by (5) is a complete panel surface that cannot be used directly on current GPUs for parallel tessellation. We need to divide it into patches for parallel

processing. During the partitioning process, we need to match the support nodes of the neighboring patches.

Like ACC and ALSS in [22] and [23], we approximate the patches of an IDSS in Bezier form, rather than B-spline form. The masks from neighbor patches are converted as masks inside the patch so that the surface of the patch can be computed by fetching only the local Bezier control points inside the patch. We call the area described in Fig. 2 a woven patch, an area of an IDSS represented in Bezier form. It can also be described as an atom primitive for GPU parallel tessellation.

Woven patches are represented as scale-varying geometries at three scales. Each woven patch is represented as a bicubic Bezier patch at the panel scale, defined by its control net according to (8), interlaced swept segments, defined by nodes according to (12), at the yarn scale and intertwined helices, according to (14), at the fiber scale. The woven patches are the basic primitives for parallel tessellation on DX11 GPUs. They are also the instances for parallel tessellation on current GPUs, which enables us to semi-uniformly tessellate a woven fabric patch-by-patch.

4 GEOMETRIC MODELS OF IDSS

In this section, we describe the geometric models of IDSSs that we use to represent woven fabrics: Bicubic Bezier approximation of ternary interpolatory subdivisions at the panel scale; Swept surfaces at the yarn scale; Intertwisted migrating surfaces at the fiber scale.

4.1 The Geometric Model at the Panel Scale: The Bicubic Bezier Approximation of Ternary Interpolatory Subdivisions

4.1.1 Ternary Interpolatory Subdivision

Approximate subdivision and interpolatory subdivision are two of the most currently researched types of schemes for surface representation. They can achieve smooth limit surfaces from coarse control meshes. Approximate subdivision schemes such as the Doo-Sabin [24] and Catmull-Clark and [25] usually result in shrinkage. However, for interpolatory subdivision schemes the limit surfaces interpolate the control meshes. The interpolatory feature is an important attribute for representation of woven fabric at the panel scale. This is why we choose to employ interpolatory subdivision.

Among all interpolatory subdivision schemes, we choose the ternary interpolatory subdivision [26], [27], [28], [29]. The ternary subdivisions can facilitate parametric approximation based on bicubic Bezier patches, because the first refinement of a ternary subdivision patch results in a 4×4 control net. This control net is coincident with that of a bicubic Bezier patch. For the scheme we used in our implementation, we refer the reader to [26]. We follow this formulation to provide a complete treatment of boundary regions, while the interpolatory surface is C^2 continuous.

4.1.2 Bicubic Bezier Approximation

A ternary interpolatory subdivision surface cannot be directly reformulated into Bezier or B-Spline form, however bicubic Bezier patches can be used to approximate it. Applying interpolatory subdivision, the vertices inserted at

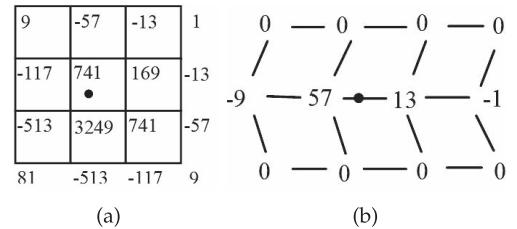


Fig. 3. Regular masks for Bezier approximation construction of ternary interpolatory subdivision surface. (a) Mask for interior vertex. (b) Mask for edge vertex.

each refinement step are located on the limit surface. Unlike approaches described in [22] and [23], we applied bicubic Bezier approximation to ternary interpolatory subdivision surfaces rather than Loop subdivision or Catmull-Clark subdivision surfaces, as in regular cases. We let the bicubic Bezier surface interpolate all inserted points in the first refinement of ternary subdivision, of the form:

$$M_R R = M_Q Q, \quad (6)$$

where M_R (see Appendix A, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TVCG.2012.66>) is a 16×16 ternary interpolatory subdivision matrix and it is extracted from Hassan et al.'s formulation (by setting the parameter $\mu = 4/45$ in that formula where $4/45$ is the median value in the range which ensures the surface is C^2) [26]. R is the set of 16 handle vertices (1 vertex on each of the 4 corners of the woven patch, 12 vertices surrounding the woven patch on the boundary of same-scale neighboring patches). M_Q (see Appendix B, which is available in the online supplemental material) is a 16×16 Bernstein basis matrix extracted at the corresponding parametric position. Q is the set of the 16 Bezier control vertices for the patch. The $M_R R$ is the limit positions of the ternary interpolatory subdivision surface corresponding to the 16 parametric positions. The solution of (6) is

$$Q = M_Q^{-1} M_R R. \quad (7)$$

Equation (7) can be expressed as three masks in the regular case. Fig. 3 describes the masks of the interior points and edge points. The vertex point is exactly equal to the limit point in corner cases.

We use (u, v) to express the parametric position in a woven fabric, where $u \in [0, m_w]$ and $v \in [0, m_h]$. m_w is the width and m_h is the height of a woven fabric. $\{u\}$ is the fractional part of the parameter u and $\{v\}$ is the fractional part of the parameter v . The entire bicubic Bezier patch is thus represented as

$$l_p(u, v) = \sum \sum q_{i,j} Y_i(\{u\}) Y_j(\{v\}), \quad (8)$$

where $Y_i(\{u\})$ and $Y_j(\{v\})$ are the Bernstein bases. Apart from the geometric patch, the panel space needs to be established, as defined by the Darboux frame $(\tau_p(u, v), n_p(u, v), b_p(u, v))$ at parametric position (u, v) . The tangent can be a Bezier combination of discrete control tangents. Therefore, the discrete control tangents are calculated by applying tangent masks on corner vertices of the original mesh and on the vertices of first refinement, such that

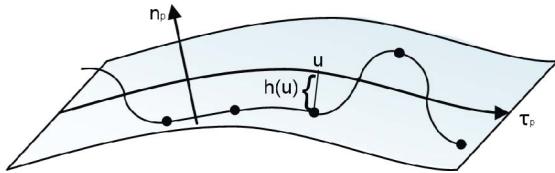


Fig. 4. Path of weft yarn over panel as displacement. The panel is defined in world space and the yarn is defined in panel space. The black dots are the interlacing nodes.

$$\begin{aligned} M_\tau M_R R &= M_Q T \\ \Rightarrow T &= M_Q^{-1} M_\tau M_R R, \end{aligned} \quad (9)$$

where M_τ is the matrix extracting the tangents from the subdivision surface after one refinement. From the solution of (9), $T = \{\tau_{i,j}\}$ is the set of 16 control tangents used to form the Bezier patch. Similar to the calculation of control tangents, we calculate the set $B = \{b_{i,j}\}$ of 16 control binormals.

The tangent, normal and binormal of the panel-space Darboux frame (τ_p, n_p, b_p) at parametric position (u, v) are calculated as

$$\begin{aligned} \tau_p(u, v) &= \sum \sum \tau_{i,j} Y_i(\{u\}) Y_j(\{v\}), \\ n_p(u, v) &= \tau_p(u, v) \times b_p(u, v), \\ b_p(u, v) &= \sum \sum b_{i,j} Y_i(\{u\}) Y_j(\{v\}). \end{aligned} \quad (10)$$

Based on (10), we establish the panel space.

4.2 The Geometric Model at the Yarn Scale: The Yarn Path from Weaving Pattern

At this scale, a yarn is represented as a swept surface. This is obtained when a curve, termed the yarn path, is surrounded by a shell. The formulation of the shell will be addressed in Section 4.4. The yarn path, shown in Fig. 4, is obtained using a cosine curve to interpolate interlaced nodes. These interlaced nodes are up and down along a yarn over the panel surface indicating local displacement from the panel surface along the panel normal. In a single intersection between a warp yarn and a weft yarn there are two types of interlaced nodes, up for weft and down for warp, or their reverse configuration. The intersections between the warp and weft yarns are also termed interlaced points in the weaving patterns. The black dots of Fig. 4 are the interlacing nodes which are extracted from the weaving pattern.

As shown in Fig. 5, the weaving pattern with size $m_w \times m_h$ covering the cloth surface domain is interlaced as an $m_w \times m_h$ grid of intersections between the warp and weft yarns. At each grid intersection the warp and weft yarns are assumed to be offset along opposite directions (for single layer weaving pattern), and the weaving point value $\eta(u, v)$ provides this offset.

4.2.1 Sampling of Interlaced Nodes

The weaving pattern can be treated separately as two displacement maps, one for the horizontal yarn, and the other for the vertical. They are commonly multichannel color maps for representing multilayer weaving structure, each channel of which indicating the interlaced nodes of a single weaving layer. The simplest case is the two-layer weave structure, which leads to two opposite displacement maps as shown in Fig. 5.

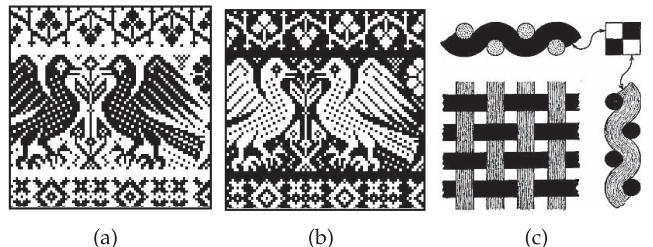


Fig. 5. Two displacement maps extracted from weaving pattern and a weaving example. (a) Displacement map extracted for weft yarns. (b) Displacement map extracted for warp yarns. (c) A simplest 4×4 weaving example.

4.2.2 Sampling of Yarn Path

We employ an interpolatory curve based on the cosine function to sample the weft path between two adjacent interlaced nodes

$$\begin{aligned} h(u) &= \frac{1}{2}(h_f(u) + h_c(u)) \\ &\quad + \frac{1}{2}(h_f(u) - h_c(u)) \cos((u - \lfloor u \rfloor)\pi), \end{aligned} \quad (11)$$

$$h_f(u) = \eta(\lfloor u \rfloor, v), h_c(u) = \eta(\lceil u \rceil, v),$$

where $h_f(u)$ and $h_c(u)$ are the floor and ceil offset scalars respectively of two neighboring interlaced nodes of the v th weft yarn.

4.2.3 Yarn Space Definition

The local position $l_y(u)$ of a vertex at u on the path of the v th weft yarn can be represented as

$$l_y(u) = (0, Hh(u), 0)^T, \quad (12)$$

where H is the crimp amplitude, a physical attribute of each yarn. We introduce variable ϑ along the tangent τ_p of the panel space for calculating the derivative.

Taking the derivative with respect to interim variable ϑ and substituting 0 for ϑ , the tangent and the normal of weft yarn are

$$\begin{aligned} \tau_y(u) &= (S, Hh'(u), 0)^T, \\ n_y(u) &= (Hh'(u), S, 0)^T, \\ b_y(u) &= \tau_y(u) \times n_y(u). \end{aligned} \quad (13)$$

The binormal $b_y(u)$ of weft yarn is the cross-product of the tangent and the normal. The factor S indicates the average distance between two interlaced nodes.

Similarly, we can extract $h(v)$, position $l_y(v)$, tangent $\tau_y(v)$, binormal $b_y(v)$, and normal $n_y(v)$ of warp yarn at parametric position v . Fig. 6 illustrates the varying yarn-space Darboux frame at parametric position u along a weft yarn path.

4.3 The Geometric Model at the Fiber Scale

Sripateep and Bohez proposed a mathematical model based on a migrating helix to simulate fiber migration and packing [30]. Their migration parameters include twisting tension, migration period, and amplitude. We employ their model at the fiber scale. The configuration of the cross-section of a

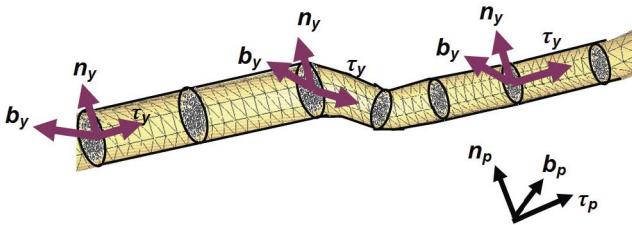


Fig. 6. Varying coordinate system of yarn space along weft yarn path. Coordinate axis are Darboux frames.

fiber can be described as a superquadric. The local position $l_f(u)$ of a fiber at u is

$$l_f(u) = \begin{pmatrix} 0 \\ r(u)\sin(\phi + \omega u) \\ r(u)\cos(\phi + \omega u) \end{pmatrix}, \quad (14)$$

where $r(u)$ is the varying migrating distance of a fiber from the yarn center in the yarn cross-section, ϕ is the initialized intertwined phase and ω is the intertwined speed of the fiber.

4.4 The Shell of Panel, Yarn, and Fiber

Sections 4.1, 4.2, and 4.3 addressed the domain surface, yarn path and fiber path of IDSSs. They are the skeletons of IDSSs. We need to build shells for solid representation of these skeletons. For the weft case, the shell expression is to represent a panel as a subdivision surface, a yarn as a cylinder, and a fiber as a twisted pipe

$$l_s(v) = \begin{pmatrix} 0 \\ K\sin(\theta(v)) \\ K\cos(\theta(v)) \end{pmatrix}, \quad (15)$$

where $l_s(v)$ is the local position of shell. The factor K is set as 0 at the panel scale, and it is set as the radius of a yarn at the yarn scale, and it is set as the radius of a fiber at the fiber scale. This makes (15) impact only the geometry at the yarn scale and the fiber scale. Angle $\theta(v)$ is the angular measure of the cross-section of the shell. It is extracted from global parameter v for weft yarns and is extracted from u for warp yarns. Taking the k th weft yarn as an example, the range $v \in [\frac{1}{3}k, \frac{2}{3}k]$ in the global parametrical domain is linearly mapped as the angular range $\theta \in [0, 2\pi]$ in the yarn cross-section.

Equation (5) produces the unified expression for all scales.

- At the panel scale, the crimp amplitude H is 0; the initialized intertwined phase ϕ is $\frac{\pi}{2}$; the intertwined speed ω is 0 so that both M_y and M_f are identity matrices. Equation (5) is simplified as $\beta_p l_p + \beta_s M_p l_s$ which produces a smooth subdivision surface to represent a woven panel;
- At the yarn scale, M_f is identity matrix for rendering so that (5) is simplified as $\beta_p l_p + \beta_y M_p l_y + \beta_s M_p M_y l_s$ which produces swept surfaces to represent yarns;
- At the fiber scale, none of M_p , M_y , and M_f is identity matrix. Equation (5) cannot be further simplified. It produces migrating surfaces to represent fibers.

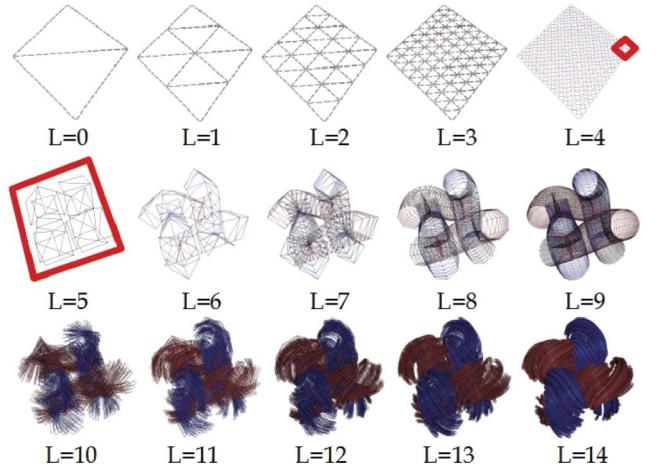


Fig. 7. Tessellation of a woven patch at varying tessellation factors L . Upper row are tessellation at the panel scale; middle row are tessellation at the yarn scale. Lower row are tessellation at the fiber scale.

5 TESSELLATION AND LOD TRANSITION

We need to perform tessellation of IDSS in a similar fashion as other high order surfaces so that they can be rendered on GPUs in parallel. An IDSS is an entire surface that we have to divide the woven fabric into patches for parallel rendering. These patches can be treated as primitives for GPU parallel processing. The IDSS tessellation is uniform with respect to regular power-of-two quads, however our woven patches are assigned different resolutions/tessellation patterns, determined by a tessellation factor for adaptive (semiuniform) tessellation.

Two common artifacts in adaptive tessellation are “popping” between frames and “cracks” in each frame. The two techniques we have chosen to reduce or remove these artifacts are geomorphing of each woven patch between frames and watertight tessellation between woven patches in a single frame.

5.1 Tessellation Levels

As the woven patches can be assigned different tessellation levels, the tessellation factor calculation should be assigned per woven patch. There are two factors contributing to the final metric criterion of tessellation factor L (which is also referred to the LoD value): the distance from the eye position p_e of the camera to the center of the woven patch p_c and the size c of the woven patch. Using this denotation, the tessellation factor L can be calculated using

$$L = \frac{c}{C|p_c - p_e|}, \quad (16)$$

where the coefficient C is a empirical value and the tessellation factor L can be any nonnegative floating-point number.

Fig. 7 shows the tessellation of a woven patch at varying tessellation factors. Each row stands for a scale and each element represents a tessellation in that scale. Each woven patch has 16 weft yarns and 16 warp yarns and 16×16 interlaced nodes. The second and third rows in Fig. 7 show the tessellation at finer resolutions, but only the tessellations at the yarn and fiber levels in the red area are shown.

5.2 Geomorphing to Reduce Popping

Adding or removing vertices changes the geometry of a woven patch and produces a noticeable “popping” artifact. To reduce this artifact and provide a progressive transition for scale-varying geometries between frames, we use geomorphing when a woven patch switches between intrascale LoDs and interscale LoDs.

5.2.1 Intrascale Geomorphing

To blend intra-LoDs, we use weight values α_p , α_y and α_f , respectively, at the panel, yarn, and fiber scales:

$$\begin{aligned}\tilde{l}_p &= (1 - \alpha_p)l_p + \alpha_p\bar{l}_p, \\ \tilde{l}_y &= (1 - \alpha_y)l_y + \alpha_y\bar{l}_y, \\ \tilde{l}_f &= (1 - \alpha_f)l_f + \alpha_f\bar{l}_f,\end{aligned}\quad (17)$$

where \tilde{l}_p is the final morphing position, \bar{l}_p is the bilinear interpolation position of the four nearest points of the coarser LoD and l_p is the position from (8). For example, if we want to get a morphing position at local parametric position (0.375, 0.625) of LoD 3, we calculate the position of the coarser LoD, 2, at parametric positions (0.25, 0.5), (0.5, 0.5), (0.25, 0.75), (0.5, 0.75) and take the bilinear average of them. The general expression for calculating \bar{l}_p is given by

$$\begin{aligned}\bar{l}_p(u, v) &= \frac{1}{4}l_p\left(\frac{\lfloor gu \rfloor}{g}, \frac{\lfloor gv \rfloor}{g}\right) + \frac{1}{4}l_p\left(\frac{\lfloor gu \rfloor}{g}, \frac{\lceil gv \rceil}{g}\right) \\ &\quad + \frac{1}{4}l_p\left(\frac{\lceil gu \rceil}{g}, \frac{\lfloor gv \rfloor}{g}\right) + \frac{1}{4}l_p\left(\frac{\lceil gu \rceil}{g}, \frac{\lceil gv \rceil}{g}\right),\end{aligned}\quad (18)$$

where g is the resolution of the woven patch.

5.2.2 Interscale Geomorphing

Interscale popping artifacts also occur during LoD transitions. These artifacts are removed by the contribution factors β_i of scales:

$$\begin{aligned}\tilde{l}_p(u, v) &= \tilde{l}_p(u, v) + \beta_y M_p(u, v)\tilde{l}_y(u) \\ &\quad + \beta_f M_p(u, v)M_y(u)\tilde{l}_f(u) \\ &\quad + M_p(u, v)M_y(u)M_f(u)l_s(v),\end{aligned}\quad (19)$$

where the coefficients β_y and β_f range from 0 to 1, and indicate how much the yarn is offset from the panel surface and the fiber offset is from the yarn surface, respectively.

As shown in Fig. 8b, the coefficient β_y is applied to a geomorph between the panel and yarn scales: a quad at the panel scale is subdivided into four quads of two overlapping layers at the yarn scale. A Pair of quads composes a yarn segment, which morphs from a plane to form folded quads. After morphing, pairs of quads become interlaced yarn segments at a weave point.

As shown in Fig. 8d, the coefficient β_f is applied to a geomorph between the yarn and fiber scales. Each cylindrical segment of yarn is subdivided into multiple overlapped cylindrical geometries which then are simultaneously scaled and intertwined to form fiber. As β_f increases, these cylindrical geometries shrink into filaments and intertwine. As β_f decreases, these filaments merge and untwist.

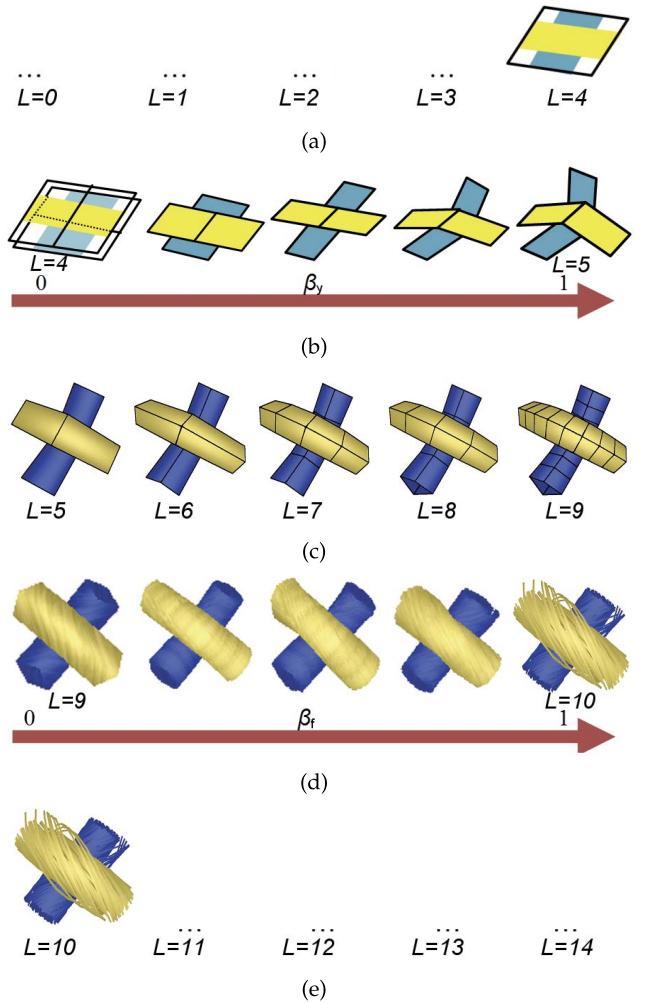


Fig. 8. Interscale geomorphing. (a) LOD levels 0 to 4 at the panel scale. (b) Interscale geomorphing between panel and yarn scales using β_y . (c) LOD levels 5 to 9 of two interlaced yarn segments at the yarn scale. (d) Interscale geomorphing between yarn and fiber scales using β_f . (e) LOD levels 10 to 14 at the fiber scale.

This linearly morphs the vertices in one LoD to those in the next LoD and can be easily implemented using alpha and beta values:

$$\left\{ \begin{array}{l} \alpha_p = L - \lfloor L \rfloor, \alpha_y = \alpha_f = \beta_y = \beta_f = 0, L \in [0, 4] \\ \beta_y = L - \lfloor L \rfloor, \alpha_p = \alpha_y = \alpha_f = \beta_f = 0, L \in [4, 5] \\ \alpha_y = L - \lfloor L \rfloor, \alpha_p = \alpha_f = \beta_y = \beta_f = 0, L \in [5, 9] \\ \beta_f = L - \lfloor L \rfloor, \alpha_p = \alpha_y = \alpha_f = \beta_y = 0, L \in [9, 10] \\ \alpha_f = L - \lfloor L \rfloor, \alpha_p = \alpha_y = \beta_y = \beta_f = 0, L \in [10, 14] \end{array} \right\}. \quad (20)$$

5.3 Watertight Tessellation to Remove Cracks

Scale-varying geometries also introduce “crack” artifacts occurring at the boundaries of two adjoining woven patches of different tessellations within a single frame.

5.3.1 Intrascale Cracks

Intrascale cracks are caused by simple topological mismatching where there is a vertex from a high-resolution patch located on the edge of a low-resolution patch. We remove these vertices using the snap function from Dyken’s approach [31], which moves the parametrical coordinates

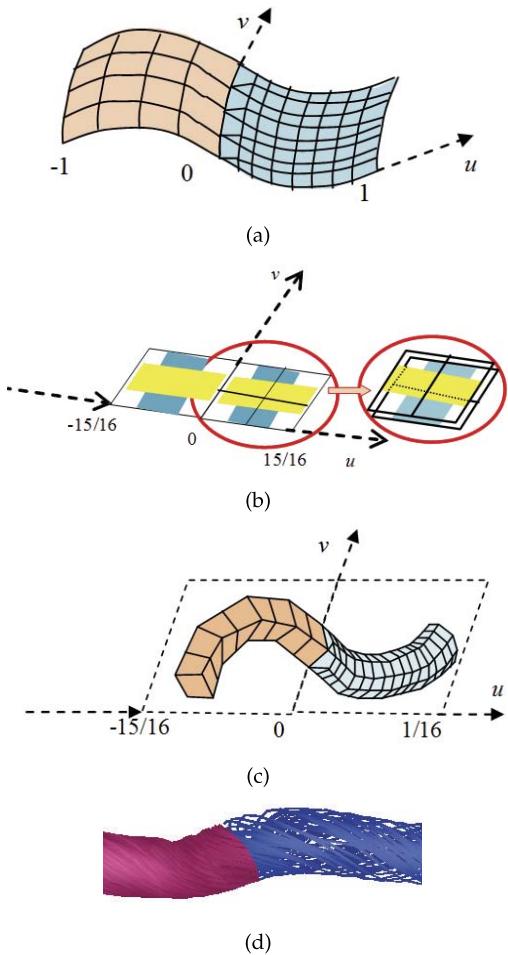


Fig. 9. Crack removal between intrascale LODs and between interscale LODs. (a) Removal of intrascale cracks at the panel scale. (b) Removal of interscale cracks between panel and yarn scales. (c) Removal of intrascale cracks at the yarn scale. (d) Removal of interscale cracks between yarn and fiber scales.

(the hanging node in Dyken’s approach) of the finer resolution to the nearest parametrical coordinates shared by the coarser resolution on the boundary. For our application, the dyadic snap function ϕ can be rewritten as

$$\phi_p(\{u\}, \{v\}) = \begin{cases} \tilde{p}_p(\text{round}(\{u\}g_l)/g_l, \{v\}), \text{left} \\ \tilde{p}_p(\text{round}(\{u\}g_r)/g_r, \{v\}), \text{right} \\ \tilde{p}_p(\{u\}, \text{round}(\{v\}g_t)/g_t), \text{top} \\ \tilde{p}_p(\{u\}, \text{round}(\{v\}g_b)/g_b), \text{bottom} \\ \tilde{p}_p(\{u\}, \{v\}), \text{inner}, \end{cases} \quad (21)$$

where $\{u\}$ and $\{v\}$ are the fractional parts of u and v , and the $(\{u\}, \{v\})$ is the local parametric position of a woven patch, and g_l , g_r , g_t , and g_b are the resolutions of four neighboring woven patches. This function only maps the vertices on boundary patches in an intrascale tessellation to ensure topologically consistency with neighboring tessellations.

Fig. 9a shows an example of the intrascale tessellation of two adjacent woven patches at the panel scale. The vertices on the left boundary of the blue woven patch are moved to the nearest vertices of the coarser red woven patch. The operation was implemented using (21).

Fig. 9c shows another example of the intratessellation of two adjacent woven patches, this time at the yarn scale. For

clarity, the figure shows only one of the weaving nodes of the woven patch. We use (21) to map the vertices on the left boundary of blue yarn segment to its nearest vertex on the same boundary of the lower detail red yarn segment. This implementation only uses the “round” instruction, which is easy to code in the vertex shader.

5.3.2 Interscale Cracks

Interscale cracks occur because of the different topologies of panel, yarn, and fiber. They are found between panel level and yarn level and between yarn and fiber level.

Fig. 9b shows an example of the interscale tessellation of two adjacent woven patches between the panel scale and the yarn scale. The left woven patch is at the panel scale where $L = 4$, as shown in Fig. 8a, and the right woven patch is at the yarn scale where $L = 5$ as shown in Fig. 8b. The left woven patch is tessellated as a single quad using image-based rendering to draw the interlaced shape, and the right woven patch is tessellated as two quads for the weft thread and two quads for the warp. The factors β_y of boundary vertices of the yarn patch are set to 0 to satisfy that the yarn scale terms $\beta_y M_p \tilde{l}_y$ of (19) are 0. Therefore, the geometry of these two woven patches can be linked.

Similarly, interscale tessellation of two adjacent woven patches between the yarn scale and the fiber scale requires addition effort. The fiber scale term $\beta_f M_p M_y \tilde{l}_f$ of (19) for the boundary vertices of the fiber patch are set to 0 that the boundary vertices are not offset while the inner vertices of the fiber patch are. As shown in Fig. 9d, this additional effort can guarantee that the boundary vertices of the fiber patch are overlapped with the boundary vertices of the yarn patch.

6 IMPLEMENTATION

In this section, we describe our implementation of IDSSs on DX11 GPUs and then on current generation GPUs.

6.1 Implementation on DX11 GPUs

DX11 GPUs support complete hardware tessellation in the graphics pipeline by providing three new stages: the Tessellator, the Hull Shader, and the Domain Shader. We use (7) to calculate the control points of Bezier patches and (16) to calculate the tessellation factors in the hull shader. After configuring tessellation within the tessellator, we use (19) to calculate positions of tessellated vertices of woven patches in the domain shader. The following describes these procedures.

6.1.1 Hull Shader

The code in Listing 1 is rewritten based on the hull shader of “SubD11” example of the Microsoft DirectX SDK, Nov 2008. The input of the hull shader includes the woven patch index and the control vertex index.

Listing 1: Hull Shader

```
BEZIER_CONTROL_POINT SubDToBezierHS (
    InputPatch<VS_CONTROL_POINT_OUTPUT,
    <MAX_POINTS> p,
    uint i : SV_OutputControlPointID,
    uint PatchID : SV_PrimitiveID)
```

TABLE 1
Experiment Result for Woven Fabrics Applying Different Schemes

experiment fabrics		uniform L=7 (yarn scale)				uniform L=9 (yarn scale)				adaptive			
id	# ^{*4}	#yarns	t(ms) ^{*1}	FPS ^{*2}	#tri ^{*3}	t(ms) ^{*1}	FPS ^{*2}	#tri ^{*3}	t(ms) ^{*1}	FPS ^{*2}	#tri ^{*3}	# ^{*5}	# ^{*6}
0	8 × 8	256	115.82	8	524288	403.21	2	2097152	45.22	22	196608	0/0/3	59
1	16 × 16	1024	402.35	2	2097152	1782.33	0	8388608	42.43	23	188416	0/42/0	214
2	4 × 4	128	34.23	29	131072	119.74	8	524288	49.27	20	229376	0/3/2	9
3	8 × 8	256	117.21	8	524288	427.25	2	2097152	42.66	23	162304	9/55/0	0
4	16 × 16	1024	362.15	3	2097152	1569.60	1	8388608	40.16	25	151424	216/40/0	0
5	32 × 32	2048	1538.23	1	8388608	6223.21	0	33554432	37.21	27	142336	146/33/0	85

^{*1} t(ms): time of each frame; ^{*2} FPS: frames per second; ^{*3} #tri: number of triangles rendered triangles

^{*4} number of total patches; ^{*5} number of patches at the panel/yarn/fiber scale ^{*6} number of culled patches

```
{
    //loaded patch data from a buffer
    Buffer.Load(SV_PrimitiveID)
    switch(i)
    {
        case 5,6,10,9:
            //Fig.3 (a) from Equation (7)
            Output.vPosition= interior mask;
            break;
        case 1,2,13,14,4,8,7,11:
            //Fig. 3 (b) from Equation (7)
            Output.vPosition= edge mask;
            break;
        case 0,3,15,12:
            Output.vPosition= vertex mask;
            break;
    }
    Output.vTangent= Equation (10);
    Output.vBinormal= Equation (10);
    return Output;
}
```

The hull shader first loads the precomputed data of a woven patch from buffers, and then selects a Bezier construction mask according to the indices of the Bezier control vertices. The construction masks (Fig. 3) are closed forms of the Bezier approximation of ternary interpolatory subdivisions, which are easier to implement than (7), which requires the multiplication of multiple matrices. We also calculate the tangents, normals, and binormals of a woven patch in the hull shader using (10).

6.1.2 Domain Shader

The code in Listing 2 is rewritten from the “BezierEvalDS” function of “SubD11” example.

Listing 2: Domain Shader

```
[domain("quad")]
DS_OUTPUT BezierEvalDS(HS_CONSTANT_
DATA_OUTPUT
input, float2 uv : SV_DomainLocation,
const OutputPatch<BEZIER_CONTROL_POINT,
16> bezpatch)
{
float4 BasisU = BernsteinBasis(UV.x);
float4 BasisV = BernsteinBasis( UV.y );
float3 l_p = Equation (8);
float3 tau_p = Equation (10);
```

```
float3 b_p = Equation (10);
float3 n_p = cross(tau_p,b_p);
float3x3 M_p = float 3x3(tau_p,n_p,b_p);
float3 l_y = Equation (12);
float3x3 M_y = Equation (13);
float3 l_f = Equation (14);
float3 l_s = Equation (15);
float3 p_p = l_p + beta_y*mul(l_y, M_p) +
beta_f*mul(mul(l_f, M_y), M_p)+;
mul(mul(mul(l_s, M_f), M_y), M_p); //((19))
Output.vPosition = mul(float4(p_p, 1),
g_mViewProjection);
return Output;
}
```

It first calculates the Bernstein terms for the u and v directions, then uses Bezier Combination, (8) and (9) to calculate the position, tangent, and binormal at the panel scale from Bezier Control Patches. The tangent, binormal, and normal form a matrix which is the yarn-space Darboux Frame at parametric position (u, v) . We calculate local positions at the yarn and fiber scales, and then use (19) to synthesize them.

6.2 Implementation on Current GPUs

In order to make our framework compatible with current GPUs, we employed two GPU passes: a GPU pass to simulate the hull shader and a rendering pass in which the vertex shader simulates the work of the domain shader. Between the first and second GPU passes, tessellation instancing using ARP (adaptive refinement kernel) [18] is performed.

7 RESULTS AND DISCUSSION

This section presents runtime results including performance data and screenshots that have been collected using IDSSs. We implemented the approach on a NVIDIA 8400 MG mobile GPU using Direct3D 10 and HLSL. The experiment data collected is shown as Table 1. The experiment is based on our previous work [38], [39], [40] on effective physical-based deformation, collision detection, and self-collision detection at the panel scale. We began by placing woven fabric with a light source over its center and tessellating it on the fly. The woven fabric was deformed to produce folds, drapes, and wrinkles at the panel scale.

The results show that the IDSS can represent and tessellate a woven fabric for scale-variation at interactive frame rates even on a low performance DX10 mobile GPU. Comparing between different schemes within the same fabric id in Table 1, the adaptive scheme can select an

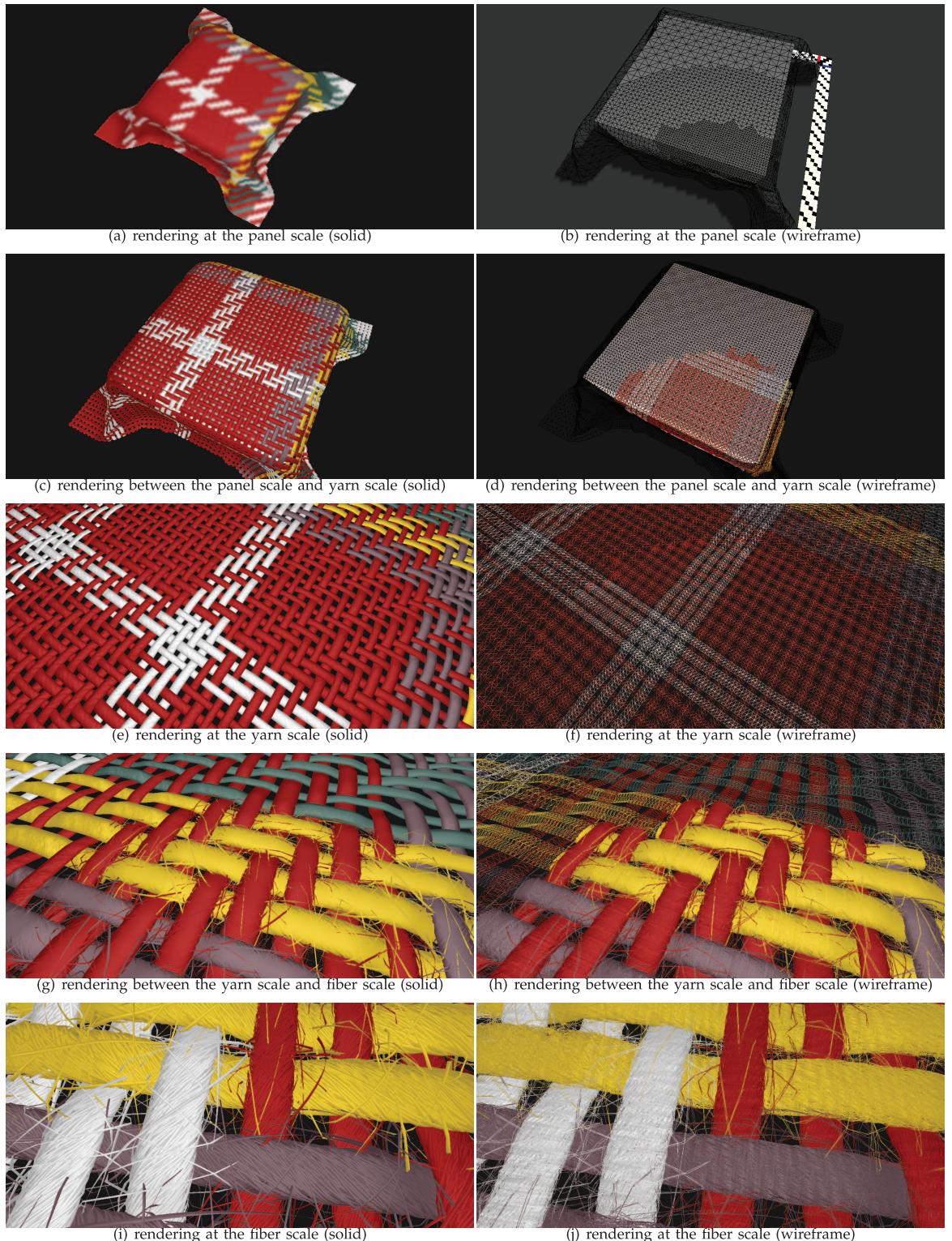


Fig. 10. Snapshots of IDSS representation and tessellation. (a) and (b) Intrascale tessellation at the panel scale. (c) and (d) Interscale tessellation between panel scale and yarn scale. (e) and (f) Intrascale tessellation at the yarn scale. (g) and (h) Interscale tessellation between yarn scale and fiber scale. (i) and (j) Intrascale tessellation at the fiber scale. (a), (c), (e), (g), and (i) solid rendering. (b), (d), (f), (h), and (j) wireframe rendering.

appropriate tessellation rather than the fixed, super high-resolution tessellation selected of the uniform scheme. We noticed that the average rendering frame rate of the adaptive scheme at varying view distances is stable, albeit semiuniform. Comparing between different fabrics within the adaptive scheme in Table 1, varying the view distances

leads to a smooth transition between the fiber and panel scales. From the data collected using the adaptive scheme (the last five columns in Table 1), when the view zooms out from the fiber scale to the panel scale, the frame time is stable, although the number of woven patches in each scale are changed.

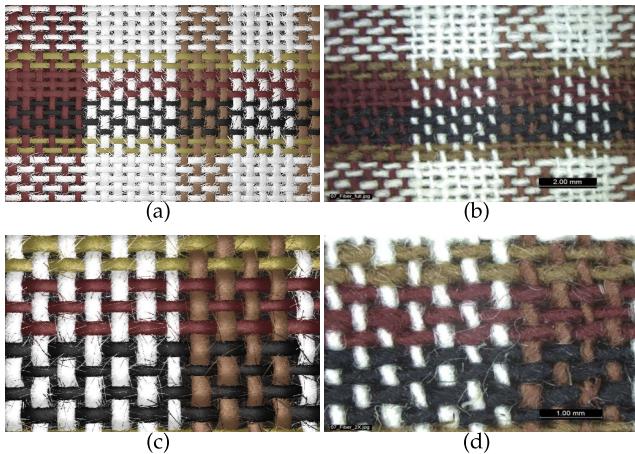


Fig. 11. Comparison between IDSS rendering results and photos taken in the microscope. (a) Rendering result at 2 mm. (b) Photo taken at 2 mm in the microscope. (c) Rendering result at 1 mm. (d) Photo taken at 1 mm in the microscope.

In addition, we noticed that the frame rate decreases sharply when the number of tessellated triangles and woven patches increases. The most time-consuming codes are the texture sampling of the weaving pattern using the “tex2D” instruction and the matrix transformations of (19) in the vertex shader for each tessellated vertex.

Fig. 10 presents the snapshots with solid shading and wireframe to show the adaptive tessellation of an IDSS. The IDSS can reproduce complex geometric structures of woven fabrics at both interscale and intrascale. The snapshots are captured from far to close. Figs. 10a and 10b show the intrascale tessellation at the panel scale. Figs. 10c and 10d show the interscale tessellation between the panel and yarn scales. Figs. 10e and 10f show the intrascale tessellation at the yarn scale. Figs. 10g and 10h show the interscale tessellation between yarn and fiber scales. Figs. 10i and 10j show the intrascale tessellation at the fiber scale. The solid rendering snapshots: Figs. 10a, 10c, 10e, 10g, and 10i show that the intrascale transition is smooth while interscale transition is approximately smooth. It can be seen that any crack artifacts have been removed and the tessellation between woven patches is watertight. The wireframe rendering snapshots, Figs. 10b, 10d, 10f, 10h, and 10j show that the woven fabrics exhibit scale-varying geometries. In Fig. 10d, the woven patches near the viewpoint are represented as yarn geometries while the others are represented as panel geometries. In Fig. 10h, the focused woven patches are represented as intertwined fiber assemblies while the others are represented as cylindrical geometries.

Comparisons between IDSS rendering results and photos taken in the microscope are shown in the Fig. 11. The comparisons are performed, respectively, at 1 and 2 mm corresponding to the view distances in the microscope.

More results can be found in the accompanying video, demonstrating geomorphing for reducing popping between frames. A low-resolution version of the video can be found at <http://www4.comp.polyu.edu.hk/~csjhzhang/IDSS1.wmv>. The high-resolution video can be found at <http://www4.comp.polyu.edu.hk/~csjhzhang/IDSS2.wmv>.

8 CONCLUSION AND FUTURE WORK

In this paper, we propose the Intertwisted Displacement Subdivision Surface, a representation and tessellation of the scale-varying geometry of woven fabrics. This permits smooth, continuous interscale and intrascale multiple-view scalability in woven fabric visualization. The proposed IDSS representation extends the DSS by adding capacity for introducing interlaced, intertwined vector displacement and a three-scale transformation synthesis. This formulation is especially suitable for scale-varying applications in the detailed representation of woven fabrics.

The geometric details of woven fabrics are modeled separately at the panel scale, the yarn scale and the fiber scale. The geometry at the panel scale is modeled as a smooth ternary interpolatory subdivision surface and then approximated by bicubic Bezier patches. Finally, transitions between interscale LoDs and intrascale LoDs are addressed. We reduce the inherent popping artifacts between frames and remove cracks within a single frame. The proposed representation has a wide range of potential applications, including garment design, 3D modeling, games and any application requiring real-time cloth simulation. In the future, we endeavor work on the bicubic Bezier approximation of ternary interpolatory subdivisions for irregular surfaces and knitted fabrics.

ACKNOWLEDGMENTS

The work was supported by The Hong Kong Research Institute of Textiles and Apparel under Grant ITP/012/07TP (<http://www.hkrita.com>), and Science Foundation of China under Grant 60973084, and US Natural Science Foundation (NSF) of Guangdong under Grant (915106410-1000106), and Fundamental Research Funds for the Central Universities (2009zz0016).

REFERENCES

- [1] A. Lee, H. Moreton, and H. Hoppe, “Displaced Subdivision Surfaces,” *Proc. 27th Ann. Conf. Computer Graphics and Interactive Techniques*, pp. 85-94, 2000.
- [2] D. Blythe, “The Direct3D 10 System,” *ACM Trans. Graphics*, vol. 25, no. 3, pp. 724-734, 2006.
- [3] B. Dudson, “Tessellation of Displaced Subdivision Surfaces in DX11,” *GPU-BBQ*, 2008.
- [4] J.F. Blinn and M.E. Newell, “Texture and Reflection in Computer Generated Images,” *Comm. ACM*, vol. 19, no. 10, pp. 542-547, 1976.
- [5] J.F. Blinn, “Simulation of Wrinkled Surfaces,” *Proc. ACM SIGGRAPH*, vol. 12, no. 3, pp. 286-292, 1978.
- [6] T. Kaneko, T. Takahei, M. Inami, N. Kawakami, Y. Yanagida, T. Maeda, and S. Tachi, “Detailed Shape Representation with Parallax Mapping,” *Proc. The 11th Int'l Conf. Artificial Reality and Telexistence (ICAT '01)*, vol. 11, pp. 205-208, 2001.
- [7] R.L. Cook, “Shade Trees,” *Proc. ACM SIGGRAPH*, vol. 18, no. 3, pp. 223-231, 1984.
- [8] J. Peng, D. Kristjansson, and D. Zorin, “Interactive Modeling of Topologically Complex Geometric Detail,” *Proc. ACM SIGGRAPH*, pp. 635-643, 2004.
- [9] S.D. Porumbescu, B. Budge, L. Feng, and K.I. Joy, “Shell Maps,” *Proc. ACM SIGGRAPH*, pp. 626-633, 2005.
- [10] W. Donnelly, “Per Pixel Displacement Mapping with Distance Function,” *GPU Gems 2*, pp. 123-137, Addison-Wesley Professional, 2005.
- [11] M.M. Oliveira, G. Bishop, and D. McAllister, “Relief Texture Mapping,” *Proc. 27th Ann. Conf. Computer Graphics and Interactive Techniques*, pp. 359-368, 2000.

- [12] L. Wang, X. Wang, X. Tong, S. Lin, S. Hu, B. Guo, and H.Y. Shum, "View-Dependent Displacement Mapping," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 334-339, 2003.
- [13] X. Wang, X. Tong, S. Lin, S. Hu, B. Guo, and H.Y. Shum, "Generalized Displacement Maps," *Proc. Eurographics Symp. Rendering*, pp. 227-234, 2004.
- [14] P. Musialski, R.F. Tobler, and C.A. Wuthrich, "Multi-resolution Geometric Details on Subdivision Surfaces," *Proc. Fifth Int'l Conf. Computer Graphics and Interactive Techniques*, pp. 211-218, 2007.
- [15] V. Settgast, K. Müller, C. Fünfzig, and D. Fellner, "Adaptive Tesselation of Subdivision Surfaces," *Computers and Graphics*, vol. 28, pp. 73-78, 2004.
- [16] M. Bunnell, "Adaptive Tesselation of Subdivision Surfaces with Displacement Mapping," *GPU Gems 2*, pp. 109-122, Addison-Wesley Professional, 2005.
- [17] T. Boubekeur and C. Schlick, "Generic Mesh Refinement on GPU," *Proc. ACM SIGGRAPH/Eurographics Graphics Hardware*, pp. 99-104, 2005.
- [18] T. Boubekeur and C. Schlick, "Generic Adaptive Mesh Refinement," *GPU Gems 3*, Addison-Wesley Professional, 2007.
- [19] T. Boubekeur and C. Schlick, "A Flexible Kernel for Adaptive Mesh Refinement on GPU," *Computer Graphics Forum*, vol. 27, no. 1, pp. 102-113, 2008.
- [20] T. Boubekeur and M. Alexa, "Phong Tessellation," *ACM Trans. Graphics*, vol. 27, no. 5, pp. 141:1-141:5, 2008.
- [21] A. Tatarinov, "Instanced Tessellation in DirectX 10," <http://developer.download.nvidia.com/presentations/2008/GDC/InstTessCompatible.pdf>, 2008.
- [22] C. Loop and S. Schaefer, "Approximating Catmull-Clark Subdivision Surfaces with Bicubic Patches," *ACM Trans. Graphics*, vol. 27, no. 1, pp. 1-11, 2008.
- [23] G. Li, C. Ren, J. Zhang, and W. Ma, "Approximation of Loop Subdivision Surfaces for Fast Rendering," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 4, pp. 500-514, Apr. 2011.
- [24] D. Doo and M. Sabin, "Behaviour of Recursive Division Surfaces Near Extraordinary Points," *Computer-Aided Design*, vol. 10, pp. 356-360, 1978.
- [25] E. Catmull and J. Clark, "Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes," *Computer-Aided Design*, vol. 10, pp. 350-355, 1978.
- [26] M.F. Hassan, I.P. Ivrissimitzis, N.A. Dodgson, and M.A. Sabin, "An Interpolating 4-Point C₂ Ternary Stationary Subdivision Scheme," *Computer Aided Geometric Design*, vol. 19, no. 1, pp. 1-18, 2002.
- [27] G. Li and W. Ma, "Interpolatory Ternary Subdivision Surfaces," *Computer Aided Geometric Design*, vol. 23, no. 1, pp. 45-77, 2005.
- [28] H. Wang and K. Qing, "Improved Ternary Subdivision Interpolation Scheme," *Tsinghua Science and Technology*, vol. 10, no. 1, pp. 128-132, 2005.
- [29] C. Beccari, G. Casciola, and L. Romani, "An Interpolating 4-Point C₂ Ternary Non-Stationary Subdivision Scheme with Tension Control," *Computer Aided Geometric Design*, vol. 24, no. 4, pp. 210-219, 2007.
- [30] K. Sriprateep and E.L.J. Bohez, "A New Computer Geometric Modelling Approach of Yarn Structures for the Conventional Ring Spinning Process," *J. Textile Inst.*, vol. 100, pp. 223-236, 2008.
- [31] C. Dyken, M. Reimers, and J. Seland, "Semi-Uniform Adaptive Patch Tessellation," *Computer Graphics Forum*, vol. 28, pp. 2255-2263, 2009.
- [32] F. Durupinar and U. Güdükbay, "Procedural Visualization of Knitwear and Woven Cloth," *Computers and Graphics*, vol. 31, no. 5, pp. 778-783, 2007.
- [33] Y. Xu, Y. Chen, S. Lin, H. Zhong, E. Wu, B. Guo, and H.Y. Shum, "Photorealistic Rendering of Knitwear Using the Lumislice," *Proc. SIGGRAPH*, pp. 391-398, 2001.
- [34] H. Zhong, Y. Xu, B. Guo, and H. Shum, "Realistic and Efficient Rendering of Free-Form Knitwear," *J. Visualization and Computer Animation, Special Issue on Cloth Simulation*, 2000.
- [35] N. Adabala, N.M. Thalmann, and G. Fei, "Visualization of Woven Cloth," *Proc. 14th Eurographics Workshop Rendering*, pp. 178-185, 2003.
- [36] N. Adabala, N.M. Thalmann, and G. Fei, "Real-Time Rendering of Woven Clothes," *Proc. ACM Symp. Virtual Reality Software and Technology*, pp. 41-47, 2003.
- [37] G. Simon, "Real-Time Approximations to Subsurface Scattering," *GPU Gems 1*, pp. 263-277, Addison-Wesley Professional, 2004.
- [38] W.S.K. Wong and G. Baciu, "Dynamic Interaction Between Deformable Surfaces and Nonsmooth Objects," *IEEE Trans. Visualization and Computer Graphics*, vol. 11, no. 3, pp. 329-340, May/June 2005.
- [39] W.S.K. Wong and G. Baciu, "Image-Based Collision Detection for Deformable Cloth Models," *IEEE Trans. Visualization and Computer Graphics*, vol. 10, no. 6, pp. 649-663, Nov./Dec. 2004.
- [40] W.S.K. Wong and G. Baciu, "A Randomized Marking Scheme for Continuous Collision Detection in Simulation of Deformable Surfaces," *Proc. ACM Int'l Conf. Virtual Reality Continuum and its Applications*, pp. 181-188, 2004.



Jiahua Zhang received the BS degree in computer science and engineering at South China University of Technology, in 2007. Now, he is working toward the PhD degree from the Department of Computing, The Hong Kong Polytechnic University. His main interests include GPU programming and GPU architecture.



George Baciu received the BMath degree in computer science and applied mathematics in 1985, and the MSc and PhD degrees in engineering from the University of Waterloo, in 1987 and 1992, respectively. He started the GAMA Lab at The Hong Kong Polytechnic University.



Dejun Zheng received the BSc and MSc degrees of textile design and materials from the Zhejiang University of Sci-Tech, in 2004 and 2007, respectively. Now, he is working toward the PhD degree from the Department of Computing, The Hong Kong Polytechnic University. His interests include textile color technology, textile design, and textile CAD.



Cheng Liang received the BS degree in computer science and engineering at South China University of Technology, in 2007. She is a research assistant in the Department of Computing, The Hong Kong Polytechnic University.



Guqing Li received the BS degree in mathematics from the University of Science and Technology of China, MS degree from Naikai University, and PHD degree of computer application from Chinese Academy of Science. He is a professor of computer science and engineering at South China University of Technology. His main research interests include digital geometry processing and reverse engineering.



Jinlian Hu received the PhD degree from UMIST, United Kingdom, in 1990. She is a professor, in the Institute of Textile and Clothing, the Hong Kong Polytechnic University. She is the head of PolyU's Shape Memory Textile Centre, specializing in textile fabrics and clothing construction.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.