Modeling the Internal Architecture of Composites

F. Conde-Rodríguez, Á.L. García-Fernández

Department of Computer Science, University of Jaén, EPSJ

J.C. Torres

Department of Software Engineering, University of Granada, ETSIIT

Abstract

This paper introduces a flexible and easy to use method for designing complex composite heterogeneous materials. These materials feature two distinct phases called *core* and *matrix* that remain separate and distinct. Moreover, composite materials have an internal microarchitecture that have to be precisely modeled. All the microarchitecture examples that are shown in this paper have been modeled in the same way, without any particular case nor having to use different implementation strategies or changing the source code depending on the microarchitecture.

The microarchitecture is modeled with a function that combines a material distribution function, which models the proportion of each phase at each point of the solid and determines the thickness of the core phase, and a cellular noise function based on distance fields that determines the shape, size, and distribution of the microarchitecture. Modeling the microarchitecture using two components gives our model great flexibility. In addition, it also allows to vary the size or thickness of the microarchitecture continuously inside the solid.

With this method, it is possible to model complex composite materials in which the phases (core and matrix) are in turn other composites with two distinct phases. Another important advantage of this method is that a complex object consisting of several different parts made of different materials can be modeled as a single computational object, which is very suitable for editing or computing simulations

Keywords: • **Computing methodologies** → **Volumetric models;** Parametric curve and surface models.

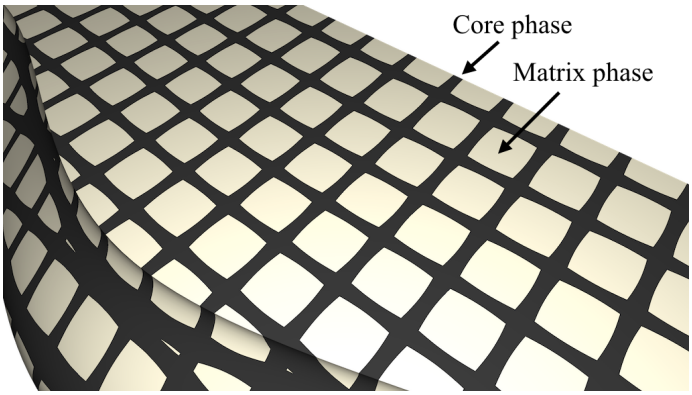


Figure 1: Example of composite material with a square cell microarchitecture. Core phase is coloured in black, and matrix phase is coloured in ivory

1. Introduction

Composite heterogeneous materials, or more briefly composites, are materials whose individual constituent components remain separate and distinct within the finished material. Those constituent materials are chemically dissimilar and separated by a distinct interface [1, 2]. In general, the constituent materials of a composite are called *core* and *matrix* phases (see Figure 1):

- The **core component** is usually a homogeneous material that acts as a structural element. It is typically arranged in structured patterns like honeycomb and wires, and this geometric design maximizes certain properties such as stiffness or strength, and minimizes others such as weight.
- The **matrix component** is continuous and surrounds the core. The main role of the matrix is to bind the core together and protect it from the environment.

One important aspect to be considered when designing new composite heterogeneous materials is its internal architecture. Material architecture refers to the way in which the different constituent materials are arranged. It dramatically affects the physical and mechanical properties of the new material. For example, a completely metallic panel is very rigid but may be too heavy. However, a fiberglass panel reinforced with a regular grid of metal cells retains a high rigidity while also being light [3].

Designing a composite heterogeneous material and its internal architecture is not an easy process. It is necessary to have the assistance of a computer application, based on a well stated framework, in order to address all the complex aspects of the design process, including the size, shape and distribution of the component substructures.

1.1. Main contributions of this paper

A comprehensive framework for modeling heterogeneous solids in a simple and efficient way using Bézier hyperpatches was introduced in [4]. This framework allows ensuring the continuity of both the geometry and the material distribution over

*Corresponding author
Email address: fconde@ujaen.es (F. Conde-Rodríguez)

the hyperpatch junctions when needed. Section 3 summarizes the basic concepts of this framework in order to make this paper self-contained.

This work extends the framework mentioned above, in order to make it able to represent composite heterogeneous materials and their internal microarchitecture in an accurate and highly configurable way. We present a method to create and edit the internal microarchitecture of a composite. Our method allows modeling a wide variety of microarchitectures (see some examples all over the paper), no matter its scale and thickness, in the same way, without the need for special cases nor different implementation strategies, i.e. the data model to represent the microarchitecture and the algorithm to evaluate it remain always the same, and it is not necessary to modify them depending on the specific features of a given desired microarchitecture.

This method also allows modeling complex composite materials in which the core and matrix phases are in turn other composites made of two phases. This way, it is possible to model, for example, a board with a square cell pattern made of a metallic material, and whose holes are filled with resin with suspended particles.

Another important advantage of the proposed method is the fact that a complex object, composed of several parts made of different materials, can be modeled as a single computational object. This makes our method very suitable for editing purposes or running simulations.

Moreover, the shape and distribution of the internal microarchitecture, as well as its thickness at a given point in the solid volume are not linked to its location and orientation in the modeling space $\mathbb{R}^3$. As a result, a geometric deformation of the solid does not affect its internal microarchitecture. In other words, each point in the solid volume preserves its microarchitecture no matter its position and orientation, or the solid deformations. This makes the modeling process fast and simple.

2. Previous work

In [5] Pasko et al. proposed several approaches for modeling both regular and irregular microstructures within the FRep framework. It is an interesting work that uses the periodic function *sin* combined by the intersection and union operators to obtain periodic regular architectures. They also proposed an approach to model microstructures based on cells. In this case, they first define a geometric model of the architecture and then replicate it along the FRep. We propose a simpler representation for the microarchitecture that works on the parametric domain instead of the Euclidean domain; this allows more control on the variation of features like the size or thickness of the microarchitecture over the solid volume. Moreover, our framework allows interactive editing of the models, making it very suitable for mechanical design applications.

In [6], Liu et al. presented an interesting method based on Markov Random Fields (MRF) to generate microstructures whose geometry is reconstructed from a small material sample image. They presented a method to solve the inverse problem of material modeling. The presented method allows them to generate a great variety of random heterogeneous materials. Our approach is different, as it allows to interactively model the solid microarchitecture either from scratch or from a small set of feature points (that could be obtained by sampling the target material), but in every case, the designer is enabled to change the microarchitecture by changing the properties and the location of these feature points in the solid volume.

In [7], Martínez et al. proposed an interesting method to directly model microarchitectures inspired by Voronoi open-cell foams. Their method does not require any further global optimization process, and the microarchitecture is directly generated. Their method is inspired by the seminal work on cellular solid textures by Worley [8], but it only applies to Voronoi-like microarchitectures. As an example, they apply the proposed method to the fabrication of objects with spatially varying elasticity. The approach that we apply in our proposal allows to produce a wider variety of microarchitectures, and provides more control on the microarchitecture features.

In [9] Elber et al. presented an interesting constructor for micro-structures based on the general idea of FFD (Free Form Deformations). The idea is to embed a geometric tile within a tri-variated solid. The tile can be a polygonal mesh, a set of parametric curves, a set of parametric surfaces or another tri-variated solid. The microstructures adapt to the shape of the tri-variated solid when it is deformed. They can model a broad set of microstructures straightforwardly. This work was extended by Massarwi et al. [10], adding support for fractal-like micro-structures. Compared to this approach, our proposal provides the designer with more control over the microarchitecture features, namely the shape, distribution and size of the phases, as well as the variation of these attributes over the solid volume. This allows us to model from completely random to completely regular material microarchitectures straightforwardly and in an interactive way.

In this work, we propose using heterogeneous Bézier hyperpatches to represent the volume of the solids. There are many references in the literature that deal with analogous mathematical representations for solid volumes; for example, Lin et al. propose an iterative fitting algorithm to interpolate a B-Spline solid (with Bézier conditions at the corners) from a tetrahedral mesh [11], in order to obtain a suitable representation for performing isogeometric analysis. Another interesting work dealing with tricubic B-Spline geometry is the one from Sasaki et al. [12]: they also produce a tricubic B-Spline model by means of an iterative fitting algorithm, but in this case, the input data is a random sample (thousands of points distributed all over the geometry of the solid) of the material properties. Sasaki et al. then use the B-Spline model to produce the volume slices and the toolhead movements for 3D printing. None of these works deals with the internal microarchitecture of the solids. Our work not only allows to interactively model the solid volume, but also its internal microarchitecture, hence enabling us to create enhanced solid representations.

3. Heterogeneous solid modeling

Our work is based on the framework for heterogeneous solid modeling presented in [4]. However, we are going to summarize in this section the basics of this framework, in order to make this paper self-contained. Please refer to the cited paper for a deeper insight on the framework.

Definition 1 (Material distribution). *Let* $\mathbf{M} = \{m_1, m_2, ..., m_n\}$ *be a set of primary materials and n its cardinality. The* material distribution *at any given point* $\mathbf{p}$ *in an heterogeneous solid is a tuple* $\mathbf{a} = (a_1, a_2, ..., a_n)$*, in which each* a_i *represents the volume fraction of the material* m_i *at that point.*

Given the definition of material distribution, the space of valid material distributions $\mathbf{V}$ is defined as

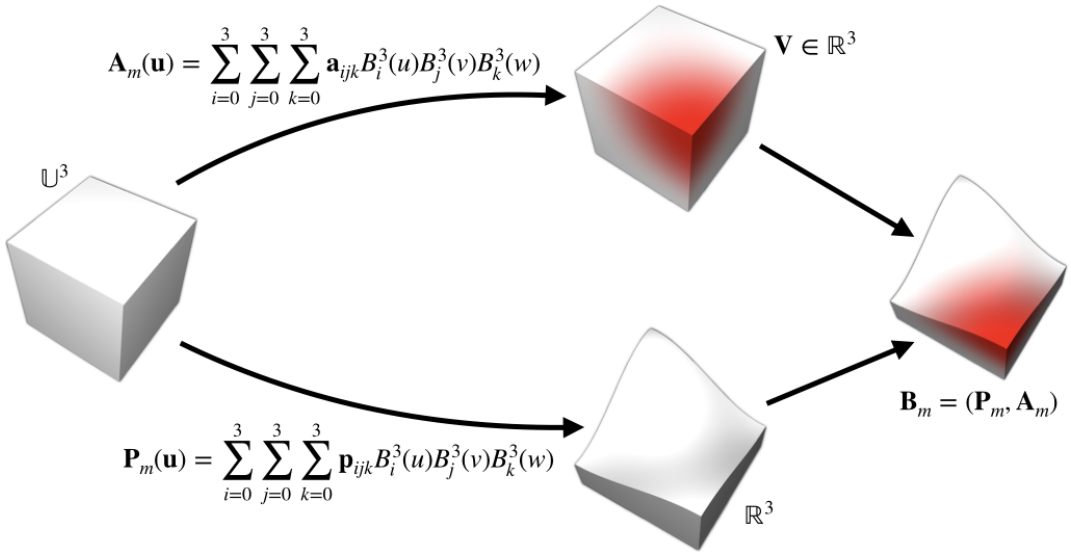


Figure 2: Projection of the parametric domain into the modeling and valid material distribution spaces as defined by an heterogeneous Bézier hyperpatch

$$\mathbf{V} = \left\{ \mathbf{a} \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} a_i = 1;\ a_i \geq 0 \right\} \tag{1}$$

Definition 2 (Heterogeneous geometric coefficient). *We define an heterogeneous geometric coefficient* $\mathbf{h}$ *as a pair formed by a 3D point and a valid material distribution*

$$\begin{aligned} \mathbf{h} &= (\mathbf{p}, \mathbf{a}) \\ \mathbf{p} &= (x, y, z) \in \mathbb{R}^3 \\ \mathbf{a} &= (a_1, a_2, ..., a_n) \in \mathbf{V} \end{aligned} \tag{2}$$

Heterogeneous geometric coefficients represent not only geometry, but also material properties. This way, a set of heterogeneous coefficients are used to define heterogeneous cells that will be used to describe solids:

Definition 3 (Heterogeneous Bézier Hyperpatch). *An heterogeneous Bézier hyperpatch* $\mathbf{B}_m$ *is a volume defined by the function*

$$\begin{gathered} \mathbf{B}_m(u, v, w) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} \mathbf{h}_{ijk} B_i^3(u) B_j^3(v) B_k^3(w) \\ u, v, w \in [0, 1] \end{gathered} \tag{3}$$

where the $\mathbf{h}_{ijk}$ *are the heterogeneous geometric coefficients that describe the shape and material distribution along the volume of the hyperpatch, and* B_i^3, B_j^3 *and* B_k^3 *are the well-known Bézier blending functions.*

An heterogeneous Bézier hyperpatch is therefore defined by $4 \times 4 \times 4 = 64$ heterogeneous geometric coefficients. It defines two projections from a 3D solid parametric domain (a unit size cube in which every point $\mathbf{u}$ is univocally identified by a tuple (u, v, w) such that $u, v, w \in [0, 1]$): a projection into the *modeling space* (i.e. the Euclidean space $\mathbb{R}^3$) that is a set of points $\mathbf{P}_m$, and a projection $\mathbf{A}_m$ into the *space of valid material distributions*, that describes the mixture of materials $\mathbf{a}$ at each point $\mathbf{p} \in \mathbf{P}_m$ (see Figure 2).

Definition 4 (Heterogeneous solid). *An heterogeneous solid* $\mathbf{B}$ *is a finite collection of n pairwise disjoint heterogeneous Bézier hyperpatches*

$$\begin{gathered} \mathbf{B} = \{\mathbf{B}_1, \mathbf{B}_2, ...\mathbf{B}_n\} \\ \forall\, \mathbf{P}_i \in \mathbf{B}_i,\ \forall\, \mathbf{P}_j \in \mathbf{B}_j,\ i \neq j,\ \mathbf{P}_i \cap^* \mathbf{P}_j = \emptyset \end{gathered} \tag{4}$$

In other words, this framework uses a cell decomposition scheme to represent solids, and each of these cells is a heterogeneous Bézier hyperpatch. In the rest of the paper we will use either *hyperpatch* or *cell* indistinctly to refer to the elements that compose an heterogeneous solid.

Although our framework works on the basis of a decomposition using heterogeneous Bézier hyperpatches, the concepts and algorithms proposed in this work could be adapted to other types of hyperpatch (trivariate B-Spline, for example), or to Bézier hyperpatches with higher degree. However, tricubic Bézier hyperpatches provide us with enough flexibility to model complex figures, and at the same time their evaluation and associated calculations are simpler.

Representing heterogeneous solids with this framework allows to accurately model not only the shape, but also the material distribution along the solid volume, due to the fact that both properties depend on the parametric coordinates of each point in the solid, and these do not change under modeling operations. Moreover, heterogeneous geometric coefficients provide a simple and intuitive edition tool for designers.

The heterogeneous geometric coefficients that influence the most on the shape and material of the hyperpatches are those located at the corners of the control point net of each hyperpatch. These points are labeled as *corner points*. Continuity conditions on both the shape and the material distribution along the joints between hyperpatches depend on the alignment and material distribution values of these corner points respectively, as described in [4].

In order to avoid deformations in the internal structure of the hyperpatches that might cause faulty material distributions, a validity condition is imposed on the location of the heterogeneous geometric coefficients [13, 4]. The rest of the paper assumes that the hyperpatches comply with this condition.

Taking this framework as foundation, in the following sections

we introduce a new method for modeling the material microarchitecture of heterogeneous solids represented with this model.

4. Modeling microarchitecture

This section introduces our proposal of a new method to model material microarchitecture, based on the framework summarized in the previous section. As explained in section 3, each point in an heterogeneous Bézier hyperpatch $\mathbf{B}_m$ is identified by a three parameter tuple $\mathbf{u} = (u, v, w)$, such that $u, v, w \in [0, 1]$. For simplicity, in this section we are going to use $\mathbf{u}$ to refer to the solid points.

4.1. Basic process

Let $\mathbf{M}$ be a set of two materials $\mathbf{M} = \{core, matrix\}$. We define a function $m(\mathbf{u}) : \mathbb{U}^3 \rightarrow \mathbf{M}$ to characterize the microarchitecture present at a given point $\mathbf{u}$ of an heterogeneous Bézier hyperpatch $\mathbf{B}_m$. The function $m(\mathbf{u})$ returns the material (*core* or *matrix*) that is present at that point.

The function $m(\mathbf{u})$ is in turn based on two functions $f_{core}(\mathbf{u})$ and $f_{mat}(\mathbf{u})$ whose domain is $\mathbb{U}^3$ and whose range is the closed interval $[0, 1]$.

The function $f_{core}(\mathbf{u}) : \mathbb{U}^3 \rightarrow [0, 1]$ defines the shape and distribution of the core component of the composite, i.e. the internal structure of the core phase of the material. Therefore, $f_{core}(\mathbf{u})$ determines the general shape of the composite architecture.

The function $f_{mat}(\mathbf{u}) : \mathbb{U}^3 \rightarrow [0, 1]$ defines the probability of finding the core or the matrix material phases at the point $\mathbf{u}$ in $\mathbf{B}_m$. It is a cutoff function that determines the thickness of the core phase with respect to the matrix one.

Definition 5 (Microarchitecture function). *We define the microarchitecture function $m(\mathbf{u}) : \mathbb{U}^3 \rightarrow \mathbf{M}$ as:*

$$m(\mathbf{u}) = \begin{cases} core & if \quad f_{mat}(\mathbf{u}) < f_{core}(\mathbf{u}) \\ matrix & if \quad f_{mat}(\mathbf{u}) \geq f_{core}(\mathbf{u}) \end{cases} \tag{5}$$

$$\forall\ \mathbf{u} = (u, v, w) \in \mathbb{U}^3$$

Each point $\mathbf{u} = (u, v, w) \in \mathbb{U}^3$ such that $f_{mat}(\mathbf{u}) < f_{core}(\mathbf{u})$ is assumed to be occupied by the core material and similarly, each point $\mathbf{u} \in \mathbb{U}^3$ such that $f_{mat}(\mathbf{u}) \geq f_{core}(\mathbf{u})$ is assumed to be occupied by the matrix material (see Algorithm 1).

Algorithm 1 Evaluate microarchitecture at point $\mathbf{u}$ of a hyperpatch $\mathbf{B}_m$

Input: Value of the cutoff function $f_{mat}(\mathbf{u})$.
Input: Value of the core function $f_{core}(\mathbf{u})$ at the point $\mathbf{u}$
Output: Value of the microarchitecture function $m(\mathbf{u})$ at the point $\mathbf{u}$. It is the material (core or matrix) that best represents the material mixture at the point $\mathbf{u}$

```
if f_mat(u) < f_core(u) then
    Return core material
else
    Return matrix material
end if
```

Figure 3 shows a simple 1D example of the process. In this example, the functions $f_{core}(u)$ and $f_{mat}(u)$ have been defined as:

$$f_{core}(u) = \frac{sin(2\pi x)}{2} + 0.5\ , \quad f_{mat}(u) = 0.6$$

Our work aims at achieving a framework that allows modeling straightforwardly a range of microarchitectures as wide as possible, without any particular case nor having to use different implementation strategies for developing the functions for each microarchitecture. Therefore, we have chosen as $f_{core}(\mathbf{u})$ the cellular noise function based on distance fields published by Worley in [8]. This function is widely used for generating procedural textures.

As Worley describes, the cellular noise function requires a set of **feature points FP** $= \{\mathbf{fp}_1, \mathbf{fp}_2, ..., \mathbf{fp}_n\}$, located in the parametric space $\mathbb{U}^3$. In our framework, the amount and distribution of these feature points is freely chosen by the solid designer, and determine the material microarchitecture. Note that the feature points are independent from the heterogeneous geometric coefficients of the hyperpatches. The decision on the position and feature values on these points is up to the solid designer; this provides the designer with a very powerful semantic tool to define the microarchitecture of the solid material.

Given a hyperpatch $\mathbf{B}_m$, the value of the function $f_{core}(\mathbf{u})$ for a point $\mathbf{u}$ in $\mathbf{B}_m$ is computed in the following way: first of all, the distances from $\mathbf{u}$ to every feature point in the parametric space of the hyperpatch $\mathbb{U}^3$ are computed. Then, these distances (named **basis functions**) are sorted, so that $F_1(\mathbf{u})$ is the shortest distance and $F_n(\mathbf{u})$ is the longest one. Then, the value of $f_{core}(\mathbf{u})$ is computed as a linear combination of the basis functions, but as Worley states in his work, it is enough with the first four basis functions to get a good result (see Figure 4):

$$f_{core}(\mathbf{u}) = c_1F_1 + c_2F_2 + c_3F_3 + c_4F_4 \tag{6}$$

Due to the way they are defined, both the basis functions and the $f_{core}(\mathbf{u})$ function are continuous in $\mathbb{U}^3$. Moreover, as the calculations are performed in the parametric space, the actual position of the point in the modeling space $\mathbb{R}^3$ is not relevant.

As $f_{core}(\mathbf{u})$ represents a material fraction, it is expected to have a value in the range [0,1]. Therefore, for those cases in which the choice of values for the coefficients c_i can result in $f_{core}(\mathbf{u})$ producing values outside the [0,1] range, it is necessary to normalize these values. To do so, $f_{core}(\mathbf{u})$ is evaluated at a large enough number of random points in $\mathbb{U}^3$, and its maximum (v_{max}) and minimum (v_{min}) values are stored for performing the normalizing operation in subsequent function calls. The calculation of the minimum and maximum values should only be done once, since their value does not change if the feature point set **FP** or the coefficients c_i are not changed.

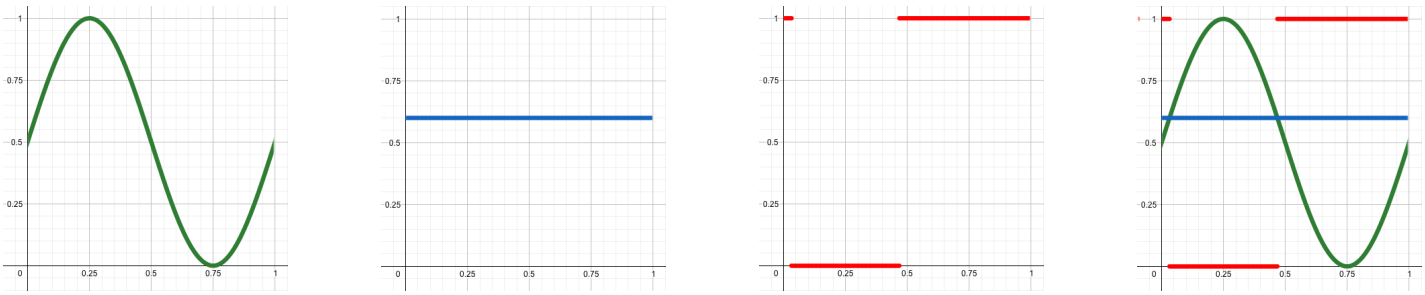
Definition 6 (Normalized core function). *The normalized core function $\bar{f}_{core}(\mathbf{u})$ is defined as:*

$$\bar{f}_{core}(\mathbf{u}) = \frac{f_{core}(\mathbf{u}) - v_{min}}{v_{max} - v_{min}} \tag{7}$$

Regarding the function $f_{mat}(\mathbf{u}) : \mathbb{U}^3 \rightarrow [0, 1]$, remember that it is a cutoff function that defines the probability of finding the core or the matrix material phases at the point $\mathbf{u}$ in $\mathbf{B}_m$. This function determines the thickness of the core phase with respect to the matrix one. In other words, the function $f_{mat}(\mathbf{u})$ represents the material distribution function $\mathbf{A}_m(\mathbf{u})$ of the hyperpatch $\mathbf{B}_m$, with a material set of two elements.

As explained in section 3, a material distribution is a tuple of as many values as elements in the material set. For the case of composite materials, valid material distributions consist of two-tuples (a_{core}, a_{matrix}), that can be rewritten as $(a_{core}, 1 - a_{core})$. Therefore, the function $f_{mat}(\mathbf{u})$ is defined for our purposes as:

$$f_{mat}(\mathbf{u}) = \mathbf{A}_m(\mathbf{u}) = a_{core} \tag{8}$$



(a) $f_{core}(u)$ (b) $f_{mat}(u)$ (c) $m(u)$ (d) Process

Figure 3: Simple 1D example that shows how the microarchitecture is calculated. a) $f_{core}(\mathbf{u})$ represents the distribution of the core composite. b) $f_{mat}(\mathbf{u})$ represents the overall proportion between the core and the matrix phases. c) $m(\mathbf{u})$ represents the presence of the core (1) or matrix (0) phases at the point $\mathbf{u}$. d) Superposition of the three functions

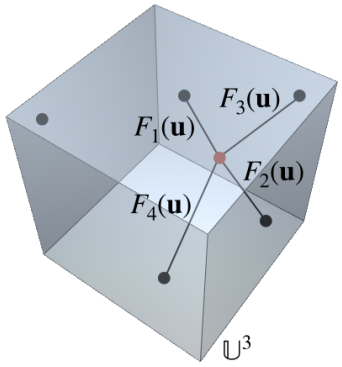
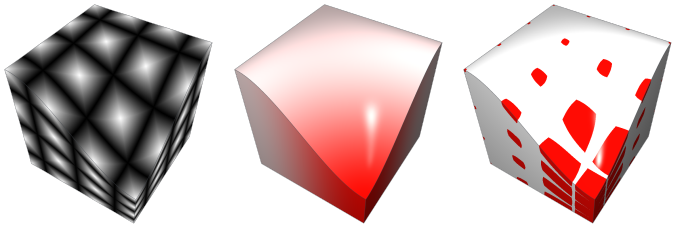


Figure 4: Example of feature point set (in black) and first four basis functions $F_i(\mathbf{u})$ for a given point $\mathbf{u} \in \mathbb{U}^3$ (in red)



(a) (b) (c)

Figure 5: Functions used to define the microarchitecture. a) Cellular noise function $\bar{f}_{core}(\mathbf{u})$ based on distance fields. It defines the shape and distribution of the core component. b) Cutoff function $f_{mat}(\mathbf{u})$. It defines the material distribution (i.e. the probability of finding the core or the matrix phases) at a given point. The size of the core phase with respect to the matrix phase depends on its value. c) $\bar{f}_{core}(\mathbf{u})$ and $f_{mat}(\mathbf{u})$ are combined into the material function $m(\mathbf{u})$. It allows us to define easily and accurately the shape, size and distribution of the internal architecture of the composite material

Figure 5 illustrates the process we present for modeling the internal microarchitecture of composite materials.

As the cutoff function $f_{mat}(\mathbf{u})$ can vary continuously over each hyperpatch, it allows us to modify the thickness of the core phase with respect to the matrix phase. This way, we can model pieces made of composite materials with a thicker or thinner material microarchitecture where needed, in order to, for example, reduce the weight or any other physical property (see Figure 6).

Using this framework, it is possible to model very different material microarchitectures from a fixed feature point set just by changing the values of the coefficients c_i in $\bar{f}_{core}(\mathbf{u})$ (see Figure 7). For example: if $c_1 = 1$ and the other coefficients are set to 0, then $\bar{f}_{core}(\mathbf{u}) = F_1(\mathbf{u})$, and the resulting microarchitecture consists of spheres surrounding the feature points; if $c_1 = -1$, $c_2 = 1$ and $c_3 = c_4 = 0$, then $\bar{f}_{core}(\mathbf{u}) = F_2(\mathbf{u}) - F_1(\mathbf{u})$, and the resulting microarchitecture mimics the Voronoi diagram of the feature points.

Our method can also model composite materials with cavities, since one of the two materials, *core* or *matrix*, can be set to vacuum.

4.2. Continuity

In our framework, an heterogeneous solid is a cell decomposition in which each cell is an Heterogeneous Bézier Hyperpatch (See definitions 3 and 4).

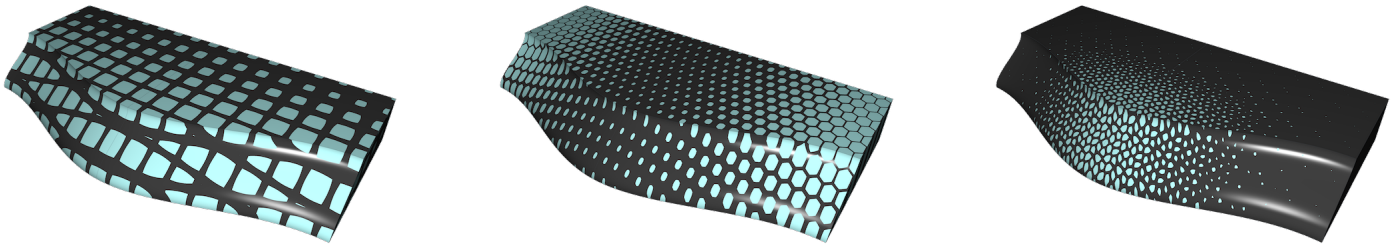
The continuity inside each cell is guaranteed by $\bar{f}_{core}(\mathbf{u})$ and $f_{mat}(\mathbf{u})$. The basis functions $F_i(\mathbf{u})$, from which $\bar{f}_{core}(\mathbf{u})$ is obtained, are calculated as distances from $\mathbf{u}$ to some feature point $\mathbf{fp}_i$. As $\mathbf{u}$ changes, $F_i(\mathbf{u})$ varies smoothly since the distance between $\mathbf{u}$ and the feature point also varies smoothly. On the other hand, $f_{mat}(\mathbf{u}) = \mathbf{A}_m(\mathbf{u})$ is also continuous inside the cell, as it is the convex combination of the 64 valid material distributions $\mathbf{a}_{ijk}$ associated to the heterogeneous geometric coefficients, using the cubic Bézier blending functions B^3 as weights.

As stated before, the material function $m(\mathbf{u})$ is composed of two functions: $\bar{f}_{core}(\mathbf{u})$ that determines the shape and size of the microarchitecture, and $f_{mat}(\mathbf{u})$ that determines its thickness. Both functions are independent, hence it is possible to configure the continuity of both aspects separately. This allows modeling very interesting situations such as those shown in Figure 8.

Moreover, our framework can also guarantee the continuity of the microarchitecture at the junction between two adjacent cells in those areas of the solid object where they are needed. The function $\bar{f}_{core}(\mathbf{u})$ is intrinsically continuous between cells, because the distance calculation algorithm considers the parametric domains of contiguous cells as connected. In addition, the function $f_{mat}(\mathbf{u}) = \mathbf{A}_m(\mathbf{u})$ can be defined to be continuous or discontinuous at the junction between cells as needed (see [4]) by assigning the same or a different material distribution $\mathbf{a}$ to the heterogeneous geometric coefficients that are next to the corner points of the hyperpatches. This allows modeling discontinuities in the microarchitecture wherever it is necessary, as shown in Figure 8.

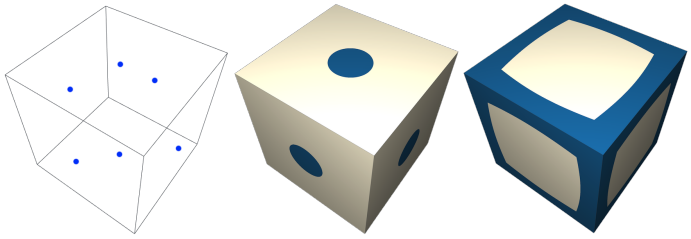
4.3. Periodicity

The function $\bar{f}_{core}(\mathbf{u})$ as defined above determines the microarchitectural pattern of the composite material. This pattern is applied as a unit to the hyperpatch, but it is very easy to make it



(a) (b) (c)

Figure 6: Several examples of pieces in which the thickness of the material microarchitecture varies over the solid volume. These changes are achieved through the assignment of different material distribution values to the heterogeneous geometric coefficients $\mathbf{h}_{ijk}$ of the hyperpatches



(a) (b) (c)

Figure 7: Different microarchitectures computed with the same feature point set and changing the values of the coefficients c_i. As can be seen, the same feature point set with different coefficients results in very different microarchitectures. a) Feature point set. In this example, we have placed a feature point in the center of each of the six faces of the unit-size cube defined by the parametric space $\mathbb{U}^3$. b) $\bar{f}_{core}(\mathbf{u}) = F_1(\mathbf{u})$. c) $\bar{f}_{core}(\mathbf{u}) = F_2(\mathbf{u}) - F_1(\mathbf{u})$.

repeatable over the hyperpatch volume by substituting the value of $\mathbf{u} = (u, v, w)$ with the following $\mathbf{u}'$:

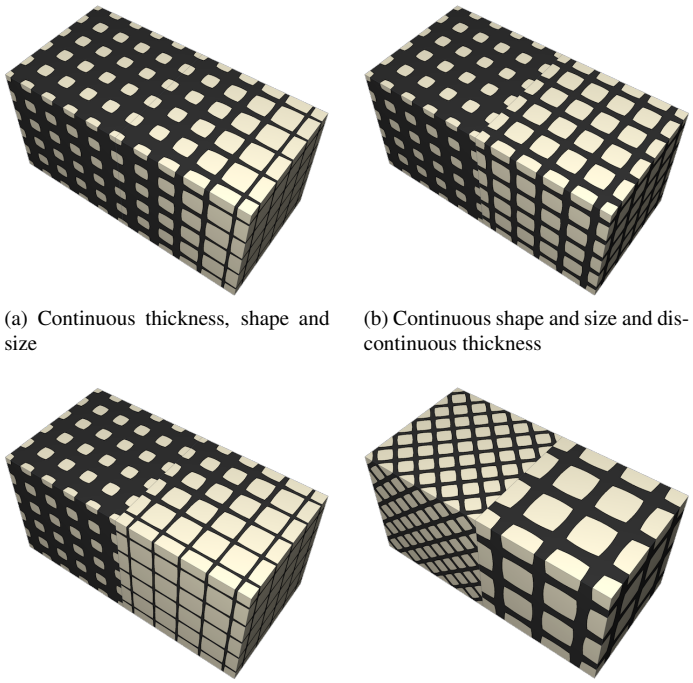
$$\mathbf{u}' = (u', v', w') = (fract(u \cdot pr_u), fract(v \cdot pr_v), fract(w \cdot pr_w)) \tag{9}$$

in which the values pr_u, pr_v and pr_w are the periods (i.e. number of times the pattern is repeated in each parametric direction, see Figure 9), and $fract$ is a function that returns the fractional part of its parameter. The use of $fract$ assures that the value of u', v' and w' will always belong to the range $[0, 1)$. Moreover, the period values can be associated to the heterogeneous coefficients $\mathbf{h}_{ijk}$, so that the actual values of pr_u, pr_v and pr_w are interpolated over the hyperpatch volume (see Figure 10) using the Bézier blending functions in the same way as with the material distributions and the shape.

The possibility of having independent variable period values for each parametric direction all over each hyperpatch makes our framework extremely flexible, allowing the designer to easily adapt the material microarchitecture to the specific needs of the modeled solid.

4.4. Abrupt termination

Sometimes it is necessary that the core phase of a composite material occupy all the volume at some locations of the solid (to create a reinforcement or a rigid border, for example, see Figure 11 b) and c)). In order to be able to model these situations with our framework, we have added to the heterogeneous geometric coefficient definition a new element, $\mathbf{l}$ that consists of an upper and a lower limit for the function $\bar{f}_{core}(\mathbf{u}')$, namely l_{max} and l_{min}. These limits are applied to the calculation of the material at a



(a) Continuous thickness, shape and size

(b) Continuous shape and size and discontinuous thickness

(c) Continuous shape and size and mixed continuity in thickness

(d) Discontinuous thickness, shape and size

Figure 8: Continuity between cells. Our framework allows us to choose the type of continuity at each corner point. In addition, it is possible to establish continuity in size and shape and thickness of the microarchitecture separately

given point $\mathbf{u}' \in \mathbf{B}_m$ by replacing the evaluation of the function $\bar{f}_{core}(\mathbf{u}')$ with:

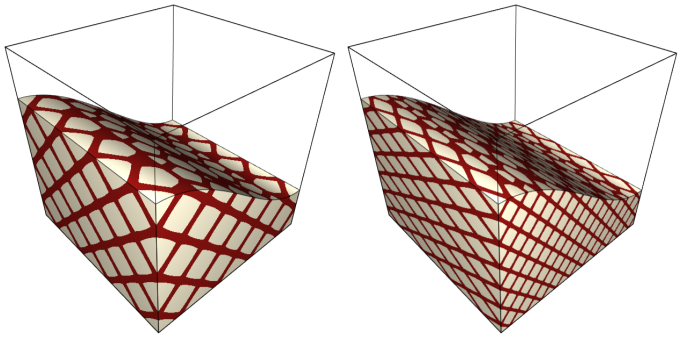
$$\bar{f}'_{core}(\mathbf{u}') = min\,(\,l_{max},\, max\,(\,l_{min}, \bar{f}_{core}(\mathbf{u}')\,)\,) \qquad l_{min} \in [0,1] \qquad l_{max} \in [0,1] \qquad l_{min} < l_{max} \tag{10}$$

Therefore, the function $m(\mathbf{u}')$ for choosing the material to be used at a given point in a hyperpatch becomes:

$$m(\mathbf{u}') = \begin{cases} core & if \quad f_{mat}(\mathbf{u}') < \bar{f}'_{core}(\mathbf{u}') \\ matrix & if \quad f_{mat}(\mathbf{u}') \geq \bar{f}'_{core}(\mathbf{u}') \end{cases} \tag{11}$$

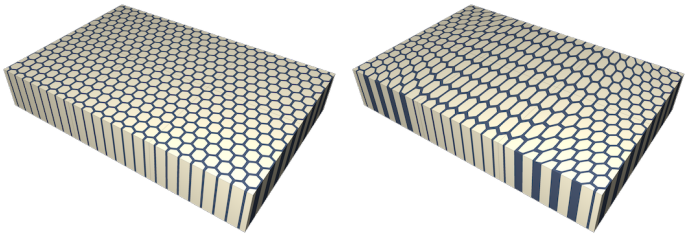
$$\forall\ \mathbf{u}' = (u', v', w') \in \mathbb{U}^3$$

Due to the way heterogeneous geometric coefficients are defined and used, the limits l_{min} and l_{max} are also interpolated over the hyperpatch volume, in the same way as the other coefficient



(a) (b)

Figure 9: Effect of the periods on a hyperpatch microarchitecture a) The periods in u, v and w are equal ($pr_u = pr_v = pr_w$). b) The periods in u, v and w are different ($pr_u \neq pr_v \neq pr_w$)



(a) (b)

Figure 10: The microarchitecture periods in u, v and w can be interpolated over the hyperpatch volume by assigning different period values to the heterogeneous geometric coefficients $\mathbf{h}_{ijk}$

properties. This provides the designer with the capability of easily modifying the thickness of the reinforcement or border if needed (see Figure 11 d)).

Finally, the new definition of heterogeneous geometric coefficient as described in sections 4.3 and 4.4 is the following:

Definition 7 (Heterogeneous geometric coefficient). *We define an heterogeneous geometric coefficient for composites* $\mathbf{h}$ *as a tuple formed by a 3D point, a valid material distribution, a triad of periods in each parametric direction and the limit values for the material distribution.*

$$\begin{aligned} \mathbf{h} &= (\mathbf{p}, \mathbf{a}, \mathbf{pr}, \mathbf{l}) \\ \mathbf{p} &= (x, y, z) \in \mathbb{R}^3 \\ \mathbf{a} &= (a_1, a_2, ..., a_n) \in \mathbf{V} \\ \mathbf{pr} &= (pr_u, pr_v, pr_w) \\ \mathbf{l} &= (l_{min}, l_{max}) \end{aligned} \tag{12}$$

$$pr_u \in [1, \infty] \quad pr_v \in [1, \infty] \quad pr_w \in [1, \infty]$$
$$l_{min} \in [0, 1] \quad l_{max} \in [0, 1] \quad l_{min} < l_{max}$$

4.5. 2D and 3D composites

The process described so far produces 3D microarchitectures, however, there are 2D microarchitectures that are also very useful. A 2D microarchitecture varies only in two of the three parametric directions u, v, w, and remains exactly the same along the remaining parametric direction. The honeycomb microarchitecture is a typical example of 2D microarchitecture. Figure 12 shows an example of 2D and 3D microarchitecture.

Given a hyperpatch in which a set of feature points **FP** have been defined, a 2D microarchitecture can be computed by setting the parametric direction d in which the microarchitecture will be deployed. Then, the feature points are projected into a 2D parametric plane $\mathbb{U}^2$ by discarding their d coordinate, and the basis functions F_i are computed as the 2D distances between the projected feature points and the projection (also obtained by discarding its d coordinate) of the point $\mathbf{u}$ on the same parametric plane. Figure 13 shows this process for obtaining the microarchitecture from Figure 12 left; in this example, d is set to w.

Algorithm 2 Evaluate a material composite $m(\mathbf{u})$ at a point $\mathbf{u}$ in an heterogeneous Bézier hyperpatch $\mathbf{B}_m$

Input: Parametric coordinates of the point $\mathbf{u} = (u, v, w) \in \mathbb{U}^3$
Input: Set of 4×4×4 = 64 heterogeneous geometric coefficients $\mathbf{h}_{ijk}$, $i, j, k \in \{1, 2, 3, 4\}$ of the hyperpatch $\mathbf{B}_m$
Input: Composite feature point set
Input: Coefficients c_i, $i \in \{1, 2, 3, 4\}$ for the linear combination of the composite basis functions
Input: Set of materials $\mathbf{M} = \{core, matrix\}$
Output: Material (*core* or *matrix*) at the point $\mathbf{u}$

1: Compute $\sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} \mathbf{h}_{ijk} B_i^3(u) B_j^3(v) B_k^3(w)$ to obtain the interpolated values $\mathbf{h} = (\mathbf{p}, \mathbf{a}, \mathbf{pr}, \mathbf{l})$ at $\mathbf{u}$
2: $\mathbf{u}' \leftarrow (\, fract\,(u * pr_u),\ fract\,(v * pr_v),\ fract\,(w * pr_w)\,)$
3: Compute $F_n(\mathbf{u}'), \quad i \in \{1, 2, 3, 4\}$
4: $\bar{f}_{core}(\mathbf{u}') \leftarrow c_1 F_1(\mathbf{u}') + c_2 F_2(\mathbf{u}') + c_3 F_3(\mathbf{u}') + c_4 F_4(\mathbf{u}')$
5: $\bar{f}'_{core}(\mathbf{u}') \leftarrow min\,(\, l_{max},\ max\,(\, l_{min}, \bar{f}_{core}(\mathbf{u}')\,)\,)$
6: **if** $\mathbf{a} < \bar{f}'_{core}(\mathbf{u}')$ **then**
7: Return material *core*
8: **else**
9: Return material *matrix*
10: **end if**

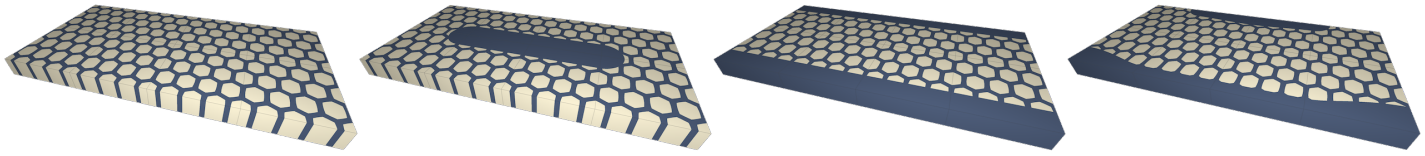
4.6. Recapitulation

Algorithm 2 summarizes the whole process for computing the material (core or matrix) at a given point $\mathbf{u}$ in an heterogeneous hyperpatch $\mathbf{B}_m$.

As shown in the algorithm, a material composite can be evaluated with a reduced, fixed-size set of data and simple, mostly vectorial operations on 2 and 3 component vectors. Moreover, the data and the calculations depend only on the hyperpatch being evaluated, and therefore, the algorithm can be easily parallelized using modern GPUs, dramatically improving its performance.

5. Crafting complex composites

There are two main aspects that have to be considered when designing complex composites. The first one is the proper choice of the feature point set and the coefficients c_i for the $f_{core}(\mathbf{u})$ function. The second one is the fact that our framework allows each composite material (*core* or *matrix*) to be in turn composites made of another two materials (see Figure 14); this provides great modeling flexibility. Let us review these two aspects in detail.



(a) (b) (c) (d)

Figure 11: Effect of applying limits to the function $f(\mathbf{u})$. a) Reference model with honeycomb microarchitecture and no limits. b) Using limits in order to create an internal reinforcement. c) Using limits in order to create homogeneous borders. d) Using varying limits in order to create borders with changing thickness. Different limit values are assigned to each heterogeneous geometric coefficient; the interpolation of these values over the hyperpatch volume result in a continuous series of thickness changes in the border

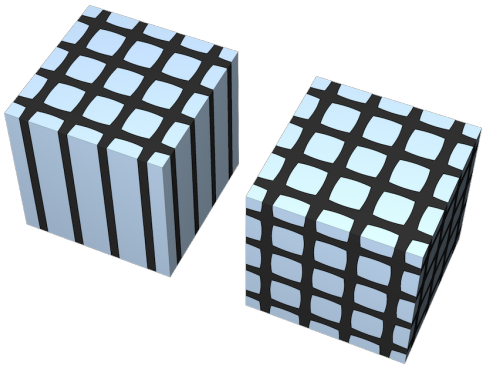


Figure 12: Composite material with 2D microarchitecture (left) and with 3D microarchitecture (right)

point	coordinates	point	coordinates
$\mathbf{fp}_1$	(0.0, 0.0, 0.0)	$\mathbf{fp}_2$	(0.0, 1.0, 0.0)
$\mathbf{fp}_3$	(1.0, 0.0, 0.0)	$\mathbf{fp}_4$	(1.0, 1.0, 0.0)
$\mathbf{fp}_5$	(0.0, 0.0, 1.0)	$\mathbf{fp}_6$	(0.0, 1.0, 1.0)
$\mathbf{fp}_7$	(1.0, 0.0, 1.0)	$\mathbf{fp}_8$	(1.0, 1.0, 1.0)
$\mathbf{fp}_9$	(0.0, 0.5, 0.5)	$\mathbf{fp}_{10}$	(1.0, 0.5, 0.5)
$\mathbf{fp}_{11}$	(0.5, 0.0, 0.5)	$\mathbf{fp}_{12}$	(0.5, 1.0, 0.5)
$\mathbf{fp}_{13}$	(0.5, 0.5, 0.0)	$\mathbf{fp}_{14}$	(0.5, 0.5, 1.0)

Table 1: Location of the feature points in the cubic parametric domain $\mathbb{U}^3$ according to the face-centered cubic crystal structure. There is one feature point $\mathbf{fp}_i$ at each of the eight corners of the domain, and another one at the center of each of its faces. This feature point set allows us to create interesting microarchitectures

5.1. Design of a composite material

In order to achieve useful microarchitectures, the feature points have to be conveniently distributed over the parametric domain of the hyperpatch $\mathbb{U}^3$. Small changes on the position of the feature points can produce significant differences in the shape and distribution of the microarchitecture. A good starting point can be to use known distributions, like the *face-centered cubic crystal structure* that appears in some metals. Table 1 shows the location of the feature points according to this distribution.

Once the feature points have been located, it is necessary to choose the coefficients c_i that define the $f_{core}(\mathbf{u})$ function. The same feature point set can produce very different microarchitectures just by changing the values of these coefficients. Figure 7 b) and c) show two very different microarchitectures that are obtained from the same feature point set but using different coefficients.

As explained in section 4.1, it is necessary for $f_{core}(\mathbf{u})$ to have

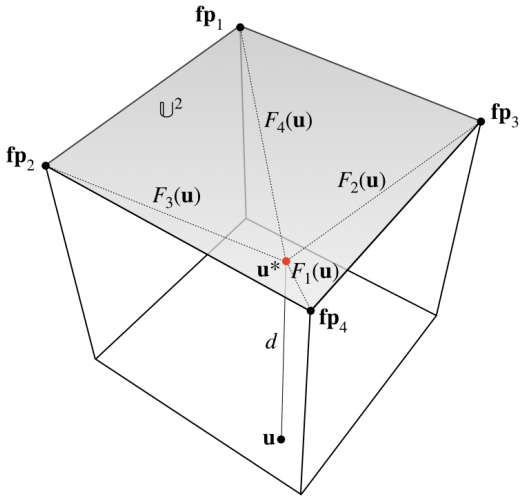
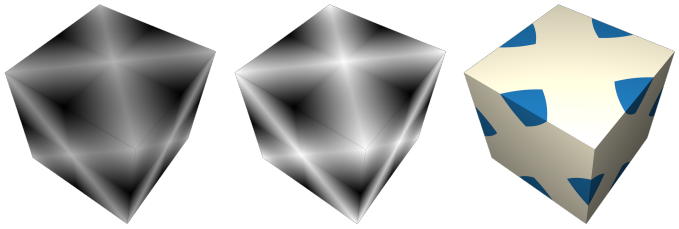


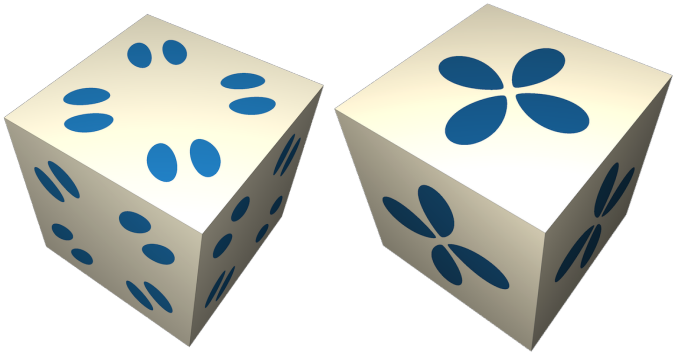
Figure 13: Example of calculation of 2D microarchitecture. The deployment direction d is set to w. In this example, the feature points are located at the corner points of the cell

Figure 14: Example of complex composite. This composite is made of a core phase with honeycomb architecture and a matrix phase which is in turn a composite with fiber architecture and with higher core phase density at the endings.The microarchitecture is 2D as described in Section 4.5. The feature points for the honeycomb are located at the coordinates {(0, 0),(1/2, 0),(1, 0),(1/4, 1/2),(3/4, 1/2),(0, 1),(1/2, 1),(1, 1)} and periodic in $\mathbf{u}$ and $\mathbf{v}$, with $f_{core}(\mathbf{u}) = F2 - F1$. The fiber architecture was created with the feature point set {(0, 0), (1, 0), (0, 1), (1, 1), (1/14, 8/16), (2/14, 4/16), (3/14, 12/16), (4/14, 2/16), (5/14, 10/16), (6/14, 6/16), (7/14, 14/16), (8/14, 1/16), (9/14, 9/16), (10/14, 5/16), (11/14, 13/16), (12/14, 3/16), (13/14, 11/16)}, and $f_{core}(\mathbf{u}) = F2 - F1$



(a) (b) (c)

Figure 15: Effect of normalizing the value of the function $f_{core}(\mathbf{u})$. a) Representation of the not-normalized value of $f_{core}(\mathbf{u})$. b) Representation of the normalized value of $\bar{f}_{core}(\mathbf{u})$. The maximum and minimum values used for the normalization were computed empirically. c) Microarchitecture obtained with the function $\bar{f}_{core}(\mathbf{u}) = 2F_3(\mathbf{u}) - F_2(\mathbf{u}) - F_1(\mathbf{u})$. The feature point set used in this example is distributed according to the face-centered cubic crystal structure (Table 1)



(a) (b)

Figure 16: Examples of material microarchitectures obtained by applying different $\bar{f}_{core}(\mathbf{u})$ functions. a) Microarchitecture obtained with the function $\bar{f}_{core}(\mathbf{u}) = F_1(\mathbf{u}) + F_2(\mathbf{u}) + F_3(\mathbf{u})$. b) Microarchitecture obtained with the function $\bar{f}_{core}(\mathbf{u}) = 3F_1(\mathbf{u}) - F_2(\mathbf{u}) + F_3(\mathbf{u}) + F_4(\mathbf{u})$. The feature point set used in these examples is distributed according to the face-centered cubic crystal structure (Table 1)

a value in the range $[0, 1]$, and this is achieved by normalizing its value using the maximum and minimum values that it may produce (see Definition 6). In order to compute these extreme values, a random sampling of $f_{core}(\mathbf{u})$ is performed, and the maximum and minimum values obtained are conveniently stored. Our experiments show that 100000 samples are enough to get values close enough to those extreme values.

Table 2 shows several core functions and the maximum and minimum values used for normalizing them. The feature point set is distributed according to the face-centered cubic crystal structure in all the cases.

Figure 15 c) shows the microarchitecture obtained using the function $\bar{f}_{core}(\mathbf{u}) = 2F_3(\mathbf{u}) - F_2(\mathbf{u}) - F_1(\mathbf{u})$. Figure 15 a) represents the not-normalized value of $f_{core}(\mathbf{u})$, while Figure 15 b) represents the normalized value of the same function. Figure 16 shows another two examples using different $\bar{f}_{core}(\mathbf{u})$ functions from Table 2

5.2. *Designing a complex composite material*

As stated before, the *core* and *matrix* materials of a composite can be either primary materials or other composites made in turn of other primary or composite materials, and so on (see Figure 17).

A **primary material** has physical and/or chemical properties of interest, like colour, reflection index, thermal conductivity or

$f_{core}(\mathbf{u})$	min	max
F_1	0.001306	0.496692
F_2	0.353736	0.701418
$F_2 - F_1$	0.000000	0.701052
$F_1 + F_2$	0.707107	0.993436
F_3	0.408409	0.705434
$2F_3 - F_2 - F_1$	0.000296	0.705607
$F_1 + F_2 + F_3$	1.224746	1.494825
F_4	0.434643	0.746173
$3F_4 - F_3 - F_2 - F_1$	0.000296	0.705607
$F_1 + F_2 + F_3 + F_4$	1.732099	2.119050

Table 2: Some examples of $f_{core}(\mathbf{u})$ functions, and their maximum and minimum values for a feature point set distributed according to the face-centered cubic crystal structure (Table 1)

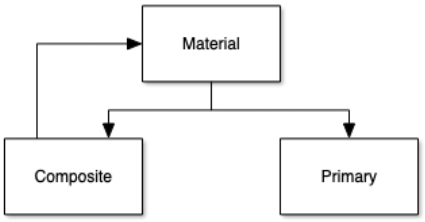


Figure 17: A material can be either a primary material or a composite. Composite materials in turn are made of two materials that may be primary or composites. This way, very complex materials can be easily designed

breaking strength, among others. Primary materials can be used in more than one hyperpatch; therefore, in our implementation, the materials are stored separately, and the cells keep references to them.

A **composite material** consists of:

- Two materials, that can be either primary or other composites.
- A set of feature points **FP**, so that each composite that is used in a hyperpatch can have a different internal microarchitecture. It is important to note that the same set of feature points could be used for more than one composite, and that is the reason why in our implementation, these sets are stored separately. There is a reference from each cell to the set of feature points for each composite it is made of.
- A set of coefficients c_i, as well as the maximum and minimum values that will be used for evaluating the material function.

For each different material in a hyperpatch $\mathbf{B}_m$ the following data is necessary:

- The **material** information. It can be a primary or a composite material (see Figure 17). Materials can be shared by more than one hyperpatch, therefore, each hyperpatch references the materials it needs.
- The material-related components of the $4 \times 4 \times 4 = 64$ heterogeneous geometric coefficients of the hyperpatch:

$(\mathbf{a}_{ijk}, \mathbf{pr}_{ijk}, \mathbf{l}_{ijk})$, with $i, j, k \in \{0, 1, 2, 3\}$. These values determine the thickness, period and limits of the material microarchitecture. Different materials can have different values for these components.

The geometry-related component of each heterogeneous geometric coefficient, p_{ijk}, is stored only once.

These two blocks of data allow to compute the functions $\bar{f}'_{core}(\mathbf{u})$ and $f_{mat}(\mathbf{u})$ that are necessary to produce the microarchitecture. The value of the function $\bar{f}'_{core}(\mathbf{u})$ is obtained from the composite material data, while the value of the function $f_{mat}(\mathbf{u})$ is obtained from the material-related components of the heterogeneous geometric coefficients of the hyperpatch.

All the composite material examples shown in this paper have been evaluated in the same way, as described in Algorithm 2. There are no special cases nor situations that require specific treatment. This provides our model with great flexibility, and allows us to easily model complex composites.

6. Results

Figure 18 shows the modeling process of a typical fiber-reinforced material. Figure 19 shows different alternatives that can be obtained from the initial model, just by changing some of the parameter values.

To prove the validity of the proposed framework, we have modeled several complex objects made of composites with different microarchitectures (see Figures 20 and 21) that mimic different types of materials with and without reinforcements. The advantages of our framework are:

- The materials are assigned on a per-cell basis. Each cell has a reference to its material, and the neighbouring cells may have a reference to the same material or to a completely different one.
- The edition of the geometry is independent from the material edition.
- Different primary materials and composites can be easily used in the same solid model.
- Although the materials may differ from one cell to the other, the whole object model is handled as a single heterogeneous solid. This simplifies tasks like edition or simulation.

All the figures in this paper were created using a 2014 MacBook Pro computer with a 4-core 2,5 GHz Intel Core i7 processor, 16GB of RAM memory and an NVIDIA GeForce GT750M graphics card with 2GB of memory. The rendering algorithm is implemented in GPU using Metal. In order to do this, it is necessary to tesselate the hyperpatches, but it is not necessary to generate a very detailed triangulation, as the parametric coordinates of each fragment are passed as input to the shaders (together with the rest of the microarchitecture related parameters). The evaluation in GPU of the functions described in the previous sections determines the pixel colour. The visualization algorithm runs smoothly at a frame rate of 30fps with 4K image resolution, allowing interactive editing of the models.

7. Discussion and conclusions

This paper presents a framework for modeling the internal microarchitecture of composite heterogeneous materials to be used for representing heterogeneous solids. The solids are represented by means of a cell decomposition, in which each cell is in turn modeled as an heterogeneous Bézier hyperpatch. The main advantages of this framework are the following:

- The shape, size, and volume fraction of the phases at a point in a cell is not linked to the position or orientation that the cell occupies in the modeling space $\mathbb{R}^3$, but to the parametric representation of the point in the parametric domain $\mathbb{U}^3$. This is because we use different functions for mapping the parametric domain $\mathbb{U}^3$ to the geometry and the material (see Figure 2). As a result, geometric deformations on the solid do not affect its internal microarchitecture (see Figure 22). This makes the modeling process easier.
- Our framework allows modeling and editing interactively a wide range of microarchitectures for composite materials in the same way, no matter their shape, size, scale and thickness. None of the examples shown in this paper has required different implementation strategies, nor special treatment.
- The data required for evaluating the material function $\mathbf{m}(\mathbf{u})$ is very small. Moreover, the algorithms are simple, mostly based on vector operations. This makes it very suitable for its implementation in GPU. Executing the algorithms in GPU has allowed us to obtain frame rates clearly over 30fps with 4K image resolution, making it possible to interactively edit the models.
- It is possible to model complex composite materials in which either the *core* or the *matrix*, or both, are in turn other composites made of two materials. See Figures 14, 20 and 21 for examples.
- A complex object, composed of different parts made of distinct materials, can be represented as a single computational model. This way, both editing and running numerical simulations on the object is straightforward.
- The main features of the material microarchitecture, like period or thickness, are computed from the heterogeneous geometric coefficients $\mathbf{h}_{ijk}$ of the hyperpatches. This information is interpolated over the volume of each hyperpatch, and this provides a very powerful and intuitive tool for editing. Assigning material properties to specific zones in the solid is easy for a user.

Acknowledgements

This work has been partially funded by the Spanish Ministry of Economy and Competitiveness through grants TIN2017-85259-R and TIN2017-84968-R with ERDF funds.

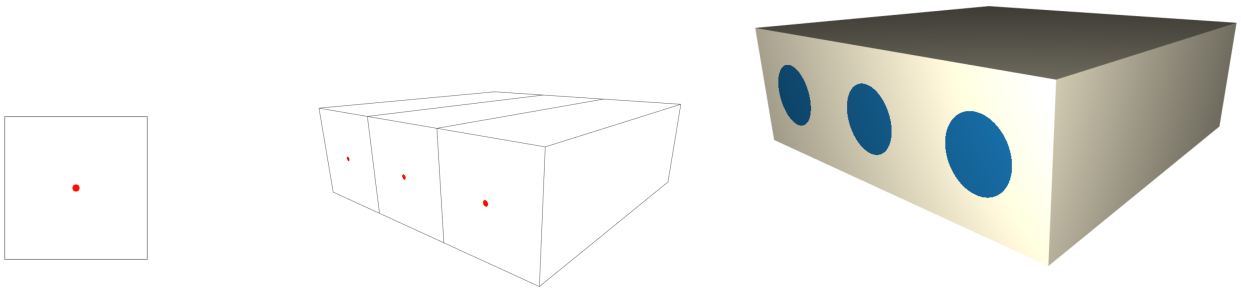
References

[1] F. Matthews, R. Rawlings, Composite Materials: Engineering and Science, CRC Press, 1999.

[2] F. Campbell, Structural Composite Materials, ASM International, 2010.

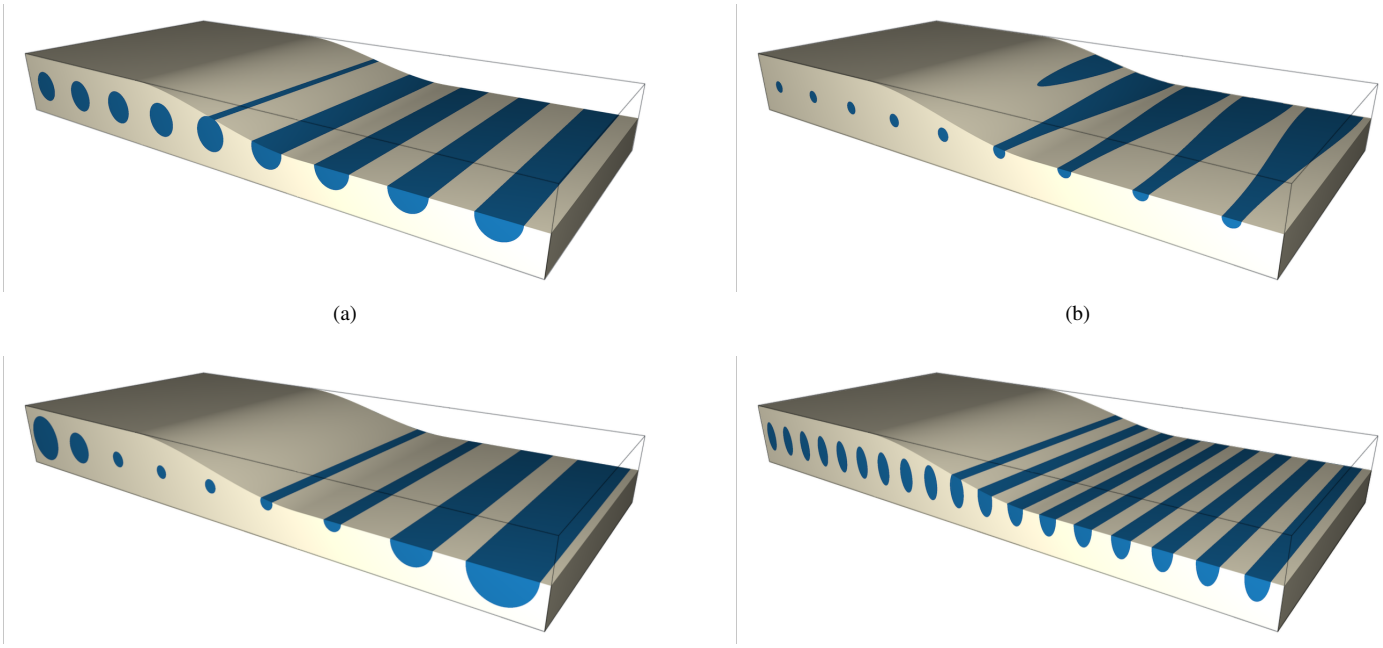
[3] W. Regli, J. Rossignac, V. Shapiro, V. Srinivasan, The new frontiers in computational modeling of material structures, Comput.-Aided Des. 77 (2016) 73–85. doi:10.1016/j.cad.2016.03.002.

[4] F. Conde-Rodríguez, J.-C. Torres, Á.-L. García-Fernández, F.-R. Feito-Higueruela, A comprehensive framework for modeling heterogeneous objects, The Visual Computer 33 (1) (2017) 17–31. doi:10.1007/s00371-015-1149-0.



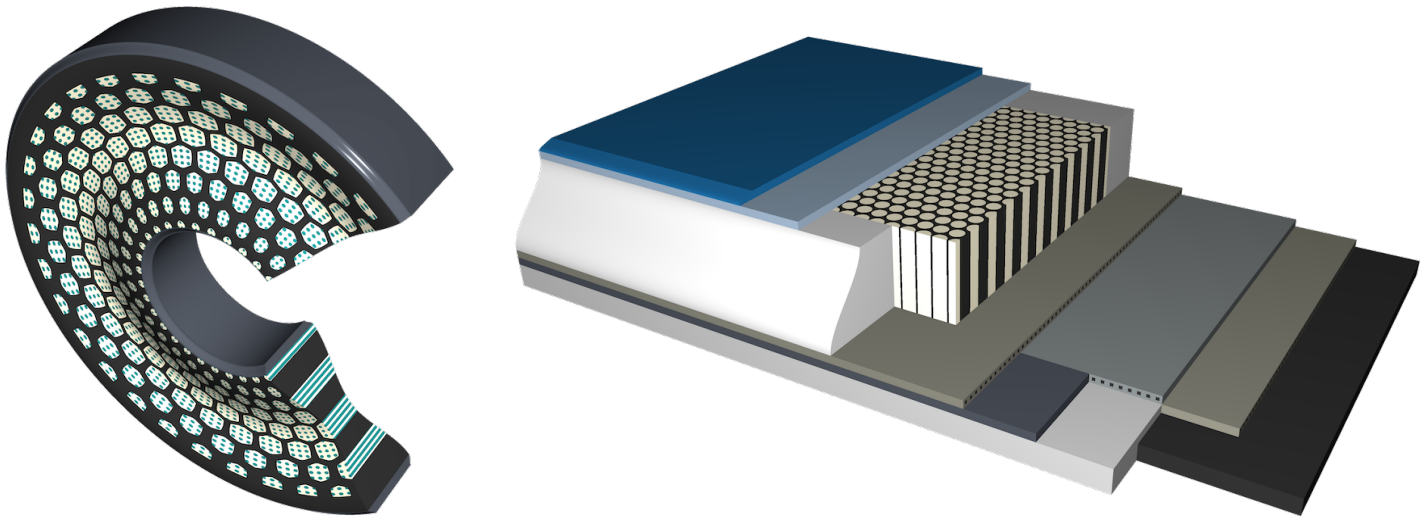
(a) (b) (c)

Figure 18: A typical fiber-reinforced material has a 2D microarchitecture that can be easily modeled. a) The feature point set only contains the point (0.5,0.5). b) Then, the period is set as desired (in this example $pr_u = 3$), and finally, $f_{core}(\mathbf{u})$ is defined as needed (in this case, $f_{core}(\mathbf{u}) = F_1$); the material distribution for all the heterogeneous geometric coefficients in this case is (0.5, 0.5). The final result is shown in (c)



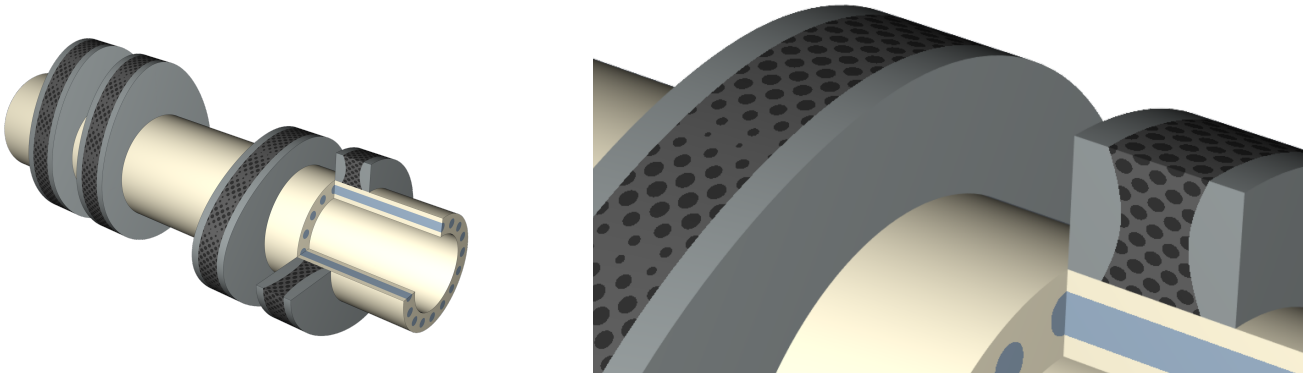
(a) (b) (c) (d)

Figure 19: Broken out sections showing that a) starting from a typical fiber-reinforced material modeled as shown in Figure 18 (with $pr_u = 9$), different microarchitectures can be easily modeled by changing the parameters. In b), the material distribution **a** for the four front-facing corner points is (0.2, 0.8), and (0.8, 0.2) for the four back-facing corner points. In c), the 16 heterogeneous geometric coefficients located on the left and the right sides have material distribution (0.8, 0.2), while the 32 heterogeneous coefficients in the middle have material distribution (0, 1). Finally, in d) the period value is doubled with respect to the original (i.e. $pr_u = 18$)



(a) (b)

Figure 20: Two examples of heterogeneous solids made of several materials. a) Cut section of a solid made of four primary materials. The inner and outer rings are made of a single material (in grey), while the filling between them is made of a composite with honeycomb microarchitecture. The core phase of the composite is made of an homogeneous material (in black) with an increased thickness near the rings, and a reinforced edge next to the outer ring, and the matrix phase of the composite is in turn another composite made of two materials (in blue and ivory); the microarchitecture of this composite consists of fibers distributed in a regular pattern. b) Cut section of a real ski; skis are made of several composites in order to increase rigidity, while keeping low weight. *(Note: the thickness of some parts has been increased to allow of a better viewing)*



(a) (b)

Figure 21: a) Model of a camshaft with a broken out section showing the internal microarchitecture of the materials. b) Close up view of the material microarchitectures

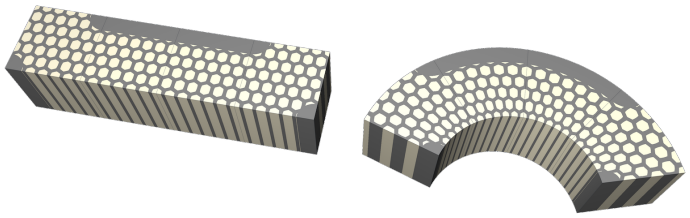


Figure 22: Geometric deformations of a solid do not affect the internal microarchitecture of the material. Left: original model. Right: deformed model. The material microarchitecture remains stable

[5] A. Pasko, O. Fryazinov, T. ., P. Fayolle, V. Adzhiev, Procedural function-based modelling of volumetric microstructures, Graphical Models 73 (2011) 165–181. doi:10.1016/j.gmod.2011.03.001.

[6] X. Liu, V. Shapiro, Random heterogeneous materials via texture synthesis, Computational Materials Science 99 (2015) 177 – 189. doi:https://doi.org/10.1016/j.commatsci.2014.12.017.

[7] J. Martínez, J. Dumas, S. Lefebvre, Procedural voronoi foams for additive manufacturing, ACM Trans. Graph. 35 (4). doi:10.1145/2897824.2925922.

[8] S. Worley, A cellular texture basis function, in: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96, ACM, New York, NY, USA, 1996, pp. 291–294. doi:10.1145/237170.237267.

[9] G. Elber, Precise construction of micro-structures and porous geometry via functional composition, in: M. Floater, T. Lyche, M.-L. Mazure, K. Mørken, L. L. Schumaker (Eds.), Mathematical Methods for Curves and Surfaces, Springer International Publishing, Cham, 2017, pp. 108–125. doi:https://doi.org/10.1007/978-3-319-67885-6_6.

[10] F. Massarwi, J. Machchhar, P. Antoln, G. Elber, Hierarchical, random and bifurcation tiling with heterogeneity in micro-structures construction via functional composition, Computer-Aided Design 102 (2018) 148–159. doi:10.1016/j.cad.2018.04.017.

[11] H. Lin, S. Jin, Q. Hu, Z. Liu, Constructing B-spline solids from tetrahedral meshes for isogeometric analysis, Computer Aided Geometric Design 35-36 (2015) 109–120. doi:10.1016/j.cagd.2015.03.013.
URL http://www.sciencedirect.com/science/article/pii/S0167839615000369

[12] Y. Sasaki, M. Takezawa, S. Kim, H. Kawaharada, T. Maekawa, Adaptive direct slicing of volumetric attribute data represented by trivariate B-spline functions, The International Journal of Advanced Manufacturing Technology 91 (5) (2017) 1791–1807. doi:10.1007/s00170-016-9800-0.
URL https://doi.org/10.1007/s00170-016-9800-0

[13] F. Conde-Rodriguez, J.-C. Torres-Cantero, A Simple Validity Condition for B-Spline HyperpatchesAccepted: 2015-11-11T18:52:49Z Publisher: Eurographics Association. doi:http://dx.doi.org/10.2312/egs.20011010.
URL https://diglib.eg.org:443/xmlui/handle/10.2312/egs20011010