

# Topologically Robust B-spline Reconstruction of Fibers from 3D Images

Dennis Mosbach, Katja Schladitz, Bernd Hamann, Hans Hagen

**Abstract** The micro-structure of wood-based insulation materials is analyzed to gain insight into how features on microscopic scales influence macroscopic thermal conductivity. Three-dimensional (3D) image data obtained by micro-computed tomography reveals a complex structure formed by cellulose fibers. To study the effect of geometry changes, simple B-spline representations of these fibers are highly desirable. A straightforward solution is to extract a triangulated isosurface from the 3D image and partition it into quadrilateral macro-cells with disk-like topology. For each cell, a B-spline surface is constructed by minimizing a least squares error term. However, the physical processing of the material affects the structure of the fibers. The resulting changes in surface topology cause difficulties for the quadrilateral partitioning. Image processing tools can solve these topological issues, but they also impact geometry. We present a novel approach that splits geometry and topology processing of the data. It allows for topological simplification while still preserving the geometry of a scanned object. Established B-spline approximation methods are used to create a model. The involved mathematical equations are described in detail with a focus on simple implementation.

---

Dennis Mosbach <sup>1,2</sup>  
e-mail: dennis.mosbach@itwm.fraunhofer.de

Katja Schladitz <sup>2</sup>  
e-mail: katja.schladitz@itwm.fraunhofer.de

Bernd Hamann <sup>3</sup>  
e-mail: hamann@cs.ucdavis.edu

Hans Hagen <sup>1</sup>  
e-mail: hagen@cs.uni-kl.de

<sup>1</sup> Computer Graphics and HCI Group, University of Kaiserslautern, 67663 Kaiserslautern, Germany

<sup>2</sup> Image Processing Department, Fraunhofer ITWM, 67663 Kaiserslautern, Germany

<sup>3</sup> Department of Computer Science, University of California, Davis, CA 95616, U.S.A.

Our presented results demonstrate that smooth and accurate models can be created for challenging data.

## 1 Introduction

An important part in researching material properties is the generation of models representing microscopic structures which can be used for visualization and simulation purposes. How these models are obtained strongly depends on the size and complexity of the samples, the initial data acquisition, and the target application.

We are concerned with model generation of cellulose fibers. A cellulose fiber is understood as a hollow tube that is open on both ends. The empty space inside is called *lumen*. Models of such structures are helpful to understand and assess mechanical properties of a material on a macroscopic scale [1]. For example, this understanding allows researchers to establish a relation between the micro-structure and physical properties of paper [15]. Another example is the analysis of sound-dampening properties of wood-based acoustic insulation material with respect to its tortuosity, a parameter that relates physical, acoustic, and morphological properties of a material [20].

Our goal is to investigate heat conductivity properties of wood-based thermal insulation materials. The micro-structure of these materials consists of a complex system of cellulose fibers which occur in chunks, chips, and as individuals. To conduct simulations, B-spline surface models of such fibers are desired. These models should consist of a low number of continuously connected surfaces and include the inner and outer surfaces of the fiber wall. The main feature of interest is the shape of the fibers as well as the contained lumen. This means that small holes in the wall as well as roughness along the surface can be neglected and do not need to be present in the final model. B-spline surfaces are defined by a grid of control points. Storing just those control points requires far less memory than a high-resolution triangulation. Furthermore, B-splines are better suited to represent geometric detail than a down-sampled mesh. Changing control point locations directly affects the shape of the surface, which allows scientists to generate additional fibers for simulations from a small number of prototypes.

The initial data is a scan of a material sample given as 16-bit grayvalue image which is obtained by high-resolution synchrotron computed tomography. For denoising, a  $5 \times 5 \times 5$  median filter is applied after which the image is binarized with a global gray value threshold. Individual fibers are then extracted from this image based on a local shape criterion [2].

However, the physical processing of the material causes damage to the micro-structure which leads to cuts and holes in the fiber walls as well as increased surface roughness in some areas. As Figure 1 shows, these effects

complicate the topological structure of the surface by adding numerous small tunnels, holes, and cavities.

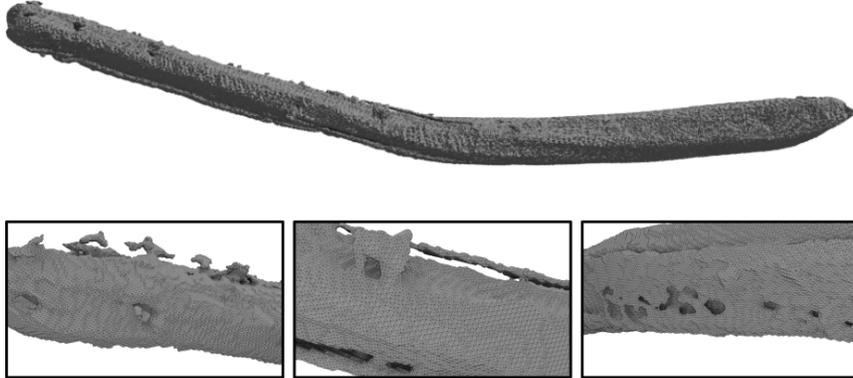


Fig. 1: Example of an extracted fiber. While the overall geometrical shape appears to be smooth, close-ups show artifacts affecting the local topology on the surface.

A straightforward solution to create a B-spline model of those fibers is to extract the isosurface of such a fiber in form of a triangulated mesh. In order to compute a continuous B-spline model, this mesh needs to be partitioned into a collection of quadrilateral macro-cells. Each macro-cell can then be approximated by a B-spline surface. Due to the non-trivial topological situation on the fiber's surface, most approaches to compute such a quadrilateral decomposition either fail or produce a very large amount of small macro cells. Standard image denoising operations can solve the topological problems, but might also introduce a geometrical error to the resulting surface.

Our contribution is a new approach that splits the data processing pipeline into geometry and topology processing. A strongly cleaned and repaired image is used to create the mesh of larger quadrilateral faces capturing the topology of the idealized cellulose fiber. This quad-mesh is then used to partition the original data, extracted from the unprocessed image, by assigning each data point to the nearest quad-face. Processing the data this way provides the necessary input for established B-spline approximation methods. The resulting B-spline models have fiber-like topology and closely approximate the geometry of the original data.

## 2 Related Work

While no methods exist to extract isosurfaces directly in form of B-splines, a variety of research has been published concerning the construction of B-spline approximations of discrete data given as point clouds or surface meshes. Such an explicit representation can easily be obtained from a 3D image by contouring methods like Marching Cubes.

Regular tensor product B-spline surfaces can only represent structures with disk-like topology. To construct a model for complex objects, the data needs to be decomposed into quadrilateral cells. The data points in each cell are then approximated by a single B-spline surface. The key challenge is to ensure certain continuity constraints along boundaries. While a watertight model, i.e.,  $C^0$ -continuous transitions along shared boundaries, is almost always a requirement, it is often desired to also have at least tangent plane continuity, i.e.,  $G^1$ -continuous transitions.

Eck and Hoppe [3] provide an in-depth description of the overall pipeline starting with a point cloud. Their first step is to establish a topological structure of the data by generating a fine surface mesh. Using a Voronoi-like tessellation and its dual Delaunay-like complex, the mesh is subdivided into a set of quadrilateral cells where each cell has a disk-like topology. Those cells are parameterized using harmonic maps. A B-spline approximation using a modification of Peters' scheme [18] is then computed to minimize the least squares error between the surfaces and the data points.

Gregorski et al. [8] propose a method to construct a set of B-spline surfaces approximating a given point cloud. They decompose their data into quadrangular cells based on a so-called strip tree which provides an adaptive subdivision of the data, similar to a quad-tree, into small boxes such that the points contained in each box can be approximated by B-spline surfaces with low error. Control points of neighboring surfaces are then adjusted during a post-processing step to ensure  $C^1$ -continuous transitions. Since they do not establish a topological structure of the surface, their approach works best on mostly flat objects.

Given a triangulated surface mesh and a manually defined subdivision into quadrilateral cells, Krishnamurthy and Levoy [13] do a remeshing of each cell to obtain a regular sampling of the data. This implicitly defines a parameterization and reduces the complexity of the B-spline approximation. However, to apply it to complex objects, an automatization of the quadrilateral decomposition is necessary.

Yoo [22] describes an approach to construct a B-spline model of human bones given as point cloud or sequence of computed tomography images. First, the input data is used to define an implicit surface on which a fine quad-mesh is constructed. Each quad and the normal vectors along its boundaries are then interpolated by a B-spline surface.

Yoshihara et al. [23] capture the topology of a given point cloud by constructing an implicit function and applying a level set method. At the cost

of geometrical accuracy in noisy regions, this allows for a stable processing of difficult data. The object’s surface is approximated by a Catmull-Clark subdivision surface and the corresponding control points are used to form a fine quad-mesh which is interpolated by B-spline surfaces.

Lin et al. [14] present a method to create a smooth B-spline model to approximate a mesh. The quadrilateral decomposition is done manually. After constructing a curve network representing the boundaries, the data points in each cell are approximated by a bi-quintic Bézier surface that interpolates the boundary curves. The resulting surface model is made  $G^1$ -continuous by also interpolating pre-computed normal vectors along the boundaries.

Zhao et al. [24] introduce an iterative approach, allowing individual surfaces to use differing knot vectors. Given a point cloud and a partition into quads, their approach constructs an initial set of B-spline surfaces to closely approximate the data and then ensures approximate  $G^1$ -continuity in a numerical post-processing step.

Based on their previous work, Peters and Fan [19] provide an in-depth theoretical analysis of  $G^1$ -continuous B-spline surface constructions. They state necessary constraints that B-splines based on a quad-mesh with arbitrary topology need to satisfy in order to obtain tangent plane continuity everywhere. A key statement is that in a general setting, a  $G^1$ -continuous B-spline surface model can only be constructed when the used knot vectors have at least two interior double knots.

Satisfying these requirements, Fan and Peters [4] provide a construction scheme for bi-cubic B-splines with exactly two double inner knots. Equations for all control points are explicitly given as linear combinations of the surrounding vertices of the quad-mesh, but the method can also be modified to approximate a set of discrete data points.

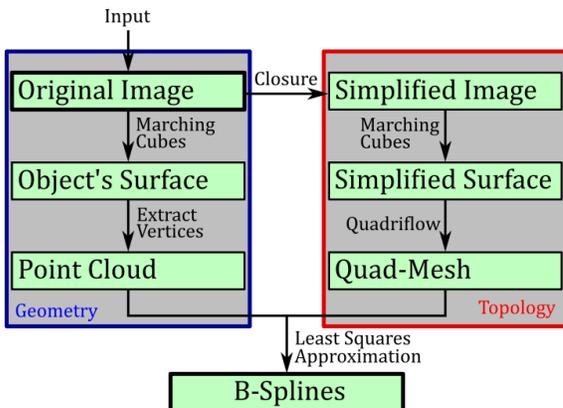
While a large variety of methods for many scenarios exists, they are primarily designed for structures that allow for a decomposition into a relatively small number of quadrilateral macro cells. However, this paper deals with rough fiber structures with complicated surface topology. A straightforward decomposition of the data into disk-like quadrilateral cells would require a very high number of small cells. Automatically removing these effects during a pre-processing step strongly affects the geometry of the resulting B-splines. Hence, an alternative solution is introduced here.

### 3 Pipeline

The key aspect of our method is splitting geometry and topology processing, see Figure 2. After the data has been processed this way, standard methods for B-spline approximation can be applied.

To compensate for small holes and the rough surface structure in the data, a topological simplification is employed. We apply a morphological closure to

**Fig. 2** Pipeline separating geometry and topology processing. To partition the data into quadrilateral cells, a morphological closure operation is applied to the input image. The isosurface of the image is extracted and re-meshed into a quad-mesh defining the topological structure of the B-spline model. The approximation is computed with respect to the data points extracted from the isosurface of the original mesh.



the volume image, i.e., we perform a sequential application of dilation and erosion on the object represented by the image’s foreground [17]. The size of the filter mask needs to be chosen sufficiently large to fill the entire structure. Efficient implementations for large filter masks are available [16]. This operation preserves the original outer boundary in smooth regions and absorbs any outside material into the fiber’s surface, see Figure 3. The inside of the fiber and holes in its wall are closed. The resulting object has the topology of a cylinder. Hence, applying Marching Cubes leads to a surface mesh with a topological structure that is well-suited to be partitioned into large and evenly sized macro-cells. A re-meshing into a small number of quadrilateral cells is performed on this mesh, using a freely available implementation of the Quadriflow algorithm [11]. The resulting quad-mesh is a “rough geometrical approximation” of the data, but it provides a suitable topological structure for a B-spline model.

For the geometry-focused part of the pipeline, the original image is considered. As with the closed image, a triangulated isosurface is extracted using Marching Cubes. The vertices of this triangulation provide the geometry data for the B-spline approximation. To associate them with the topological structure of the quad-mesh, each point is assigned to the closest quad-face. An initial parameterization is obtained by projecting the points onto their associated faces. By assigning the corners of a quadrilateral to the corners of the unit square, a correspondence is established between the position of the projected point in the quadrilateral and a tuple in  $(u, v)$ -parameter space, where  $(u, v) \in [0, 1]^2$ , see Figure 4. This initial parameterization is not optimal, but it suffices to generate a proper initial surface approximation. The parameterization is optimized during the iterative surface construction approach as described in Subsection 4.2.1.

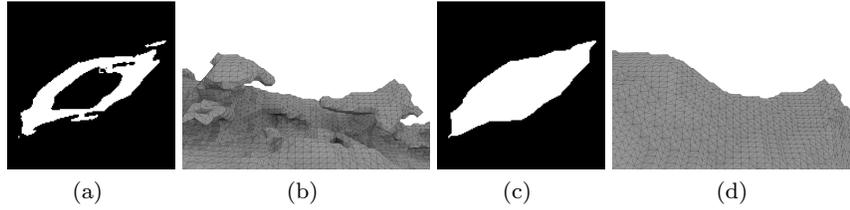


Fig. 3: Performing a morphological closure operation simplifies the topology of the object. A slice of the 3D volume data (a) and a close-up of the corresponding region in the isosurface (b) highlight regions hard to handle for subsequent processing. Applying the closure operation simplifies the surface structure, as seen in (c) and (d).

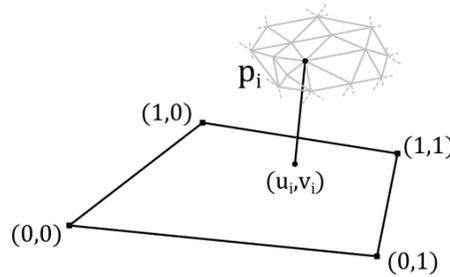


Fig. 4: Initial parameterization. Each vertex of the original isosurface is orthogonally projected onto the closest face of the quad-mesh. The corners of the quad correspond to the corners of the unit square  $[0, 1]^2$ . The relative position of the projected point in the quad defines the parameterization.

Cellulose fibers are hollow. To reconstruct a fiber wall and the contained lumen, the inside and outside of the isosurface representing the wall are considered separately. The topology is the same in both cases. Hence, the same quad-mesh can be used for both approximations. However, each data point needs to be classified as belonging to either the interior or exterior surface of the fiber wall. This classification can be done by applying a Euclidean Distance Transform to the closed image, assigning to each voxel the distance to the closest point of the isosurface [7]. As the morphological closure still mostly preserves the outer surface, data points with low distances (in our case  $\leq 2$  voxels) are considered to belong to the outside surface of the wall, while data points with larger distance values are considered to belong to the inside. Once the classification is performed, separate surface models are constructed, one for each set of data points.

The quad-mesh obtained from the simplified image is closed. Since the layout produced by the Quadriflow algorithm is sensitive to sharp features, the actual open ends of the fibers align well with individual quad-faces. To

reproduce the open structure of fibers, these quads are removed and the open space between inside and outside B-spline surfaces is closed via linear interpolation between the boundary curves.

Image processing operations, including closure, Euclidean Distance Transform and isosurface extraction with Marching Cubes, are performed with the implementations in MAVI [6]. The projection of the data points onto the quad-mesh is performed with the freely available software package libigl [12].

## 4 B-spline Approximation

This Section reviews the construction of B-spline surfaces to approximate discrete data. The concepts are based on literature [5] and the methods mentioned in Section 2. Equations that need to be solved are stated in a uniform notation with a focus on simple implementation.

As input, a set of data points and their parameterization is required. Optionally, they can also have weights assigned to them, which is useful when considering a non-uniform distribution of data points. For complex objects that cannot be modeled with a single surface, a partition into quadrilateral cells is required. The data of each cell is then approximated by an individual B-spline surface that has to satisfy certain continuity constraints with neighboring surfaces.

After introducing the basic notation in Subsection 4.1, Subsection 4.2 discusses the construction of a single surface. Finally, a global system including the constraints necessary for a continuous model consisting of multiple B-spline surfaces is introduced in Subsection 4.3.

### 4.1 B-spline Surface Notation

A B-spline surface is defined by an order  $k \in \mathbb{N}$ , a knot vector for each parameter direction, and a rectangular grid of control points. To keep the construction of a model with multiple continuously connected surfaces as simple as possible, the B-splines considered here are restricted to quadratic grids of  $n_c \times n_c$  control points and use the same knot vector

$$\tau = ( \underbrace{0, \dots, 0}_{k+1 \text{ times}}, \tau_1, \dots, \tau_{n_c-k-1}, \underbrace{1, \dots, 1}_{k+1 \text{ times}} ) \in [0, 1]^{n_c+k+1}$$

with knots  $\tau_i \leq \tau_{i+1}$  in  $u$  and  $v$  direction. The piecewise polynomial basis functions of order  $k$  defined by knot vector  $\tau$  are denoted by  $N_i(\cdot) := N_{i,k,\tau}(\cdot)$ .

The B-spline surface is given by

$$S : [0, 1]^2 \rightarrow \mathbb{R}^3$$

$$S(u, v) = \sum_{i=1}^{n_c} \sum_{j=1}^{n_c} N_i(u) N_j(v) b_{i,j}.$$

## 4.2 Single B-spline Surface Approximation

First, the approximation of a set of data points by a single B-spline surface is discussed. Given a set of  $n_p$  points  $p_i \in \mathbb{R}^3$  with assigned parameter values  $(u_i, v_i) \in [0, 1]^2$  and a weight  $w_i \in \mathbb{R}$ . The least squares error of the surface with respect to the data is denoted by

$$E_{LS} = \frac{1}{2} \sum_{i=1}^{n_p} w_i \|S(u_i, v_i) - p_i\|^2. \quad (1)$$

An equivalent notation can be used to simplify this term. Let  $b \in \mathbb{R}^{n_c^2 \times 3}$  be a one dimensional list containing the B-spline control points and let  $b_x, b_y, b_z \in \mathbb{R}^{n_c^2}$  be the vectors containing their  $x, y$ , and  $z$  coordinates. A correlation between the list index  $\bar{i}$  and the index  $i, j$  in the original grid can be established with a bidirectional mapping, e.g.,  $\bar{i} = in_c + j$ . Using the same index mapping, let

$$N(u, v) = [N_1(u)N_1(v), \dots, N_{n_c}(u)N_{n_c}(v)] \in \mathbb{R}^{1 \times n_c^2} \quad (2)$$

be a row vector containing the associated tensor products of basis functions. Then, Equation (1) can be written as

$$E_{LS} = \frac{1}{2} \sum_{i=1}^{n_p} w_i \|N(u_i, v_i)b - p_i^T\|^2.$$

Furthermore, let  $N \in \mathbb{R}^{n_p \times n_c^2}$  be the matrix containing the basis function coefficients for the parameters  $u_i, v_i$  associated to each data point  $p_i$  and let a list of all data points be denoted by  $p$ , i.e.,

$$N = \begin{bmatrix} N(u_1, v_1) \\ \vdots \\ N(u_{n_p}, v_{n_p}) \end{bmatrix}, p = \begin{bmatrix} p_1^T \\ \vdots \\ p_{n_p}^T \end{bmatrix}.$$

The least squares error can then be expressed as

$$E_{LS} = \frac{1}{2} \sum_{\sigma \in \{x, y, z\}} (Nb_\sigma - p_\sigma)^T W (Nb_\sigma - p_\sigma)$$

with the weights as diagonal matrix  $W = \text{diag}(w_1, \dots, w_{n_p})$  and gradient

$$\nabla_{\sigma} E_{LS} = N^T W N b_{\sigma} - N^T W p_{\sigma}.$$

Minimizing the least squares error is equivalent to solving  $\nabla_{\sigma} E_{LS} = 0$  for all components  $\sigma \in \{x, y, z\}$ .

Due to the discrete nature of the data, an approximation based on only the least squares error often includes unwanted wiggles. This is compensated by penalizing the deviation from a smooth surface by a fairing term. A commonly used term is the *thin-plate-energy* functional [3, 10]

$$E_{TP} = \frac{1}{2} \int_0^1 \int_0^1 \|\partial_{uu} S(u, v)\|^2 + 2 \|\partial_{uv} S(u, v)\|^2 + \|\partial_{vv} S(u, v)\|^2 \, dudv$$

which, for B-splines, can be rewritten as

$$E_{TP} = \frac{1}{2} \sum_{\sigma \in \{x, y, z\}} b_{\sigma}^T M b_{\sigma}$$

with gradient

$$\nabla_{\sigma} E_{TP} = M b_{\sigma}$$

where  $M \in \mathbb{R}^{n_c^2 \times n_c^2}$  is a matrix containing the integrals of the basis functions given by

$$\begin{aligned} M &= \int_0^1 \int_0^1 (\partial_{uu} N(u, v))^T (\partial_{uu} N(u, v)) \\ &\quad + 2(\partial_{uv} N(u, v))^T (\partial_{uv} N(u, v)) \\ &\quad + (\partial_{vv} N(u, v))^T (\partial_{vv} N(u, v)) \, dudv. \end{aligned}$$

The B-spline basis functions are piecewise polynomials. Hence, their integrals can either be determined analytically or by numeric integration, e.g., applying Gaussian quadrature to each knot interval  $[\tau_i, \tau_{i+1})$  with a high enough degree to be exact for the given case.

To compute a smooth surface that approximates the data points, a linear combination of both terms is minimized, i.e.,

$$\min. (1 - \lambda) E_{LS} + \lambda E_{TP} \tag{3}$$

with the parameter  $\lambda \in [0, 1)$  controlling the impact of the fairing term to the resulting surface. Since (3) is a system of quadratic equations with respect to the B-spline control points, its optimal solution can be found by solving

$$\begin{aligned} 0 &= (1 - \lambda) \nabla_{\sigma} E_{LS} + \lambda \nabla_{\sigma} E_{TP} \\ &= (1 - \lambda) (N^T W N b_{\sigma} - N^T W p_{\sigma}) + \lambda M b_{\sigma} \end{aligned}$$

for each coordinate  $\sigma \in \{x, y, z\}$ . It can be written as

$$((1 - \lambda)(N^T W N) + \lambda M)b = (1 - \lambda)N^T W p. \quad (4)$$

This is a linear system with respect to the control points  $b$ . It can be solved by a variety of freely available code packages, e.g., Eigen [9].

#### 4.2.1 Iterative Approximation

The described approach aims on minimizing the distance between data points  $p_i$  and the points on the surface  $S(u_i, v_i)$ , associated to them through their parameterization. However, these are not necessarily the smallest distances between the data points and the surface. It is common practice to improve the quality of the approximation by employing an iterative approach alternating between the construction of the surface and an update to the parameter values.

After a surface has been constructed, the parameter values of each data point are updated, by minimizing

$$(u'_i, v'_i) = \operatorname{argmin}_{u, v \in [0, 1]} \|S(u, v) - p_i\|.$$

which can either be done exactly using computationally expensive nonlinear optimization or approximately by considering low-degree Taylor series [21]. An example using a first-order Taylor series is shown in Figure 5. The new parameter values are restricted to the unit square. To ensure stable behavior of the method, data points that are close to surface boundaries are not assigned to neighboring surfaces, even when the distance to them might be smaller.

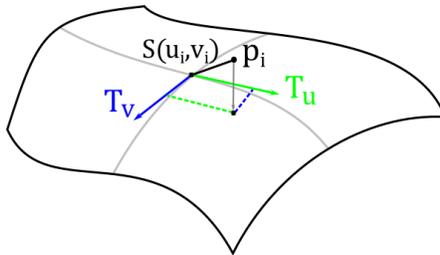


Fig. 5: Parameter optimization using first-order Taylor series. The error vector between each data point  $p_i$  and its corresponding point on the surface  $S(u_i, v_i)$  is projected onto the tangent plane. The difference between the projected point and  $S(u_i, v_i)$  is proportionally applied to the parameterization to obtain an improved parameter tuple  $(u'_i, v'_i)$ .

### 4.3 Constructing a Continuous Surface Model

When constructing a model with  $n_s$  surfaces, the same principles than for a single surface apply, but some modifications are necessary to ensure continuous transitions between neighboring surfaces. Control points and data can be stored in a global list by concatenating the lists of each individual surface. The coefficient, weight, and smoothing matrices are combined to diagonal block matrices. Using

$$\widehat{\mathbf{b}} = \begin{bmatrix} \mathbf{b}^{(1)} \\ \vdots \\ \mathbf{b}^{(n_s)} \end{bmatrix}, \widehat{\mathbf{p}} = \begin{bmatrix} \mathbf{p}^{(1)} \\ \vdots \\ \mathbf{p}^{(n_s)} \end{bmatrix},$$

$$\widehat{\mathbf{N}} = \begin{bmatrix} N^{(1)} & & 0 \\ & \ddots & \\ 0 & & N^{(n_s)} \end{bmatrix}, \widehat{\mathbf{W}} = \begin{bmatrix} W^{(1)} & & 0 \\ & \ddots & \\ 0 & & W^{(n_s)} \end{bmatrix}, \widehat{\mathbf{M}} = \begin{bmatrix} M & & 0 \\ & \ddots & \\ 0 & & M \end{bmatrix},$$

instead of the single surface terms in Equation (4) leads to the same result as constructing each surface individually.

#### 4.3.1 Achieving $C^0$ -continuity by Adding Constraints

To obtain  $C^0$ -continuity, i.e., a watertight model, the row of control points along a shared boundary between any two neighboring surfaces needs to be identical. This can be expressed by a number of constraints of the form  $\widehat{\mathbf{b}}_i - \widehat{\mathbf{b}}_j = 0$  with  $i, j$  being the indices in the global control point vector of the points that need to be identical. All constraints together can be expressed by a matrix vector product:  $\widehat{\mathbf{G}}\widehat{\mathbf{b}} = 0$  where each row of matrix  $\widehat{\mathbf{G}}$  ensures that two corresponding control points have to be identical.

With these constraints, the optimization problem (3) becomes

$$\begin{aligned} \min. \quad & (1 - \lambda)\widehat{E}_{\text{LS}} + \lambda\widehat{E}_{\text{TP}} \\ \text{s.t.} \quad & \widehat{\mathbf{G}}\widehat{\mathbf{b}} = 0. \end{aligned} \tag{5}$$

Using the method of Lagrange multipliers, an optimal solution of (5) can be found by solving

$$\left[ \begin{array}{c|c} ((1 - \lambda)\widehat{\mathbf{N}}^T\widehat{\mathbf{W}}\widehat{\mathbf{N}} + \lambda\widehat{\mathbf{M}}) & \widehat{\mathbf{G}}^T \\ \hline \widehat{\mathbf{G}} & 0 \end{array} \right] \begin{bmatrix} \widehat{\mathbf{b}} \\ \mathbf{A} \end{bmatrix} = \begin{bmatrix} (1 - \lambda)\widehat{\mathbf{N}}^T\widehat{\mathbf{W}}\widehat{\mathbf{p}} \\ 0 \end{bmatrix} \tag{6}$$

with Lagrange multipliers  $\mathbf{A}$ . This system can be implemented and solved in a straightforward way. However, adding the  $C^0$ -constraints in form of a matrix, increases the size of the overall system compared to the unconstrained case.

### 4.3.2 Achieving $C^0$ -continuity by Reduction of Variables

An alternative is to implicitly enforce  $C^0$ -continuity, by using only one variable for each set of control points that are supposed to be identical. The matrix  $G$  is not needed and the overall number of unknowns is reduced, which allows better performance when solving the system. However, an index transformation is necessary to implement this system, i.e., each index  $\hat{i}$  in the global (full) control point list  $\hat{b}$  is assigned a new index  $\tilde{i}$  in the reduced list of control points  $\tilde{b}$ . Using a matrix  $H$  with entries  $H_{\hat{i},\tilde{i}} = 1$  and zero everywhere else, the original list of control points can be reconstructed as  $\hat{b} = H\tilde{b}$ . Plugging that in into the objective function (5) and solving for the gradient being equal zero leads to

$$\left( (1 - \lambda)H^T \hat{N}^T \hat{W} \hat{N} H + \lambda H^T \hat{M} H \right) \tilde{b} = (1 - \lambda)H^T \hat{N}^T \hat{W} \hat{p} \quad (7)$$

### 4.3.3 Achieving $G^1$ -continuity

The index transformation can be modified to also ensure  $G^1$ -continuity, i.e., transitions between surfaces with continuous tangent planes. Here, the method by Fan and Peters [4] is used. Given a quad mesh, they provide equations for a smooth surface model where all B-spline control points  $\hat{b}_i$  are expressed as linear combinations of the quad mesh vertices  $q$ . Using these equations and an appropriate indexing, the overall B-spline control points can be expressed as  $\hat{b} = Vq$ .

By considering the vertex positions  $q$  as unknowns and using the matrix  $V$  instead of the index transformation  $H$ , (7) can be rewritten as

$$\left( (1 - \lambda)V^T \hat{N}^T \hat{W} \hat{N} V + \lambda V^T \hat{M} V \right) q = (1 - \lambda)V^T \hat{N}^T \hat{W} \hat{p}. \quad (8)$$

Solving this linear system corresponds to finding a quad mesh on which the scheme by Fan and Peters produces a set of B-spline surfaces minimizing the objective function (5). The resulting surfaces are  $G^1$ -continuous everywhere.

## 5 Results

We applied our method to a number of isolated fibers. Figure 6 shows an example, including intermediate steps. Further results are shown in Figure 7. Even though many complicated topological situations arise in the data, our algorithm produces stable results, see Figure 8. The overall shape and important characteristics, like length and bending, are well-preserved. Loose noise-like structures occurring in regions with high roughness are smoothed,

while larger material parts are absorbed resulting in only a small distortion of the surface. Holes in the fiber walls are closed, and the topological structure of the surface does not impact the stability of the quadrilateral partitioning.

Each resulting B-spline surface is of order  $k = 3$  and is defined by  $8 \times 8$  control points. The surface model is constructed with the  $G^1$ -continuous approach described in Subsection 4.3.3, which defines the knot vector as  $\tau = (0, 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1, 1)$ . The computation is done via three iterations of surface approximation and parameter optimization, with decreasing smoothness parameters  $\lambda_1 = 0.9, \lambda_2 = 0.5, \lambda_3 = 0.1$ .

## 6 Conclusions

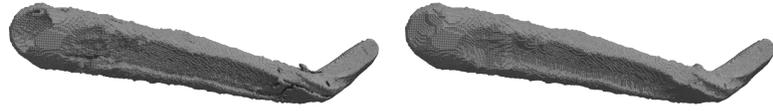
We have presented a new approach that reconstructs high-resolution cellulose fibers from 3D images with a low number of B-spline surfaces. Due to the production process of the material, the micro-structure contains complex topology that is problematic for established methods to handle properly. By splitting the processing pipeline into geometry and topology processing, we can deal with the topological difficulties and compute stable reconstructions of idealized structures.

After the data is processed with the pipeline, a B-spline surface model can be generated by using established B-spline methods. Our detailed description of the resulting linear system of equations focuses on an elegant implementation.

Our method can handle challenging situations in the image. For example, surface roughness has almost no impact. Even when larger material parts are present near the surface of a fiber, its wall is still reconstructed in a topologically correct way, and surplus material is treated as a local deformation of the surface. The triangulated meshes extracted from high-resolution images contain 423 628 to 1 203 844 triangles, and they are modeled with 236 to 676 B-spline surfaces.

We designed our method for fiber-like objects having simple overall topology to deal with local topological changes that arise in the presence of surface roughness or small holes and tunnels. Future work could address adaptations of our method to more complex shapes, e.g., chunks of connected fiber bundles.

**Acknowledgements** This research was funded by the Fraunhofer High Performance Center for Simulation- and Software-Based Innovation and supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 252408385 - IRTG 2057, as well as a grant by the Deutsch-Französische Hochschule (DFH). The wood insulation investigation was funded by the German Federal Ministry of Food and Agriculture (BMEL) through its Agency for Renewable Resources, project LowLambda.

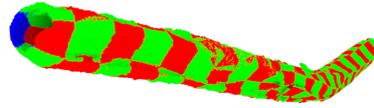


(a) Original data (553 748 triangles)

(b) Simplified surface



(c) Quadmesh (118 quads)



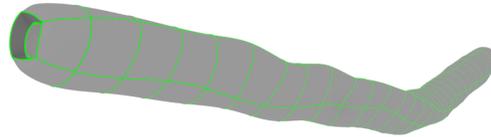
(d) Partition of original data



(e) B-spline model of inside surface



(f) B-spline model of outside surface

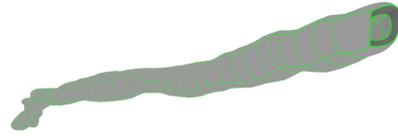


(g) Final B-spline model (236 B-spline surfaces)

Fig. 6: Main processing steps. (a) shows an isosurface of the original image. After applying a morphological closure operation, the topology is simplified (b). A quad-mesh is computed for this surface (c). The initial set of data points is partitioned by assigning each vertex to the nearest quad (d). A smooth B-spline construction is applied to model the fiber wall's inside (e) and outside surfaces (f). The final model is obtained by removing the B-spline surfaces at the end and linearly interpolating between the respective boundary curves (g).



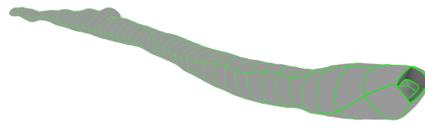
(a) Isosurface (1 203 844 triangles)



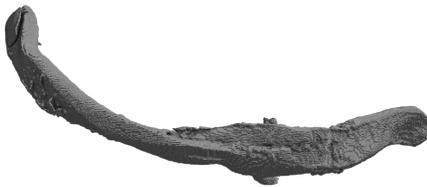
(b) B-spline model (676 B-spline surfaces)



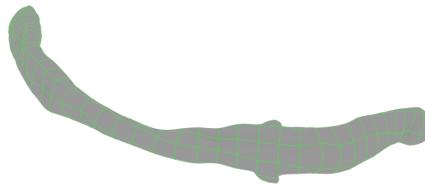
(c) Isosurface (423 628 triangles)



(d) B-spline model (432 B-spline surfaces)



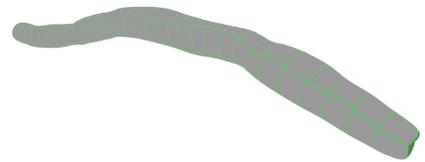
(e) Isosurface (710 808 triangles)



(f) B-spline model (580 B-spline surfaces)



(g) Isosurface (601 040 triangles)



(h) B-spline model (268 B-spline surfaces)

Fig. 7: Additional results. Isosurfaces of the exact image data (a), (c), (e), (g) and their B-spline approximations (b), (d), (f), (h).

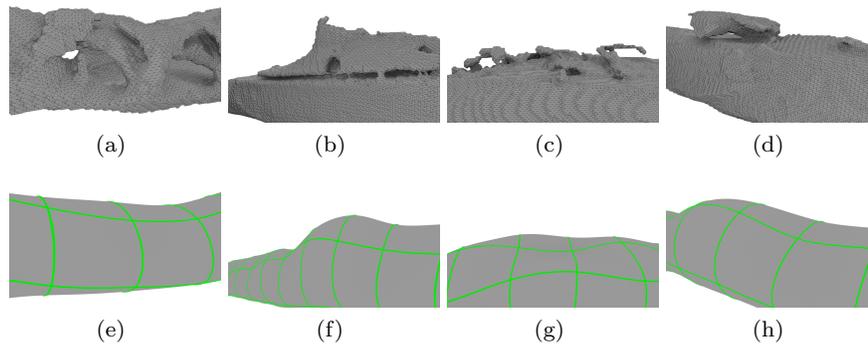


Fig. 8: Examples of data artifacts (a)-(d) and the B-spline approximations in those areas (e)-(h).

## References

1. E. Badel, C. Delisee, and J. Lux. 3D structural characterisation, deformation measurements and assessment of low-density wood fibreboard under compression: The use of X-ray microtomography. *Composites Science and Technology*, 68(7-8):1654–1663, 2008.
2. D. Dobrovolskij, M. Engelhardt, A. Rack, and K. Schladitz. Shape classification for wood based insulation material. Presentation, July 2019. contributed to ICTMS 2019, Cairns (Australia).
3. M. Eck and H. Hoppe. Automatic reconstruction of B-spline surfaces of arbitrary topological type. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 325–334. ACM, 1996.
4. J. Fan and J. Peters. Smooth bi-3 spline surfaces with fewest knots. *Computer-Aided Design*, 43(2):180–187, 2011.
5. G. Farin. *Curves and Surfaces for CAGD: A Practical Guide*. Morgan Kaufmann Publishers, 5th edition, 2002.
6. Fraunhofer ITWM, Department of Image Processing. MAVI – modular algorithms for volume images. <http://www.mavi-3d.de>, 2005. [Online; accessed 2019-09-23].
7. M. Godehardt, D. Mosbach, D. Roldan, and K. Schladitz. Efficient 3D erosion dilation analysis by sub-pixel EDT. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 243–255. Springer, 2019.
8. B. F. Gregorski, B. Hamann, and K. I. Joy. Reconstruction of B-spline surfaces from scattered data points. In *Proceedings Computer Graphics International 2000*, pages 163–170. IEEE, 2000.
9. G. Guennebaud, B. Jacob, et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010. [Online; accessed 2019-09-23].
10. H. Hagen and G. Schulze. Automatic smoothing with geometric surface patches. *Computer Aided Geometric Design*, 4(3):231–235, 1987.
11. J. Huang, Y. Zhou, M. Niessner, J. R. Shewchuk, and L. J. Guibas. QuadriFlow: A scalable and robust method for quadrangulation. In *Computer Graphics Forum*, volume 37, pages 147–160. Wiley Online Library, 2018.
12. A. Jacobson, D. Panozzo, et al. libigl: A simple C++ geometry processing library. <https://libigl.github.io/>, 2018. [Online; accessed 2019-09-23].

13. V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. In *SIGGRAPH*, volume 96, pages 313–324, 1996.
14. H. Lin, W. Chen, and H. Bao. Adaptive patch-based mesh fitting for reverse engineering. *Computer-Aided Design*, 39(12):1134–1142, 2007.
15. C. Marulier, P. J. Dumont, L. Orgéas, S. R. du Roscoat, and D. Caillerie. 3D analysis of paper microstructures at the scale of fibres and bonds. *Cellulose*, 22(3):1517–1539, 2015.
16. D. Mosbach, H. Hagen, M. Godehardt, and O. Wirjadi. Fast and memory-efficient quantile filter for data in three and higher dimensions. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 2928–2932. IEEE, 2014.
17. J. Ohser and K. Schladitz. *3D images of materials structures: processing and analysis*. John Wiley & Sons, 2009.
18. J. Peters. Constructing  $C^1$  surfaces of arbitrary topology using biquadratic and bicubic splines. In *Designing Fair Curves and Surfaces: Shape Quality in Geometric Modeling and Computer-Aided Design*, pages 277–293. SIAM, 1994.
19. J. Peters and J. Fan. On the complexity of smooth spline surfaces from quad meshes. *Computer Aided Geometric Design*, 27(1):96–105, 2010.
20. C. Peyrega, D. Jeulin, C. Delisée, and J. Malvestio. 3D morphological characterization of phonic insulation fibrous media. *Advanced Engineering Materials*, 13(3):156–164, 2011.
21. D. F. Rogers and N. Fog. Constrained B-spline curve and surface fitting. *Computer-Aided Design*, 21(10):641–648, 1989.
22. D.-J. Yoo. Three-dimensional surface reconstruction of human bone using a B-spline based interpolation approach. *Computer-Aided Design*, 43(8):934–947, 2011.
23. H. Yoshihara, T. Yoshii, T. Shibutani, and T. Maekawa. Topologically robust B-spline surface reconstruction from point clouds using level set methods and iterative geometric fitting algorithms. *Computer Aided Geometric Design*, 29(7):422–434, 2012.
24. X. Zhao, C. Zhang, L. Xu, B. Yang, and Z. Feng. IGA-based point cloud fitting using B-spline surfaces for reverse engineering. *Information Sciences*, 245:276–289, 2013.