





# Hybrid Function Representation for Heterogeneous Objects

A. Tereshin <sup>1</sup>  A. Pasko <sup>2</sup>  O. Fryazinov <sup>1</sup>  V. Adzhiev <sup>1</sup> 

<sup>1</sup>National Centre for Computer Animation, Bournemouth University, United Kingdom

<sup>2</sup>Skolkovo Institute of Science and Technology, Russia

---

## Abstract

*Heterogeneous object modelling is an emerging area where geometric shapes are considered in concert with their internal physically-based attributes. This paper describes a novel theoretical and practical framework for modelling volumetric heterogeneous objects on the basis of a novel unifying functionally-based hybrid representation called HFRep. This new representation allows for obtaining a continuous smooth distance field in Euclidean space and preserves the advantages of the conventional representations based on scalar fields of different kinds without their drawbacks. We systematically describe the mathematical and algorithmic basics of HFRep. The steps of the basic algorithm are presented in detail for both geometry and attributes. To solve some problematic issues, we have suggested several practical solutions, including a new algorithm for solving the eikonal equation on hierarchical grids. Finally, we show the practicality of the approach by modelling several representative heterogeneous objects, including those of a time-variant nature.*

**Keywords:** *hybrid representation, distance fields, eikonal solver, function representation, heterogeneous objects, volumetric modelling, time-variant objects.*

---

## 1. Introduction

Heterogeneous volumetric object modelling is a rapidly developing field and has a variety of different applications. Volume modelling is concerned with computer representation of object surface geometry as well as its interior. Homogeneous volume modelling, better known as solid modelling, deals with volume interior uniformly filled by a single material. Heterogeneous object is a volumetric object with interior structure where different physically-based attributes are defined, e.g. spatial different material compositions, micro-structures, colour, density, etc. [KT07, LFSL20]. This type of objects is widely used in applications where the presence of the interior structures is an important part of the model. Additive manufacturing, physical simulation and visual effects are examples of such applications.

The most widely used representations for defining heterogeneous objects are boundary representation, distance-based representations, function representation and voxels. Boundary representation (BRep) [LFAB14] maintains its prevailing role due to its numerous well-known advantages. It works well in solid modelling for objects consisting of a set of polygonal surface patches stitched together to envelope the uniform and homogeneous structure of its material. However, BRep is not inherently natural for dealing with heterogeneous objects, especially in the context of additive manufacturing and 3D printing [LEM\*17], where volume-based multi-material properties are paramount as well as in physical simulation

where the exact representation rather than an approximate one can be important [NMK\*06].

On the contrary, volumetric representations in the form of voxels [WYJ\*11] are more natural for defining such heterogeneous objects as they are based on volumetric grids. Voxels represent an object as a set of cubic cells at which the geometry along with the object attributes are defined. However, this representation essentially approximates both the geometry model and the material distribution in interior of the object as their definition is limited by the resolution of the voxel grid.

On the other hand, function-based, and more specifically, distance-based representations are able to represent the object and its interior structure in both continuous and discrete forms [JBS06]. They are exact, embrace a wide range of geometric shapes and naturally define many physically-based attributes. There are a lot of well-established operations for these representations. Most of them provide distances to the object surface. However, distance functions (DFs) are not essentially continuous, they can have medial gradient discontinuities and are not necessarily smooth. This potentially results in non-watertight surfaces, and in artefacts, such as creases, after applying some operations, for instance, blending and metamorphosis, which are important for many applications. Undesired artefacts (stresses, creases, etc.) can also appear as the result of defining distance-based attribute functions.

We consider function-based and distance-based representations as a promising conceptual and practical scheme to deal with het-

52 erogeneous objects, especially in the context of a number of topi- 103  
 53 cal application areas concerned with exact volume-based geomet- 104  
 54 ric modelling, animation, simulation and fabrication. However, the 105  
 55 existing representational schemes of that type appear in many vari- 106  
 56 ations and the field as a whole exhibits a rather fragmented and 107  
 57 not properly formalised suite of methods. There is an obvious need 108  
 58 for a properly substantiated and unifying theoretical and practical 109  
 59 framework. This challenge can be considered in the context of the 110  
 60 emergence of new representational paradigms suitable for the ma- 111  
 61 turing applications, such as modelling of material structures, that 112  
 62 was outlined and substantiated in [RRSS16]. 113

63 In this work we propose a novel function-based representational 114  
 64 scheme. We introduce a mathematical framework called hybrid 115  
 65 function representation for defining a heterogeneous volumetric ob- 116  
 66 ject with its attributes in continuous and discrete forms. It is based 117  
 67 on hybridisation of several DF-based representations that unifies 118  
 68 their advantages and compensates for their drawbacks. This repre- 119  
 69 sentational scheme aims at dealing with heterogeneous objects with 120  
 70 some specific time-variant properties important in physical simula- 121  
 71 tions related to both geometry and attributes. The idea was initially 122  
 72 tested in a short paper [TAFP19] where the scheme unifying the 123  
 73 function representation (FRep) and the signed distance functions 124  
 74 (SDFs) had been sketchily outlined. 125

75 The contributions of our work can be formulated as follows: 126

- 76 • We provide a thorough survey of the relevant representations 127  
 77 aiming at their classification and identifying their advantages and 128  
 78 drawbacks. We formalise the notions of the adaptively sampled 129  
 79 distance fields (ADFs) and interior distance fields (IDFs). 130
- 80 • On the basis of an analysis of the well-established FRep and DF- 131  
 81 based representations, namely SDFs, ADFs and IDFs, we formu- 132  
 82 late the requirements for a novel unifying hybrid representation 133  
 83 called HFRep. 134
- 84 • We propose a mathematically substantiated theoretical descrip- 135  
 85 tion of the HFRep with an emphasis on defining functions for 136  
 86 HFRep objects' geometry and attributes. 137
- 87 • We describe a basic algorithm for generating HFRep objects in 138  
 88 terms of their geometry and attributes, and develop its main steps 139  
 89 in a detailed step-by-step manner. 140
- 90 • We identify the problematic issues associated with several steps 141  
 91 of the basic algorithm and propose several practical solutions. In 142  
 92 particular, we present a novel hierarchical fast iterative method 143  
 93 for solving the eikonal equation on hierarchical grids in 2D. The 144  
 94 developed algorithm was used for generating HFRep based on 145  
 95 FRep and ADF. 146

## 96 2. Related works

97 There is a huge body of works dealing with different aspects of 152  
 98 representational schemes for volumetric heterogeneous objects. In 153  
 99 this section we first concentrate on those works that deal with rep- 154  
 100 resentations for geometric shapes. Then, we consider some existing 155  
 101 hybrid representations. The basic methods for defining attributes in 156  
 102 interior of volumetric objects are also reviewed. 157

### 2.1. Geometry representations

An overall object geometry can be represented by boundary sur-  
 faces or by any other solid representational scheme including pro-  
 cedurally defined scalar fields.

Boundary representation (BRep) remains the most popular rep-  
 resentation. It can be described by a polygonal or other surface  
 model. Polygonal models can be represented as nested polygonal  
 meshes bounding the regions with different material density val-  
 ues [LFAB14]. This representation scheme has the following prob-  
 lems: the possible presence of holes or gaps in a mesh, normals  
 can be flipped, triangles in the mesh can be intersecting or overlap-  
 ping with each other, polygonal shells can be noisy. Among many  
 polygon-based approaches applicable to heterogeneous objects we  
 pay attention to the diffusion surfaces introduced in [TSNI10]. This  
 approach deals with 3D surfaces with colours defined on both sides,  
 such that the interior colours in the volume are obtained by dif-  
 fusing colors from nearby surfaces. It was used for modelling ob-  
 jects with rotational symmetry. It is efficient to compute, but cross-  
 sections of the mesh obtained with further triangulation could suf-  
 fer from discretisation artefacts.

Another way to represent the overall object geometry is con-  
 structive solid geometry (CSG) [Req80]. Originally all solids were  
 homogeneous, but later primitives could carry on some information  
 that can be interpreted as a material index [Bow95]. The operations  
 on attributes corresponding to set-theoretic operations were pro-  
 vided.

The most widely used method for defining heterogeneous ob-  
 jects is the voxel representation [WYJ\*11, BKW\*18]. The object  
 is subdivided into multiple cubic cells with defined geometric and  
 attribute parts in them. However, geometric and attribute properties  
 are essentially approximated according to the voxel grid resolution.

In the context of this work we pay a special attention to defin-  
 ing a heterogeneous object geometry using different types of scalar  
 fields. The most common schemes of that type are already men-  
 tioned FRep [PASS95], SDFs [JBS06], ADFs [FPRJ00] as well as  
 the shape aware distance fields which are represented by functions  
 that we call interior distance functions (IDFs). We will discuss them  
 in more details in the next section.

Another widely used approach for obtaining a continuous dis-  
 tance based definition of the object is to compute the solution of  
 the optimal mass transportation [SRGB14]. This method assumes  
 the numerical solution of a partial differential equation (PDE) ded-  
 icated to the Monge-Kantorovich optimisation problem which can  
 be quite time-consuming. In [YYW12], volumetric objects with  
 multiple internal regions were suggested to define the object-space  
 multiphase implicit functions. These functions preserve sharp fea-  
 tures of the object and in some cases provide better results than  
 SDFs.

Distance-based objects can be also defined using the level-set  
 method [GFO18] which provides an implicit representation of a  
 moving front. The main advantage of this method is that it could  
 handle various topological changes of the object thus implement-  
 ing the dynamic implicit surfaces. The evolution of the front is con-  
 trolled by the solution of the level-set equation. The obtained func-  
 tion is transformed into a signed distance function using the solu-

158 tion of some reinitialisation equation. Level-set methods have been  
 159 used in many applications, such as shape optimisation, computa-  
 160 tional fluid dynamics, trajectory planning, image processing and  
 161 others [GFO18].

## 162 2.2. Hybrid representations

163 The main feature of any hybrid representation is that it unifies ad-  
 164 vantages of several representations and compensate for their draw-  
 165 backs. In [PASS01], the concept of hypervolumes was introduced  
 166 as an extension of the general object model [KBDH99] that uni-  
 167 fies the advantages of FRep and hybrid volumes. Hypervolume de-  
 168 scribes a heterogeneous object as  $n$ -dimensional point-set with de-  
 169 fined attributes, operations and relations over them. Another hy-  
 170 brid approach called hybrid surface representation was introduced  
 171 in [LSD04]. It is based on BRep and an implicit surface represen-  
 172 tation and was used for heterogeneous volumetric modelling and  
 173 sculpting.

174 There are approaches when an entire object can be split into dis-  
 175 joint or adjacent components sharing their boundaries. The space  
 176 partitions can be defined by additional boundary surfaces or scalar  
 177 fields. In the most general case, these partitions are represented by  
 178 mixed-dimensional cells combined into a cell complex. The combi-  
 179 nation of a cellular representation and a functionally based con-  
 180 structive representation was proposed in [AKK\*02]. This model  
 181 makes it possible to represent dimensionally non-homogeneous el-  
 182 ements and their cellular representations. The authors showed that  
 183 attributes may reflect not only material, but any volumetric distri-  
 184 bution such as density or temperature.

185 There are some works dedicated to the construction of hybrid  
 186 representations based on SDFs, ADFs and IDFs. In [TP11], the au-  
 187 thors have introduced hybridisation of meshfree, RBF-based, DF-  
 188 based and collocating techniques for solving engineering analy-  
 189 sis problems. The proposed technique enables exact treatment of  
 190 all boundary conditions and can be used with both structured and  
 191 unstructured grids. In patent [Sul15], Sullivan has introduced the  
 192 hybrid ADFs which represented the object by a set of cells. In  
 193 work [ABGA04], the authors introduced a new structure called Hy-  
 194 bridTree that is based on an extended CSG tree which unifies ad-  
 195 vantages of skeletal implicit surfaces and polygonal meshes. The  
 196 hybrid biharmonic distances that are defined similarly to diffusion  
 197 and commute-time (graph) distances were introduced in [LRF10]  
 198 for solving some shape analysis tasks. In [KKL15], a concept of  
 199 the hybrid ADF was introduced for the detailed representation of  
 200 the dynamically changing liquid-solid mixed surfaces.

## 201 2.3. Material and attribute representations

202 A notable early framework called constructive volume geometry  
 203 (CVG) for modelling heterogeneous objects using scalar fields was  
 204 described in [CT00]. The CVG algebraic representation describes  
 205 both interior and exterior of the object that use regular or hierarchi-  
 206 cal data structures. The CVG mathematical framework works with  
 207 spatial objects defined as a tuple  $O = (F_O, A_1, \dots, A_n)$ , where  $F_O$  is  
 208 an opacity field that  $F_O : \mathbb{R}^3 \mapsto [0, 1]$  and  $A_i$  are attribute fields. The  
 209 opacity field defined by the function  $F_O$  is non-distance based and

210 it is not essentially continuous. Discrete fields also can be used in  
 211 this representation using some interpolation procedure.

212 Multi-material heterogeneous volumetric objects [NKI\*18] con-  
 213 sist of three elements: object geometry, object components (e.g.  
 214 domains, partitions or cells sharing their boundaries) and material  
 215 distribution. Material distributions can be defined using material in-  
 216 dexes, piecewise polynomials or continuous scalar fields that pro-  
 217 vide a resolution independent distribution.

218 To define material in the interior of the object, a spatial parti-  
 219 tioning of the object in several spatial regions should be made. Per-  
 220 haps, the most widely used approach is a voxel representation of  
 221 the object [WYJ\*11]. In [HL09], Hiller and Lipson suggested to  
 222 use a voxel data structure as a material building-block for layered  
 223 manufacturing. In another work [DTD\*15], a bitmap voxel-based  
 224 method that uses multi-material high-resolution additive manufactur-  
 225 ing (AM) was introduced. The material properties are combined  
 226 in local material compositions that are further fetched in a AM sys-  
 227 tem. In [BKW\*18] a multi-material voxel-printing method using a  
 228 high-resolution dithering technique was introduced. The material  
 229 in the voxelised object is defined using spatial indexing.

230 Material distribution in interior of the volumetric object can also  
 231 be defined using DF-based approaches. In [BST04], DFs were used  
 232 for parameterisation of the space by distances from the material  
 233 features either exactly or approximately, taking into account that  
 234 the defining attribute function should be at least  $C^1$  continuous to  
 235 avoid creases and stresses in it. In [FSP15], an IDF based method  
 236 for defining gradient materials was introduced. IDFs are repre-  
 237 sented as an approximate Euclidean shortest path and are used for  
 238 interpolation between sources. In [SG17], the authors considered  
 239 the decomposition of the geometry using the existing class of ma-  
 240 terial distance-based functions that set up a material variation in  
 241 heterogeneous objects using the medial axes transform.

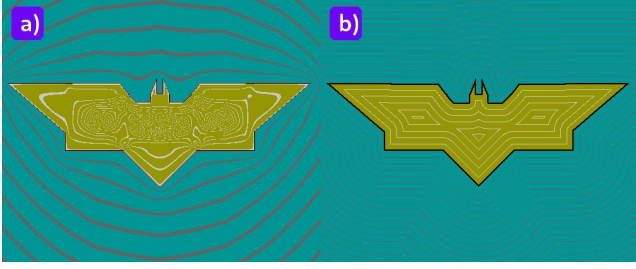
## 242 3. Distance function-based representations

243 In this section we provide some mathematical background and out-  
 244 line in a formalised manner four functionally-based representations  
 245 that will be used to devise the hybrid function representation to be  
 246 introduced in Section 4. We describe in necessary detail the mathe-  
 247 matical basics of those representations and propose the formal defi-  
 248 nitions for two of them, namely ADF and IDF. The advantages and  
 249 drawbacks of the representations are also systematically outlined.

### 250 3.1. Mathematical background and notations

251 Let us introduce the mathematical definitions which will be used  
 252 hereinafter. First, we introduce the definition of a metric space and  
 253 a distance function that follows [HDEE14]:

254 **Definition 3.1** Let  $X$  be a non-empty point set in a Euclidean vector  
 255 space  $\mathbb{R}^n$  and let function  $d : X \times X \mapsto \mathbb{R}$  be such that for points  
 256  $\forall \mathbf{p}_i \in X \subset \mathbb{R}^n$  the following conditions are satisfied:  
 257  $d(\mathbf{p}_1, \mathbf{p}_2) \geq 0$ ;  $d(\mathbf{p}_1, \mathbf{p}_2) = 0 \Leftrightarrow \mathbf{p}_1 = \mathbf{p}_2$ ;  $d(\mathbf{p}_1, \mathbf{p}_2) = d(\mathbf{p}_2, \mathbf{p}_1)$ ;  
 258  $d(\mathbf{p}_1, \mathbf{p}_2) \leq d(\mathbf{p}_1, \mathbf{p}_3) + d(\mathbf{p}_2, \mathbf{p}_3)$ . Then the function  $d(\cdot, \cdot)$  is  
 259 called a *metric* or a *distance function* on set  $X$  and the pair  $(X, d)$  is  
 260 called a *metric space*.



**Figure 1:** a) The FRep field of the functionally defined 'bat' object and b) The SDF field computed for the functionally defined 'bat' object. The colours in the pictures correspond to the point membership rule: blue colour corresponds to the negative values of the defining function, black colour corresponds to the boundary of the object and yellow colour corresponds to the positive values of the defining function.

261 In this work we are focusing on distance-based representations  
 262 for defining volumetric objects. Let us introduce a more instru-  
 263 mental notion for the distance function that satisfies **definition**  
 264 **3.1** and that we will use subsequently in the next sections, as fol-  
 265 lows [Sch97]:

**Definition 3.2** Let  $X$  be a point set in a Euclidean vector space  $\mathbb{R}^n$  and let  $\langle \cdot, \cdot \rangle$  be an inner product defined in  $\mathbb{R}^n$ . Then the Euclidean norm of the point  $\mathbf{p} \in X$  is defined as  $\|\mathbf{p}\| = \sqrt{\langle \mathbf{p}, \mathbf{p} \rangle}$ . If  $\mathbf{q} \in X$  is another point, the distance between these two points is defined as a function:

$$F_{DF}(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\| = \sqrt{\langle \mathbf{p}, \mathbf{q} \rangle} \quad (1)$$

266 In this work we deal with functionally defined objects that are  
 267 specified as closed point subsets  $G \subseteq X$ . As we are dealing with the  
 268 objects defined by the functions, a point membership classification  
 269 is used to distinguish between exterior, boundary and interior of  
 270 the object. Therefore, let us introduce a formal definition of the  
 271 boundary  $\partial G$  of the subset  $G$  as follows:

272 **Definition 3.3** Let  $G$  be a subset of the defined metric space  $(X, d)$ .  
 273 The boundary  $\partial G$  of this subset  $G$  is defined as  $\bar{G} \setminus G_{in}$ , where  $\bar{G} =$   
 274  $\bigcap \{G_C : G_C \supseteq G\}$  is a closure of a metric space  $(X, d)$ ,  $G_C$  is a  
 275 closed set in  $X$ , and interior of  $G$  is  $G_{in} = \bigcup \{G_U : G_U \subseteq G\}$ , where  
 276  $G_U$  is an open set in  $G$ .

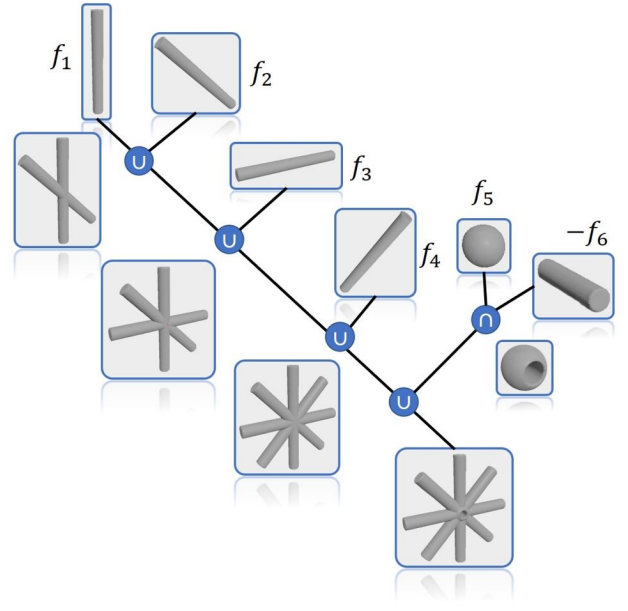
277 There are two important properties of the functions that we rely  
 278 on in the next sections: continuity and smoothness. The continuity  
 279 of the function is defined as follows [DZ10]:

**Definition 3.4** Let  $X$  be an open subset of  $\mathbb{R}^n$ . Let  $C(X)$  be the space of continuous functions  $X \mapsto \mathbb{R}^n$ . Let  $\mathbb{N}^n$  be the set of all tuples  $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ . Then  $|\alpha|$  is the order of  $\alpha$  and  $\partial^\alpha$  is the partial derivative. For an integer  $k \geq 1$

$$C^k(X) := \{f \in C^{k-1}(X) : \partial^\alpha f \in C(X), \forall \alpha, |\alpha| = k\} \quad (2)$$

where

$$|\alpha| = \sum_{i=1}^n \alpha_i, \quad \partial^\alpha = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} \quad (3)$$



**Figure 2:** A constructive tree for the FRep object in the form of a 'snow flake' that was converted to SDF. This tree consists of objects defined by SDF functions  $f_i$  stored in the tree leaves and operations applied to them stored in the tree nodes.

280 In this work we discuss functions that are either at least  $C^0$  or  $C^1$   
 281 continuous. A function  $f$  is said to be of class  $C^0$  if it is continuous  
 282 on  $X \subset \mathbb{R}^n$ . A function  $f$  is said to be of class  $C^1$  if it is differen-  
 283 tiable and continuous on  $X \subset \mathbb{R}^n$ .

Formally, smoothness of the function follows from the previous definition and can be defined as in [Sch97]:

**Definition 3.5** A function  $f : X \mapsto \mathbb{R}^n$  is called smooth if it is  $n$ -times differentiable, i.e. if it belongs to a specific class of functions that can be defined as  $C^n(X)$  for which  $f^{(n)}$  exists and it is continuous, particularly if it satisfies  $C^\infty(X) = \bigcap_{n=1}^\infty C^n(X)$ .

### 3.2. Function representation

291 Let us introduce the definition of FRep [PASS95]:

**Definition 3.6** Let the geometric shape of the object  $O_{FRep}$  be defined as a closed point subset  $G$  of  $n$ -dimensional point set  $X$  in Euclidean space  $\mathbb{R}^n$  with  $\mathbf{p} = (x_1, \dots, x_n) \in \mathbb{R}^n$  using a real-valued defining function  $F_{FRep}(\mathbf{p})$ . Then function representation is defined as

$$O_{FRep} := F_{FRep}(\mathbf{p}) \geq 0 \quad (4)$$

The FRep function (see Fig. 1, (a)) provides the information about point membership:

$$\begin{cases} F_{FRep}(\mathbf{p}) < 0 & \mathbf{p} \in X \setminus G \\ F_{FRep}(\mathbf{p}) = 0 & \mathbf{p} \in \partial G \\ F_{FRep}(\mathbf{p}) > 0 & \mathbf{p} \in G_{in} \end{cases} \quad (5)$$



292 Note, that in this work we will consistently follow this convention  
 293 for all types of distance functions, meaning that they are positive  
 294 inside and negative outside with respect to the zero-level set. The  
 295 major requirement for  $F_{FRep}(\mathbf{p})$  is to be at least  $C^0$  continuous.

296 FRep is a high-level and uniform representation of multidimensional  
 297 geometric objects. The subject of particular interest is 4D  
 298 objects with fourth coordinate specified as time. FRep generalises  
 299 implicit surface modelling and extends a CSG approach. FRep has  
 300 a closure property as operations applied to the FRep defining functions  
 301 produce continuous resulting FRep functions. The FRep object  
 302 can be defined as a primitive (e.g. sphere, torus, cylinder, octa-  
 303 hedron etc.) or as a complex object that is defined in the form of a  
 304 constructive tree. In this case, primitives are stored in the leaves of a  
 305 tree and operations are stored in its nodes.

306 There exist many well-developed operations, e.g. set-theoretic  
 307 operations, metamorphosis, blending and bounded blending, off-  
 308 setting, bijective mapping and others [PASS95]. FRep covers tradi-  
 309 tional solids [Kar99], scalar fields, heterogeneous objects including  
 310 both static and time dependent volumes [I01].

311 Fig. 1 (a) shows the FRep field obtained using 14 set-theoretic  
 312 operations applied to triangles and rectangles to construct the 'bat'.  
 313 In Fig. 2 we present a constructive tree that describes how a FRep  
 314 object in the form of a 'snow flake', that was converted to SDF, was  
 315 created using union  $\cup$  and intersection  $\cap$  set-theoretic operations.  
 316 In general case, the FRep field is not distance-based as interior and  
 317 exterior field isolines do not precisely follow the object shape in  
 318 terms of its zero-level boundary. The advantages and drawbacks of  
 319 the representation can be found in table 1, in the first column.

### 320 3.3. Signed distance function

321 Let us introduce the definition of SDF that relies on **definitions 3.1,**  
 322 **3.2 and 3.3:**

**Definition 3.7** Let  $(X, d)$  be a metric space. Let the geometric shape  
 $G$  of the object  $O_{SDF}$  be specified in  $(X, d)$  as a point subset  $G \subseteq X$ .  
 Then a signed distance function  $F_{SDF}(\mathbf{p})$  is defined as:

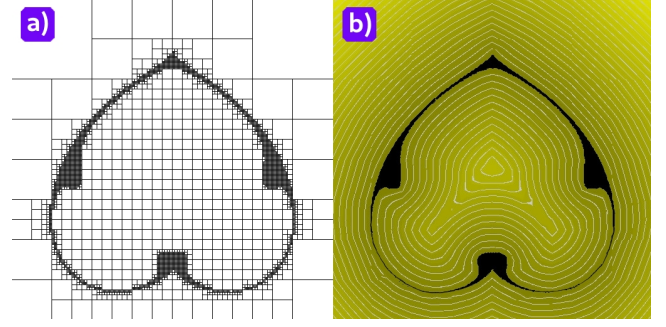
$$323 F_{SDF}(\mathbf{p}) = \begin{cases} d(\mathbf{p}, \partial G) & \text{if } \mathbf{p} \in G \\ -d(\mathbf{p}, \partial G) & \text{otherwise} \end{cases} \quad (6)$$

where  $d(\mathbf{p}, \partial G) \equiv F_{DF}(\mathbf{p}, \partial G)$ . Then the SDF representation is de-  
 fined as follows:

$$324 O_{SDF} := F_{SDF}(\mathbf{p}) \geq 0 \quad (7)$$

325 The SDF function is at least  $C^0$  continuous as it can be not differ-  
 326 entiable at some points of Euclidean space  $\mathbb{R}^n$  and it has gradient  
 327 discontinuities on the object's medial axes. The SDF representa-  
 tion provides the information about point membership in the same  
 manner as FRep.

328 There are a lot of operations defined for SDFs. One of them  
 329 is offsetting [BVG19]. A lot of operations for SDFs were de-  
 330 fined in [PT92]. These are surface interpolation, multiple-object av-  
 331 eraging, spatially-weighted interpolation, texturing, blending, set-  
 332 theoretic operation and metamorphosis. Morphological operations  
 333 such as erosion and dilation [You83] can be also applied to SDFs.



**Figure 3:** An ADF field generated for the FRep object 'heart'. The  
 distance field is restored using bilinear interpolation. a) The  
 constructed quadtree for the 'heart' object. b) The computed unsigned  
 distance field.

334 SDF can be used for a material definition in heterogeneous ob-  
 335 jects [BST04], additive manufacturing [BDN16], collision detec-  
 336 tion problems, particle simulations [KKL15] and others.

337 Fig. 1, (b) shows the SDF field generated for the 'bat' object. As  
 338 it can be seen, the isolines are spaced equidistantly and follow the  
 339 shape of the object. The advantages and drawbacks of SDF can be  
 340 seen in table 1, second column.

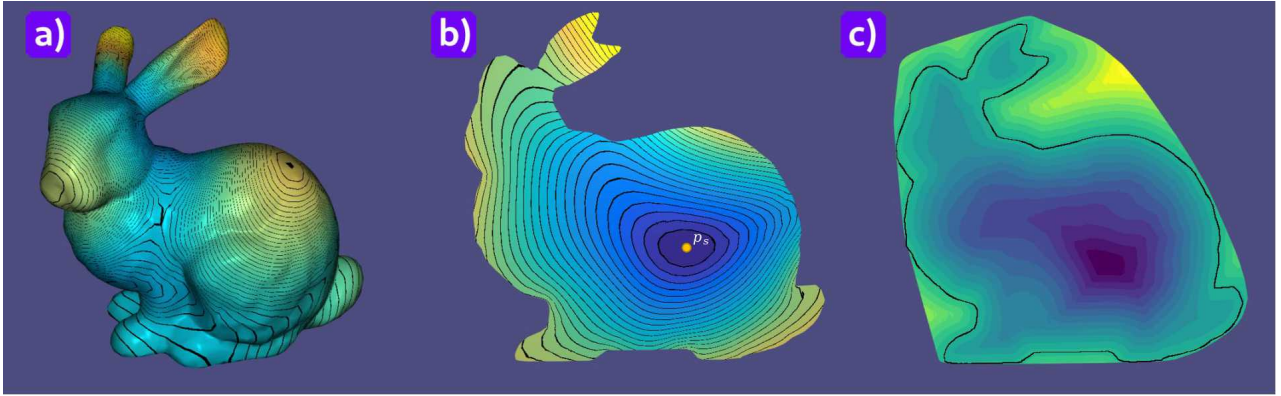
### 341 3.4. Adaptively sampled distance function

342 Adaptively sampled distance function (ADF) [FPRJ00] is a dis-  
 343 tance function that is computed on hierarchical grids, e.g. tree-like  
 344 data structures. ADF satisfies all the requirements of **definitions**  
 345 **3.1, 3.2** and **3.7**. To our knowledge, there is no well-established  
 346 formal definition of ADF in the literature. There are several works  
 347 where ADF is interpreted in a different way compared to [FPRJ00].  
 348 For example, in [WYJ\*11] ADF was defined using T-meshes with  
 349 different interpolation operation for restoring the field, in [KDB16]  
 350 it was suggested to use a hierarchical hp-adaptation for constructing  
 351 ADF, in [TF18] it was suggested to construct ADF using estimation  
 352 of the principal curvatures of the input surface. In this work we in-  
 353 troduce a formal definition of ADF. Let us first give the definition  
 354 of the hierarchical tree structure:

**Definition 3.8** Let a set of nodes and edges  $(Q, E)$  be an undirected  
 355 connected graph  $T$  that contains no loops and starts at some partic-  
 356 ular node of  $T$ . Then such a graph  $T$  is defined as a tree.

357 Let space  $\mathbb{R}^n$  be subdivided according to the local details using  
 358 some k-ary tree  $T := (Q, E)$  with nodes  $q \in Q$ . Each node  $q$  is  
 359 defined as an n-dimensional cell. According to the SDF **definition**  
 360 **3.7** we need to compute the distance to the boundary  $\partial G$  of the  
 361 geometric subset  $G$ . Taking these preliminaries into account, let us  
 362 formulate the ADF definition in the constructive manner:

**Definition 3.9** Let the geometric shape  $G \subseteq X$  of the object  $O_{ADF}$   
 be defined in a metric space  $(X, d)$ . Let  $(X, d)$  be subdivided into  
 nodes  $q \in Q$  with corner vertices  $\mathbf{p}_i$  according to the level of de-  
 tail using k-ary tree  $T := (Q, E)$ . Let the boundary  $\partial G$  be sub-  
 divided with the maximum tree depth, while  $X \setminus G$  and  $G_{in}$  be sub-  
 divided with some minimum tree depth. Let the corner vertices of



**Figure 4:** The IDF field computed for the 'Stanford Bunny' 3D mesh using the method described in [RLF09] and an SDF slice to show the difference in nature of these fields. a) the IDF field computed on the boundary of the mesh. Black isolines show how the field is changing according to the shape of the object; b) the interior slice of the mesh with computed IDFs. The yellow point  $\mathbf{p}_s$

in the slice corresponds to the 'source' point. c) the SDF slice of the same model with computed interior and exterior distances. Colour changing reflects how the distances are changing from interior to exterior of the object.

the boundary nodes  $q$  be defined as  $\mathbf{p}_{b_i}$ . Then the distance function between these points is  $d(\mathbf{p}_i, \mathbf{p}_{b_i}) \equiv F_{DF}(\mathbf{p}_i, \mathbf{p}_{b_i})$ . Thereafter, the ADF distance function  $F_{ADF}(\mathbf{p})$  on the tree  $T$  is restored at each node  $q$  using some interpolation function  $F_I(\mathbf{p})$  and is defined as follows:

$$F_{ADF}(\mathbf{p}) = \begin{cases} (F_I \circ F_{DF})(\mathbf{p}) & \text{if } \mathbf{p} \in G \\ -(F_I \circ F_{DF})(\mathbf{p}) & \text{otherwise} \end{cases} \quad (8)$$

The ADF representation is defined in the form of an inequation:

$$O_{ADF} := F_{ADF}(\mathbf{p}) \geq 0 \quad (9)$$

The ADF field generated as it was described in [FPRJ00] has  $C^0$  discontinuities where the cells of different size appear and it has  $C^1$  discontinuities caused by the bilinear/trilinear interpolation that was used for restoring a DF at each cell. The generated ADF field for the FRep 'heart' object can be seen in Fig. 3. The distance field was restored using bilinear interpolation, therefore, the field is non-smooth and has  $C^0$  discontinuities (some isolines are disconnected).

The ADF representation provides the information about point membership in the same manner as FRep. The subset  $X$  can be subdivided using one of the types of k-ary trees: quadtrees or octrees. ADF can be used for an efficient interactive real-time modelling, e.g. sculpting, of the heterogeneous objects as the tree data structure provides fast access to object's geometry and its specified attributes. ADF is also suitable for solving surface restoration problems [DCL\*08,TF18]. It supports the same operations as SDF. ADF are especially suitable for dynamic simulations [KDB16], for example, morphing between shapes, as a hierarchical data structure can efficiently be rebuilt at each animation frame [FPRJ00]. The advantages and drawbacks of ADF can be seen in table 1, third column.

### 3.5. Interior distance function

Interior distance function (IDF) is not a well-established notion yet as in literature there is neither a general approach for generating DFs of this rather broad nature nor one unique name for them. In this work we suggest to use this notion for a representation with a defining function obtained as follows: the distance function is computed on the boundary of the object as a shortest path between boundary points and then the generated distances are smoothly interpolated in the object's interior. Let us introduce the definition of IDF that relies on the definitions specified in subsection 3.1:

**Definition 3.10** Let the geometric shape  $G \subseteq X$  of the object  $O_{IDF}$  be defined in a metric space  $(X, d)$ . Let points  $\mathbf{p}_{b_i}$  belong to  $\partial G$ , and let points  $\mathbf{p}_{in_k}$  belong to  $G_{in}$ . Let a distance function  $d(\mathbf{p}_{b_i}, \mathbf{p}_{b_j}) \equiv F_{DF}(\mathbf{p}_{b_i}, \mathbf{p}_{b_j}) = \|\mathbf{p}_{b_i} - \mathbf{p}_{b_j}\|_{\mathbb{R}^n}$  between any boundary points  $\mathbf{p}_{b_i}$  and  $\mathbf{p}_{b_j}$  on a curved domain  $\partial G$  be recovered. Thereafter, by constructing an interpolation function  $F_I(F_{DF}(\mathbf{p}_{b_i}, \mathbf{p}_{b_j}), \mathbf{p}_{in_k})$  that is at least  $C^1$  continuous, boundary distances are extended to interior of the object  $O_{IDF}$ . Therefore, the IDF function can be defined as:

$$F_{IDF}(\mathbf{p}_{in_k}) = F_I(F_{DF}(\mathbf{p}_{b_i}, \mathbf{p}_{b_j}), \mathbf{p}_{in_k}) \quad (10)$$

where  $0 \leq i, j, < N$ ,  $N$  is the number of boundary points,  $0 \leq k < M$ ,  $M$  is the number of interior points. The IDF representation is defined in the form of an inequation:

$$O_{IDF} := F_{IDF}(\mathbf{p}) \geq 0 \quad (11)$$

Note, that the boundary of the object is treated as the source for generating interior distances.

Technically, IDF is usually obtained by solving a partial differential equation (PDE) or applying some numerical method, e.g., graph approaches [LRL11] or Markov chains [CL06]. Among PDE-based methods the following methods can be considered as representative: geodesic distances obtained as the solution of heat equation [CWW13], diffusion maps combined with smooth barycentric interpolation of the distances in interior of the object [RLF09], the

	FRep	SDF	ADF	IDF
advantages	<ul style="list-style-type: none"> <li>• FRep generalises implicit surface modelling and extends a constructing modelling approach;</li> <li>• FRep supports point membership;</li> <li>• FRep is closed guaranteeing to get an at least <math>C^0</math> continuous resulting function;</li> <li>• FRep covers solids, scalar fields, volumes, time-dependent volumes and hypervolumes for heterogeneous object modelling.</li> <li>• FRep has many well-developed operations that support multidimensional transformations in <math>\mathbb{R}^n</math>;</li> </ul>	<ul style="list-style-type: none"> <li>• SDF provides distances to the object surface both inside and outside it;</li> <li>• SDF is a Lipschitz continuous function;</li> <li>• SDF is Frechet differentiable almost everywhere;</li> <li>• SDF satisfies the solution of the eikonal equation;</li> <li>• SDF supports point membership;</li> <li>• SDF is effectively discretised, has a predictable field behaviour and is efficiently rendered.</li> </ul>	<ul style="list-style-type: none"> <li>• ADF data structure efficiently subdivides the Euclidean space <math>\mathbb{R}^n</math> according to the level of detail;</li> <li>• ADF distances are adaptively sampled;</li> <li>• ADF supports point membership;</li> <li>• ADF possesses an efficient memory management: in a small amount of memory a significant amount of information about the object can be stored;</li> <li>• ADF hierarchical tree data structure is fast to rebuild that makes it possible to handle time-variant objects;</li> <li>• ADF can be efficiently rendered in real time.</li> </ul>	<ul style="list-style-type: none"> <li>• IDF is shape-aware;</li> <li>• IDF is deformed with the boundary;</li> <li>• IDF is smooth;</li> <li>• IDF is suitable for the distance-based attribute definition in interior of the object.</li> </ul>
drawbacks	<ul style="list-style-type: none"> <li>• Distances can be obtained for a limited number of FRep objects;</li> <li>• FRep object can have a boundary with dangling portions that are not adjacent to the interior of the object;</li> <li>• FRep has an unpredictable non-distance based behaviour of the resulting field and, as a consequence, it is sometimes problematic to render in 3D.</li> </ul>	<ul style="list-style-type: none"> <li>• SDF is not differentiable at some points of Euclidean <math>\mathbb{R}^n</math> space. Loss of SDF differentiability happens when the current point is sufficiently close to a concave singularity (a concave corner/edge);</li> <li>• SDF has discontinuous gradients on the object's medial axes;</li> <li>• SDF is not smooth;</li> <li>• SDF is not suitable for attribute modelling due to <math>C^1</math> discontinuity.</li> </ul>	<ul style="list-style-type: none"> <li>• ADF field has <math>C^0</math> discontinuities where cells of the different size appear as the result of the hierarchical subdivision;</li> <li>• ADF field has <math>C^1</math> discontinuities that are introduced by the bilinear/trilinear interpolation [FPRJ00] during reconstruction of the field at each cell;</li> <li>• ADF is not suitable for attribute modelling due to <math>C^0</math> and <math>C^1</math> discontinuities.</li> </ul>	<ul style="list-style-type: none"> <li>• IDF can be computationally expensive;</li> <li>• IDF field accuracy for some methods is highly dependent on a time step and type of the used discretisation;</li> <li>• IDF is defined only in interior of the object.</li> </ul>

**Table 1:** Comparison table of the advantages and drawbacks of FRep, SDFs, ADFs and IDFs.

404 optimal mass transport [SRGB14] and some others. IDF is usually  
 405 used in the tasks related to shape analysis [RLF09], geometry  
 406 restoration [PS12], morphing and less commonly for an attribute  
 407 definition in interior of the object [FSP15].

408 In Fig. 4 we show how the approach described in [RLF09] can  
 409 be applied to the polygonal mesh of the 'Stanford Bunny'. The distances  
 410 are computed on the boundary, as it can be seen in the part  
 411 (a), using the diffusion maps, and then propagated in interior of the  
 412 object, as it can be seen in the part (b), using the barycentric inter-  
 413 polation. A simple procedure of IDF isolines generation is the  
 414 computing of distances from the fixed 'source' point  $\mathbf{p}_s$  to other  
 415 points of the mesh. If we compare two pictures shown in Fig. 4  
 416 (b) and (c), we can see that the distance fields obtained in interior  
 417 of the bunny are completely different. The IDF field (b) is smooth  
 418 and continuous while the SDF field (c) is not smooth and has some  
 419 sharp features in interior of the object.

420 The advantages and drawbacks of IDF can be found in table 1,  
 421 last column.

### 422 3.6. Heterogeneous objects

423 In the previous subsections we have discussed how geometric shape  
 424 of objects can be defined using distance-based methods. In this sub-  
 425 section we discuss how attributes can be considered in concert with  
 426 the geometric shape of the object to represent the heterogeneous  
 427 object. Let us first introduce a general definition of the heteroge-  
 428 nous object.

**Definition 3.11** Let the object  $O_H$  be defined as a two component  
 429 tuple: geometric shape  $G \subseteq X$  in the form of a multidimensional  
 430 point-set geometry and attributes  $A_i$  corresponding to the physical  
 431 properties of the object  $O_H$ . Then such object  $O_H$  is a heteroge-  
 432 nous object defined as:

$$433 O_H := (G, A_1, \dots, A_n), \quad (12)$$

434 where  $n \in \mathbb{N}$  is the number of attributes.

435 Attribute distributions specified in heterogeneous objects  $O_H$  can  
 436 be uniform or non-uniform. For instance, the simple example of  
 437 the uniform distribution can be a homogeneously coloured object.  
 438 As to non-uniformity, it can be presented as porous structures or  
 439 microstructures with non-linear varying density.

In this work we will apply the hypervolume model [PASS01] to  
 440 define heterogeneous objects  $O_{H_V}$  using FRep or any other distance  
 441 function-based representation. A hypervolume object is defined as  
 442 follows:

**Definition 3.12** Let the geometric shape  $G$  of  $O_{H_V}$  be defined by  
 443 a real-valued function  $F_G(\mathbf{p})$ ,  $\mathbf{p} \in \mathbb{R}^n$  that is at least  $C^0$  continuous  
 444 and let attributes be defined by any  $F_{A_i}(\mathbf{p})$ . Then heterogeneous  
 445 object  $O_{H_V}$  is defined as:

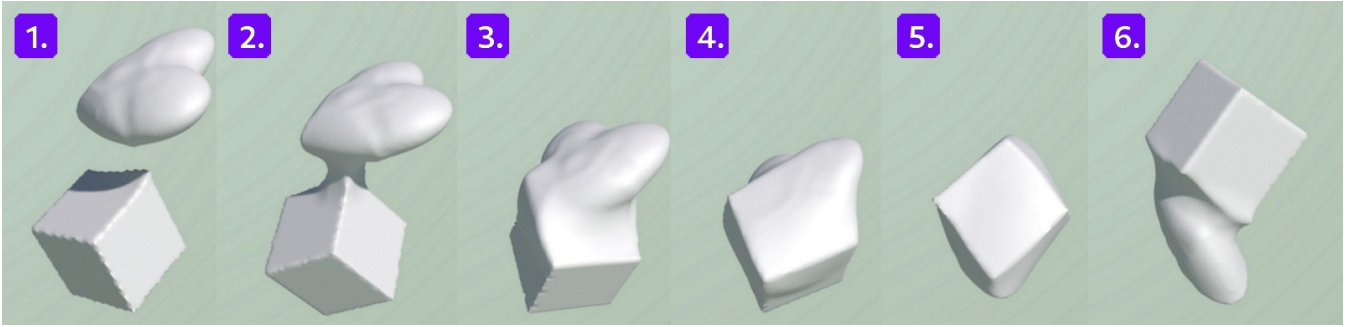
$$446 O_{H_V} := (G, A_1, \dots, A_n) : (F_G(\mathbf{p}), F_{A_1}(\mathbf{p}), \dots, F_{A_n}(\mathbf{p})), \quad (13)$$

447 where  $n \in \mathbb{N}$  is the number of attributes.

448 In general case, attribute functions  $F_{A_i}(\mathbf{p})$  are not necessarily  
 449 continuous. However, as it was shown in [BST04], better control  
 450 of the attributes on the surface and in the interior of the distance-  
 451 based objects can be achieved when the attribute defining functions  
 452 are parameterised by the distances. The main requirement for the  
 453 distance function is to be at least  $C^1$  continuous. This requirement  
 454 prevents the appearing of stress concentrations, creases and other  
 455 singularities in modelled attribute distributions.

456 There are several interesting examples discussed in [BST04]. In  
 457 particular, the distance-based smooth and differentiable attribute  
 functions were applied to represent a parabolic distribution of the  
 graded refractive index in Y-shaped solid of the waveguide. In this  
 case, it is important that the distribution of the index of refraction  
 is uniform and smooth. Another example is to use such distance-  
 based attribute functions for modelling different types of materials,  
 e.g. silicon carbide (SiC). It is important to note that the approach  
 introduced in [BST04] was not applied to such attributes as tex-  
 tures, colours and similar attributes.





**Figure 5:** The STB-based metamorphosis operation over the initially FRep 'heart' converted to HFRep and initially BRep 'cube' converted to SDF 'cube'. Supplementary video: figure5.m4v

#### 4. Hybrid function representation

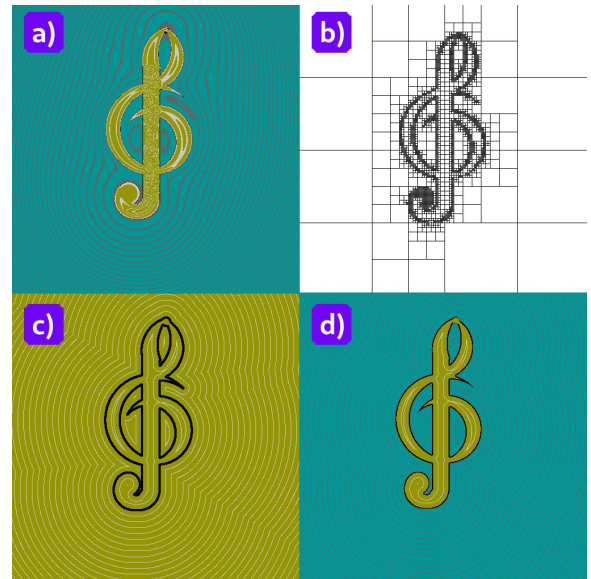
In this section we introduce and systematically describe a general approach for defining heterogeneous volumetric objects using a hybrid function representation (HFRep). First, we list the requirements for HFRep, then outline its mathematical basics, and finally describe its properties with respect to four basic DF-based representations.

##### 4.1. Problem statement

Let us give the exact problem statement. Our goal is to propose a hybrid function representation (HFRep) that is suitable for defining volumetric heterogeneous objects. We assume that the geometric shape  $G$  of the given object is defined by FRep, and its defining function is known. To devise the HFRep embracing advantages and circumventing disadvantages of FRep, SDF, ADF, IDF, it is essential to obtain a real-valued defining function in an  $n$ -dimensional Euclidean space with the following properties:

1. The HFRep function should provide sufficiently accurate distance approximation in Euclidean space  $\mathbb{R}^n$  without  $C^0$  and  $C^1$  discontinuities.
2. The HFRep function should be at least  $C^0$  continuous with possibility to enforce it to be at least  $C^1$  continuous.
3. The HFRep function should satisfy the point membership test: it should be positive in interior of the geometric shape  $G$ , take exact zero values only at the object boundary  $\partial G$  and it should be negative in exterior of the geometric shape  $X \setminus G$ ;
4. The HFRep should be a multidimensional object representation; in particular, dealing with 4D objects is of paramount importance to cover time-variant models with the fourth 'time' coordinate;
5. The HFRep representation should be suitable for the heterogeneous object modelling allowing for defining attribute functions related to the geometry;
6. The HFRep attribute functions should depend on evaluation point  $\mathbf{p} \in G$  and be parameterised by distance values of the obtained HFRep geometry function.

The fulfilment of these conditions guarantees that the generated object will be watertight and such operations as blending and metamorphosis will not suffer from creases. Overall, the defining



**Figure 6:** The illustration of HFRep based on FRep and ADF with applied PHT-spline (a polynomial spline over hierarchical T-mesh) interpolation to restore the distance field at each cell. ADFs are generated using a numerical solution of the eikonal equation on the quadtree. a) the FRep field; b) a hierarchical quadtree subdivision (the maximum tree depth equals to 10 with 4201 leaves); c) UDF computed on the quadtree with the applied PHT-spline interpolation for restoring distances at each quadtree cell; d) the HFRep field that was obtained using the generated ADF.

HFRep function that is considered in concert with attribute functions parameterised by distances will be suitable for dealing with multi-material aspects of heterogeneous objects including time-variant ones.

##### 4.2. Definition of the hybrid function representation

First, we provide a mathematical definition for the geometric aspects of HFRep. Then we add the part related to attributes. The



Inherited from FRep	Inherited from SDF	Inherited from ADF	Inherited from IDF
<ul style="list-style-type: none"> <li>The continuity of the HFRep function depends on the continuity of the FRep function.</li> <li>The HFRep object is watertight.</li> <li>HFRep represents multidimensional objects, in particular 4D objects with the fourth coordinate specified as time.</li> </ul>	<ul style="list-style-type: none"> <li>HFRep provides at least <math>C^0</math> continuous distance function;</li> <li>the HFRep object is watertight;</li> <li>the HFRep function is Lipschitz continuous and Fréchet differentiable everywhere;</li> <li>the HFRep function satisfies the solution of the eikonal equation;</li> <li>the HFRep object can be efficiently discretised and rendered.</li> </ul>	<ul style="list-style-type: none"> <li>HFRep provides at least <math>C^0</math> continuous distance function for any FRep object that was spatially subdivided according to the local details using a hierarchical data structure;</li> <li>Hierarchical data structure can also be used for defining and storing object's attributes.</li> </ul>	<ul style="list-style-type: none"> <li>HFRep provides at least a <math>C^0</math> continuous unsigned distance function for any FRep object in its interior if IDF is used for obtaining distances;</li> <li>Distances in the interior of the HFRep object are shape-aware and deformed with boundaries;</li> <li>There is also a potential for modelling attributes in interior of the volumetric object.</li> </ul>

**Table 2:** Properties of the hybrid function representation that depend on the combination of FRep with one of the distance fields.

503 geometric shape  $G$  of an HFRep object  $O_{HFRep}$  is defined as follows:  
504

**Definition 4.1** Let the geometric shape  $G \subseteq X$  of the object  $O_{HFRep}$  be defined in a metric space  $(X, d)$ . Given at least  $C^0$  or  $C^1$  continuous FRep function  $F_{FRep}(\mathbf{p})$ , the distance to the object boundary  $\partial G$  is defined as  $(F_I \circ F_{DF})(\mathbf{p}, \partial G) \equiv (F_I \circ F_{DF})(\mathbf{p})$ , where  $F_I(\cdot)$  is at least  $C^1$  continuous interpolation function and  $d(\cdot, \cdot) \equiv F_{DF}(\cdot, \cdot)$  is a distance-based function, in particular SDF, ADF or IDF. Then the HFRep function is defined as follows:

$$F_{HFRep}(\mathbf{p}) = (F_{sign} \circ F_{FRep})(\mathbf{p}) \cdot (F_I \circ F_{DF})(\mathbf{p}) \quad (14)$$

where  $F_{sign}(\cdot)$  is an at least  $C^1$  continuous function that provides a sign for the computed function  $(F_I \circ d)(\mathbf{p})$  and satisfies the FRep point membership test, equation (5). Finally, the HFRep representation is defined as:

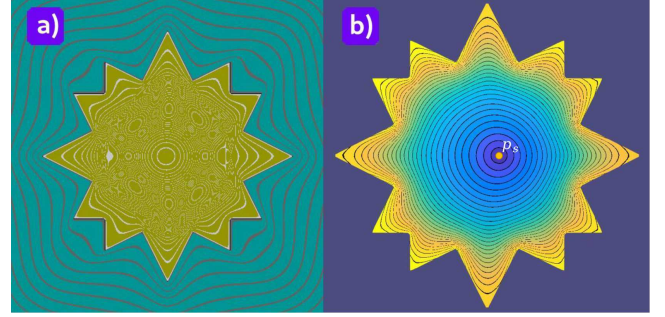
$$O_{HFRep} := F_{HFRep}(\mathbf{p}) \geq 0 \quad (15)$$

505 The continuity of the HFRep function  $F_{HFRep}(\mathbf{p})$  depends on the  
506 continuity of the FRep function  $F_{FRep}(\mathbf{p})$ . In the case when we are  
507 dealing only with geometric shapes, it is sufficient to have  $C^0$  continuity  
508 for the HFRep function. Otherwise, in case of heterogeneous  
509 object modelling, the HFRep function should belong to the class of  
510 functions that are at least  $C^1$  continuous. We will give details on  
511 how to control the continuity of the HFRep function later in this  
512 section.

513 Now let us show that  $F_{HFRep}(\mathbf{p})$  continuity is either  $C^0$  or  $C^1$ . By  
514 applying a smoothing interpolation function  $F_I(\cdot)$  that is at least  $C^1$   
515 continuous to the discrete unsigned distance field (UDF) obtained  
516 using  $F_{DF}(\mathbf{p}, \partial G) \in C^0$ , we enforce the property  $(F_I \circ F_{DF})(\mathbf{p}) \in$   
517  $C^1$ . The composition of functions  $(F_{sign} \circ F_{FRep})(\mathbf{p})$  is at least  $C^0$   
518 or  $C^1$  continuous, depending on the continuity of  $F_{FRep}(\mathbf{p})$ . The  
519 theorem about the continuity of the composition of two continuous  
520 functions was proven in [Bom67]. Therefore the continuity of the  
521 HFRep function is defined as:  $C_{HFRep} = \min(C_{F_{sign} \circ F_{FRep}}^m, C_{F_I \circ F_{DF}}^k)$ ,  
522 where  $m = 0$  or  $m = 1$ ,  $k = 1$ , i.e. the minimum class of continuity  
523 between two function compositions.

524 Now on the basis of **definition 3.12**, we can formulate the defini-  
525 tion of the heterogeneous HFRep object  $O_{Hv, HFRep}$  as follows:

**Definition 4.2** Let the geometric shape  $G$  of  $O_{Hv, HFRep}$  be defined  
541 by at least  $C^1$  continuous  $F_G(\mathbf{p}) = F_{HFRep}(\mathbf{p})$  distance-  
542 based function. Let the attribute  $A_i$  be defined as a real-valued  
543 function  $F_{A_i}(F_{HFRep}(\mathbf{p}), \mathbf{p})$ . Then the HFRep heterogeneous object  
544



**Figure 7:** (a) 'Star' object and its FRep field; (b) the HFRep 'star' object generated on the basis of the FRep object. The boundary of the FRep object (a) was extracted and then used for computing boundary distances. The obtained distances were interpolated in interior of the HFRep 'star' object using barycentric interpolation and mean-value coordinates. The isolines and colour show how the field changes from the source point  $\mathbf{p}_s$  towards the object boundary.

$O_{Hv, HFRep}$  is defined as:

$$O_{Hv, HFRep} := \begin{cases} F_G(\mathbf{p}) := F_{HFRep} \geq 0 \\ F_{A_i}(F_{HFRep}(\mathbf{p}), \mathbf{p}), \quad i = [0, \dots, n] \in \mathbb{N} \end{cases} \quad (16)$$

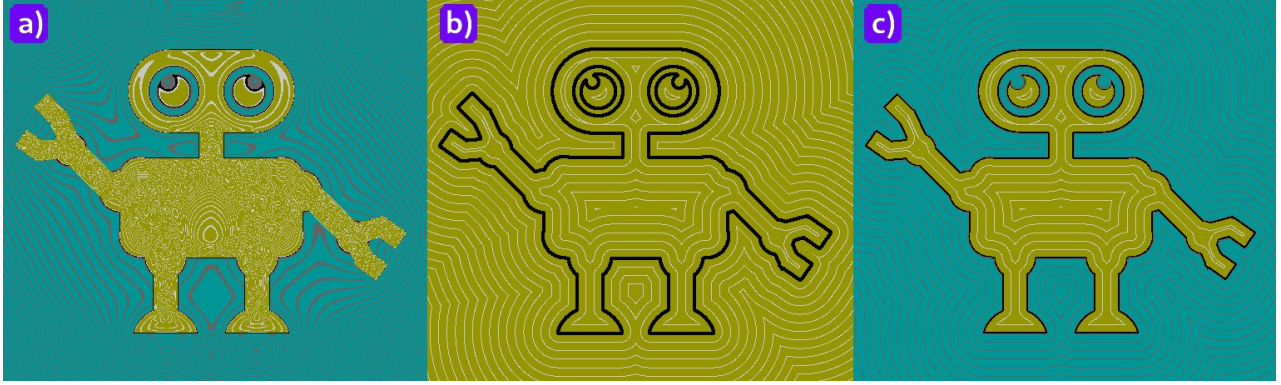
526 where  $n$  is a number of attributes.

527 The properties of the introduced hybrid function representation  
528 are outlined in table 2. It shows be-pair combinations of FRep with  
529 one of the distance fields, namely SDF, ADF or IDF.

530 Fig. 5 shows a metamorphosis between two oscillating 4D geo-  
531 metric shapes ('heart', initially the FRep object then converted to  
532 HFRep; 'cube', initially the BRep object then converted to SDF) using  
533 the space-time blending (STB) method [TAPA20]. The result is  
534 a non-distance functionally defined watertight object that is contin-  
535 uous and smooth.

536 In Fig. 6 (d), we demonstrate the restored distance field com-  
537 puted on the hierarchical grid obtained for the initial FRep object  
538 defined as a 'treble clef', Fig. 6 (a). There is neither  $C^0$  nor  $C^1$   
539 discontinuities in the field as it can be seen in Fig. 6, (c) or (d). All  
540 the isolines are smooth and continuous.

541 In Fig. 7, (b) we show a simple example of interior distances  
542 computed for the FRep 'star' object, Fig. 7 (a), that was con-  
543 structed using seven set-theoretic operations. First, the boundary of  
544 the FRep object was extracted for computing boundary distances.



**Figure 8:** The illustration of the basic algorithm: a) step 1: the computed field of the 'robot' FRep object; b) steps 2 - 3: the computed unsigned distance field that can be obtained using, e.g., the distance transform or a numerical solution of the eikonal equation. The obtained field is smoothed using some spline interpolation; c) step 4: the generated HFRep field.

545 Then, the interior of the obtained convex contour was triangulated.  
 546 Finally, the boundary distances were propagated in interior of the  
 547 shape as it is described in [RLF09]. The black isolines show that  
 548 the obtained field is at least  $C^1$  continuous as they are smoothly  
 549 changing in the object interior.

## 550 5. Basic algorithm for generating HFRep

551 Let us outline in a step-by-step manner the algorithmic solution  
 552 on generating the HFRep functions. The basic algorithm covers  
 553 all paired combinations of FRep with DF representations, namely  
 554 SDF, ADF and IDF, and allows to generate both a geometric shape  
 555 and attributes. Some steps of the basic algorithm will be slightly  
 556 different depending on the particular type of the DF paired with  
 557 FRep. All examples in this and the following sections were com-  
 558 puted on a laptop with a 2.6 GHz *Intel Skylake 6700* processor and  
 559 16 Gb of RAM.

560 Let us start from the algorithm for generating a geometric shape  
 561 of the object  $O_{HV,HFRep}$ . Fig. 8 demonstrates the generated function  
 562 field for each step of the basic algorithm.

### 563 5.1. Algorithm for HFRep geometry generation

564 1. According to the **definition 4.1**, we start the construction  
 565 of an HFRep object  $O_{HFRep}$  from defining the FRep func-  
 566 tion  $F_{FRep}(\mathbf{p})$  for its geometric shape  $G$ . The FRep function  
 567  $F_{FRep}(\mathbf{p})$  can be defined analytically, with function evaluating  
 568 algorithm or using a point cloud for which it is possible to ob-  
 569 tain a real-valued at least  $C^0$  continuous  $F_{FRep}(\mathbf{p})$ . It could also  
 570 be a a complex FRep object that is obtained in the form of a  
 571 constructive tree.

At this step we can also enforce HFRep function  $F_{HFRep}(\mathbf{p})$   
 to be at least  $C^1$  continuous as its continuity depends on the con-  
 tinuity of  $F_{FRep}(\mathbf{p})$ . We have to examine the obtained  $F_{FRep}(\mathbf{p})$   
 for continuity and differentiability. The most practically used  
 FRep set-theoretic operations in the form of the following R-

function system are [PASS95]:

$$\begin{aligned} f_{\cup}(f_1(\mathbf{p}), f_2(\mathbf{p})) &= f_1 + f_2 + \sqrt{f_1^2 + f_2^2} \\ f_{\cap}(f_1(\mathbf{p}), f_2(\mathbf{p})) &= f_1 + f_2 - \sqrt{f_1^2 + f_2^2} \end{aligned} \quad (17)$$

These functions have  $C^1$  discontinuity in points where both argu-  
 ments are equal to zero. Accordingly, the resulting function  
 will only be  $C^0$  continuous. If we need to obtain an at least  $C^1$   
 continuous resulting function, we can apply another R-function  
 system that is at least  $C^{n-1}$  continuous [Rva82]:

$$f_{\cup}(f_1(\mathbf{p}), f_2(\mathbf{p})) = \begin{cases} f_1 f_2 (f_1^n + f_2^n)^{-1/n}, & \forall f_1 > 0, f_2 > 0; \\ f_1, & \forall f_1 \leq 0, f_2 \geq 0; \\ f_2, & \forall f_1 \geq 0, f_2 \leq 0; \\ (-1)^{n+1} (f_1^n + f_2^n)^{1/n}, & \forall f_1 < 0, f_2 < 0; \end{cases} \quad (18)$$

$$f_{\cap}(f_1(\mathbf{p}), f_2(\mathbf{p})) = \begin{cases} (f_1^n + f_2^n)^{1/n}, & \forall f_1 > 0, f_2 > 0; \\ f_2, & \forall f_1 \leq 0, f_2 \geq 0; \\ f_1, & \forall f_1 \geq 0, f_2 \leq 0; \\ (-1)^{n+1} f_1 f_2 (f_1^n + f_2^n)^{-1/n}, & \forall f_1 < 0, f_2 < 0; \end{cases}$$

where  $f_1(\mathbf{p})$  and  $f_2(\mathbf{p})$  are FRep functions.

Fig. 8, (a) shows the FRep field obtained for the 'robot' object,  
 that was generated using 39 set-theoretic operations, equation  
 (17), applied to circles and rectangles. In our implementation it  
 took 0.32 seconds to generate the FRep field.

2. The values of the function  $F_{FRep}(\mathbf{p})$  are used as an input for  
 computing distance functions  $F_{DF}(\mathbf{p}, \partial G)$  that should satisfy  
 one of the **definitions 3.7, 3.9 or 3.10**. At this step we obtain  
 an unsigned distance function that is defined as:

$$F_{DF}(\mathbf{p}) = d(\mathbf{p}, \partial G), \quad \forall \mathbf{p} \in X \quad (19)$$

Fig. 8, (b) shows the unsigned distance field that was obtained  
 on the basis of a typical SDF generation algorithm [LL92]. It  
 took 0.018 seconds to generate the UDF field.

581 If the distances are computed using ADF, first, we need to subdi-  
 582 vide the space using a hierarchical data structure, e.g. quadtree,  
 583 Fig. 6, (b) and during its construction we also need to compute  
 584 basis functions, basis vertices and extraction operators for the  
 585 hierarchical splines. Then we need to compute the distances at  
 586 the corner vertices of each cell. Finally, we restore distances in  
 587 interior of each cell using at least  $C^1$  continuous spline-based  
 588 interpolation to obtain a smooth and continuous distance field,  
 589 e.g. shown in Fig. 6, (c). It took 9 seconds to generate the UDF  
 590 field using our unoptimised implementation.

591 Specifically for IDFs, the function  $F_{DF}(\mathbf{p})$  is defined according  
 592 to equation (10). Distances are computed on the boundary of the  
 593 object  $O_{FRep}$  and then interpolated in its interior. In Fig. 7 (a),  
 594 we can see the field of the FRep-defined 'star' object that was  
 595 used for generating HFRep IDF-based field that is shown in Fig.  
 596 7, (b). It took 16 seconds to generate the IDF field of this 'star'  
 597 object.

598 In Fig. 9, (a) we show a possible extrapolation scheme that can  
 599 be used to obtain distances in exterior of the object and make  
 600 an IDF-based field signed at the last step of this algorithm.  
 601 To do this, we need to use the boundary distances (Fig. 9, (a),  
 602 dark blue circles) and an appropriate at least  $C^1$  continuous  
 603 extrapolation operation, that will be used for obtaining distances  
 604 outside the object (9, Fig.7(a), red circles).  
 605

3. The distance field obtained at the previous step is unsigned  
 624 and discrete as it was computed on the finite point subset  $X \subset \mathbb{R}^n$ .  
 625 To enforce the continuity and smoothness of the computed field,  
 626 we need to apply some at least  $C^1$  continuous interpolation function  
 627 ( $F_I \circ F_{DF}$ )( $\mathbf{p}$ ) to the generated unsigned field, e.g. spline-based:  
 628

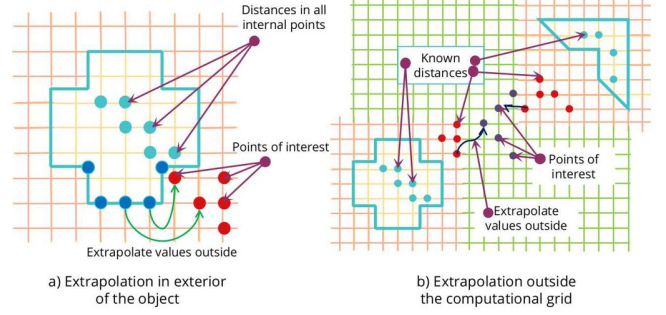
$$F_{smDF}(\mathbf{p}) = (F_I \circ F_{DF})(\mathbf{p}) \quad (20)$$

606 We also need to apply a smoothing operation to an IDF field  
 607 if at the previous step an extrapolation operation was applied.  
 608 Otherwise, IDFs are smooth as smoothness is their inherent  
 609 property. An important requirement for the interpolation  
 610 function  $F_I(\cdot)$  is to avoid introducing extra zeros in the distance  
 611 field generated using function  $F_{DF}(\mathbf{p}, \partial G)$ . It took 0.013  
 612 seconds to compute a smoothed UDF field for the 'robot' object.  
 613

4. Finally, as the distance field obtained after previous steps is  
 638 unsigned, we need to restore the field sign to distinguish between  
 639 exterior  $X \setminus G$ , boundary  $\partial G$  and interior  $G_m$  of the object  
 640  $O_{HV, HFRep}$ . We suggest to use some at least  $C^1$  continuous step-  
 641 function  $F_{st}(F_{FRep}(\mathbf{p}))$  with the scope  $[-1, 1]$ , that depends on  
 642 the values of the defining FRep function  $F_{FRep}(\mathbf{p})$  and approxi-  
 643 mates its well-defined behaviour ( $-1$  in exterior of the object,  $0$   
 644 on the boundary of the object and  $+1$  in interior of the object).  
 645 Therefore, the resulting HFRep function  $F_{HFRep}(\mathbf{p})$  is defined  
 646 according to **definition 4.1** as follows:  
 647

$$F_{HFRep}(\mathbf{p}) = (F_{st} \circ F_{FRep})(\mathbf{p}) \cdot F_{smDF}(\mathbf{p}) \quad (21)$$

648 The HFRep field generated by this function can be seen in Fig.  
 649 8, (c). It took 0.037 seconds to generate the HFRep field of  
 650 the 'robot' object. After a geometric shape of the HFRep object  
 651  $O_{HFRep}$  was generated, we can apply different operations  
 652 to it provided that they are realised by functions which are at  
 653 least  $C^0$  continuous. The HFRep object is also compatible with



**Figure 9:** Two cases when extrapolation is important to enforce the continuity of the field: a) when we have DF (light pink and light orange colours) for two objects that are distantly placed in space. In this case we need to extrapolate the distance values into the points of the green grid; b) when we computed the IDF and it is essential to obtain distances in exterior of the object.

other distance-based objects. However, to preserve the distance properties for the object obtained after applying multiple operations, we might need to apply the steps of this algorithm again to this object.

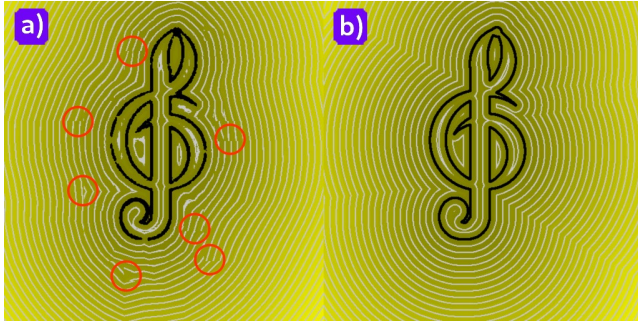
There is a limited number of operations that preserve the distance property for the geometric shape  $G$  of the object obtained after their application. These operations are rigid (Euclidean) transformations: rotations, translations, reflections or their combination. Another distance preserving operations [PT92] are affine translations, offsetting, linear surface interpolation, surface blurring and compression, set-theoretic operations in the form of  $\min(f_1(\mathbf{p}), f_2(\mathbf{p}))$  or  $\max(f_1(\mathbf{p}), f_2(\mathbf{p}))$  [Rva82]. In cases of other operations [RMD11] (e.g., scaling, blending, space-time blending, twisting, tapering and sweeping, set-theoretic operations in the form of R-functions [PASS95]) after their application, we have to apply the basic algorithm to the obtained object to restore the distance property.

To make the HFRep representation continuous on the whole domain of the Euclidean space  $\mathbb{R}^n$ , we suggest to apply some at least  $C^1$  continuous extrapolation operation to the generated field of the object. To explain this idea in more details let us consider the following example shown in Fig. 9, (b). In this figure we have two blue objects defined on their own pink grids and spaced from each other, so their defining grids are not overlapping. If we want to work with them, e.g. by applying some operation, we need somehow to define the distances in the points of interest of the green grid. One can extrapolate and average the distances between two pink grids and avoid full reinitialisation of the distances for both objects.

## 5.2. Algorithm for HFRep attribute definition

To set up the attributes in interior of the HFRep object  $O_{HV, HFRep}$ , we assume that we have obtained a  $C^1$  continuous distance function for a geometric shape. Now we can deal with the attributes that are parameterised by the distances as it was required by **definition 4.2**. Object attributes could be of different nature and there is no single





**Figure 10:** The comparison of the field restoration at each subdivided hierarchical cell using bilinear interpolation (a) and PHT-spline interpolation (b). In red circles we can see the  $C^0$  discontinuity in the field isolines where the cells of different size appear next to each other.

## 6.1. Step 2: Generation of the unsigned distance field

In this subsection we describe the solutions for generating UDFs. We show how some existing techniques can be used in this context and also introduce a novel method for the ADF generation.

**SDF generation.** To compute an approximate UDF, the most widely used class of methods is the distance transform (DT) [JBS06]. DTs are efficiently generated on regular grids. In this work we suggest using the vector DT in which the vector components are propagated across the uniform grid. It provides a sufficiently accurate distance approximation. We follow the typical vector DT algorithm described in [LL92] for 2D case and [Dan80] for 3D case.

A definitive way to obtain an accurate DF for the object is to numerically solve the eikonal equation or the level-set PDEs [GM19]. The numerical solution of PDE is quite time-consuming unless it is a multi-threading implementation of the method. The accuracy of the field is also highly dependent on the method. One of the robust methods for solving the eikonal equation is the fast iterative method (FIM) [JW08]. It numerically solves a nonlinear Hamilton-Jacobi PDE defined on a Cartesian grid with a scalar speed function:

$$H(\mathbf{p}, \nabla\phi) = |\nabla\phi(\mathbf{p})|^2 - \frac{1}{f^2(\mathbf{p})} = 0, \quad \forall \mathbf{p} \in \mathbf{X} \subset \mathbb{R}^n \quad (22)$$

$$\phi(\mathbf{p}) = 0, \quad \mathbf{p} \in \Gamma \subset \mathbb{R}^n$$

where  $\mathbf{X}$  is a domain in  $\mathbb{R}^n$ ,  $\Gamma$  is the boundary condition,  $\phi(\mathbf{p})$  is a travel time of the distance from the source to the grid point  $\mathbf{p}$ ,  $f(\mathbf{p})$  is a positive speed function and  $H(\mathbf{p}, \nabla\phi)$  is the Hamiltonian. The computed numerical solution is an unsigned distance on a uniform grid.

**ADF generation.** To generate UDF on the hierarchical grid we briefly outline an original adaptation of the FIM method for solving the eikonal equation that also utilises PHT-splines [WYJ\*11] capability of the accurate geometry restoration. Our algorithm partly relies on the algorithm introduced in [JW08] to inherit its advantages such as independent computation of each node and a simple data structure (an active list  $L$  or a doubly linked list) for handling node updates. A detailed description of the hierarchical FIM (HFIM) algorithm will be presented elsewhere.

The algorithm consists of two parts: (1) initialisation of the grid and (2) iterative updates of the numerical solution of the eikonal equation. First, we subdivide the space using quadtree/octree according to the values of the FRep field. We need to subdivide the exterior and interior of the FRep object with a small tree depth and its boundary with the maximum tree depth. While executing the hierarchical subdivision of the space, we also need to compute the basis functions for the PHT-splines [WYJ\*11] and reconstruct the PHT-spline surface that will be used for restoring distances in interior of each cell node.

1. We need to define the FRep object  $O_{FRep}$  using at least  $C^0$  continuous FRep function  $F_{FRep}(\mathbf{p})$ . In the Fig. 6, (a) we show the defined FRep object that we will further subdivide in the next step.
2. After we have defined  $O_{FRep}$  using  $F_{FRep}(\mathbf{p})$ , we need to subdivide space using some tree-structure according to the

algorithm to define all of them. In this work we consider such attributes as colours, microstructures, and simple 2D and volumetric textures based on noise functions parameterised by distances.

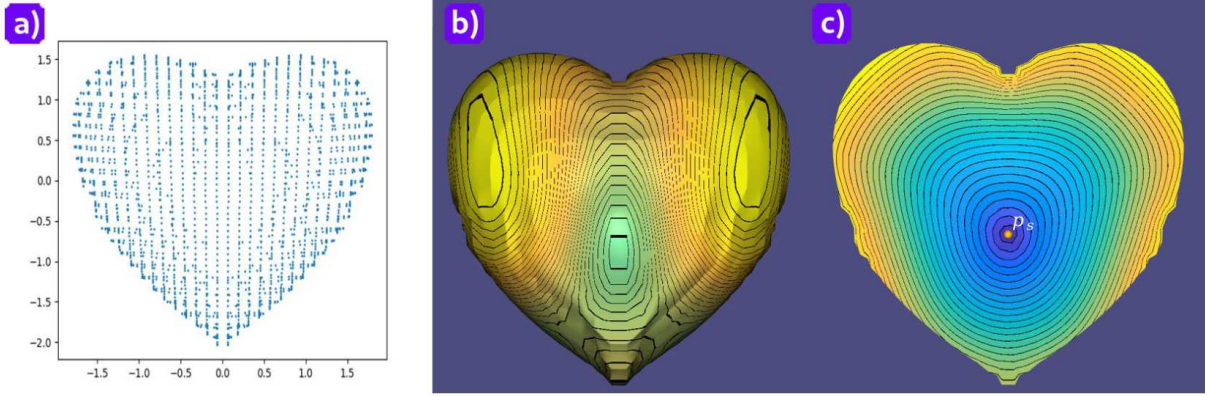
Let us formulate the basic algorithm for specifying an attribute component  $A_i$  of the  $O_{H_V, HFRep}$  on the basis of already defined geometry:

1. Depending on the nature of the attributes and how they are distributed in interior of the object  $O_{H_V, HFRep}$ , there are two possible types of object partitioning: single and multiple partitions. At this step we need to subdivide an object  $O_{H_V, HFRep}$  according to the chosen partitioning scheme.
2. Then we specify and evaluate an attribute function  $F_{A_i}(F_{HFRep}(\mathbf{p}), \mathbf{p})$  for each partition to set up the attributes at the points  $\mathbf{p} \in G$ . These functions depend on the evaluation point coordinate and are parameterised by the computed distance using  $F_{HFRep}(\mathbf{p})$  values.
3. In case when we have a multiple partitioned object with several specified attributes, we can obtain a single attribute function for all subsets  $A_i$  by applying some interpolation, e.g. transfinite interpolation [RSST01] or space-time transfinite interpolation [SFA\*15].

The more detailed discussion how to deal with attributes will be provided in section 7.

## 6. Algorithmic solutions for HFRep geometric shape generation

In this section we provide a detailed description of several particular steps of the basic algorithm outlined in the previous section. We consider a variety of combinations of FRep and SDF, ADF or IDF representations and propose a number of original solutions for solving problematic issues. The first step of the algorithm has been already discussed. In the next sections we discuss steps 2 - 4 (see Figs. 8, (b) and (c)).



**Figure 11:** HFRep based on hybridisation of FRep and IDF generated for FRep 'heart' object using the method from [RLF09]. a) the diffusion map computed on the surface of the object that is used for restoring distances at the shape boundary; b) the distances obtained at the boundary of the object shape that are shown as black isolines; c) is the tetrahedral slice of the mesh with isolines corresponding to the interior distances. The yellow point  $\mathbf{p}_s$

corresponds to the 'source' point defined in the object interior.

- 731 following rule: the exterior  $X \setminus G$  of  $O_{FRep}$  should be subdivided  
 732 with lowest tree depth, the boundary  $\partial G$  of  $O_{FRep}$  should be  
 733 subdivided with highest tree depth and the interior  $G_{in}$  of  $O_{FRep}$   
 734 should be subdivided according to the user needs. In the Fig. 6,  
 735 (b) we show the subdivided FRep object.  
 736
- 737 3. During the subdivision process of the Euclidean space, we  
 738 have to compute PHT-spline basis functions at each cell of the  
 739 constructing T-mesh including Bežier extraction operators and  
 740 new basis indices. The obtained basis functions, operators and  
 741 tensor products will be further used for the restoration of the  
 742 distance field.  
 743
- 744 4. After we finished the subdivision of the Euclidean space  
 745 according to steps two and three, we need to traverse the tree  
 746 and set to zero those vertices of the cells that store the FRep  
 747 values approximately equal to zero. The rest of the vertices are  
 748 set to a relatively huge value. Thereafter, vertices that are equal  
 749 to zero and the corresponding nodes are stored in the active list.  
 750
- 751 5. The iterative computation of the solution of the eikonal  
 752 equation on the hierarchical grid follows the logic of the  
 753 FIM algorithm [JW08], but all steps are executed taking into  
 754 account a hierarchical nature of the grid. The eikonal equation  
 755 is iteratively solved using the first order upwind Godunov  
 756 discretisation scheme that is modified for computations on the  
 757 grid with irregular steps. The computed solution is stored and  
 758 updated in all nodes that share the same vertices. The iterative  
 759 computation is finished when the active list is empty.  
 760
- 761 6. After obtaining the solution of the eikonal equation at each  
 762 corner vertex of each cell of the hierarchical grid, we can restore  
 763 the distance field using the already constructed PHT-spline surface.

764 As we have stated in subsection 3.4, the ADF field has  $C^0$  dis-  
 765 continuities that arise after the hierarchical subdivision where cells

766 of different size appear. In Fig. 10, (a), the discontinuities in the  
 767 white isolines are located in the red circles.  $C^1$  discontinuities are  
 768 introduced by the bilinear/trilinear interpolation that is used for the  
 769 field restoration in interior of each cell (see Fig. 10, a). As it can  
 770 be seen in Fig. 10, (b) the field generated by our method with PHT-  
 771 spline restoration of the field successfully solves these drawbacks.  
 772 All the isolines are continuous and smooth.

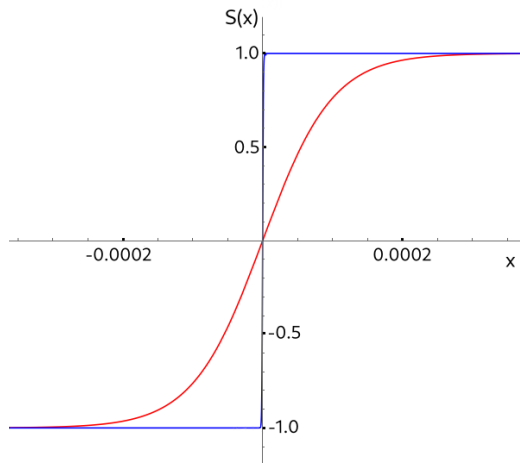
773 The accuracy of the restored field can be enhanced if an error  
 774 based refinement operation for the hierarchical is implemented.  
 775 This will allow to obtain a more accurate PHT-spline approxima-  
 776 tion of the computed field as well as a more accurate solution of  
 777 the eikonal equation on the refined grid. Our implementation of the  
 778 proposed HFIM method is also not well-optimised yet, and the al-  
 779 gorithm should be further developed for parallel computations.

780 **IDF generation.** IDFs are usually computed using the solution  
 781 of some PDE equations or, alternatively, some graph-based ap-  
 782 proach. We suggest using the approach described in [RLF09].

783 The generation of IDFs is based on propagation of the distances  
 784 computed on the boundary of the mesh in its interior. We will use  
 785 Fig. 11 with the generated IDF field to explain how this method  
 786 works. We start from triangulating an input geometric shape  $G$  of  
 787 the FRep object  $O_{FRep}$  to generate the boundary surface  $\partial G$  for fur-  
 788 ther computations. The method, described in [RLF09] was applied  
 789 to tetrahedralised meshes and consists of two parts.

790 First, we embed surface vertices  $\mathbf{p}_{b_i}$  in some  $m$ -dimensional  $\mathbb{R}^m$   
 791 space using a map  $\mathbf{p}_{b_i} \mapsto \mathbf{p}_{b_i}^* \in \mathbb{R}^m$ . This map was suggested to  
 792 compute using diffusion maps introduced in [CL06]. It can be ob-  
 793 tained by computing an eigendecomposition  $\{\lambda_k, \phi_k\}_{k=1}^n$  of a dis-  
 794 crete Laplace-Beltrami operator of the mesh. In Fig. 11(a) we show  
 795 the diffusion map obtained for the 'heart' object. The diffusion dis-  
 796 tances are computed as a Euclidean distance using obtained eigen-  
 797 values and eigenvectors [dGGV08].

After the diffusion distances were computed on the surface of  
 the mesh as it can be seen in Fig.11 (b), they are extended to the in-



**Figure 12:** Plots of hyperbolic tangent sigmoid functions with according slope values  $s_l$ . Red line:  $S_{sig}(x)$ ,  $s_l = 10^{-4}$ . Blue line:  $S_{sig}(x)$ ,  $s_l = 10^{-5}$ .

terior of the mesh using barycentric interpolation. If point  $\mathbf{q} \in G_{in}$ , then the barycentric representation of it is  $\mathbf{q} \mapsto \mathbf{q}^* = \sum_i \omega_i(\mathbf{q}) \mathbf{v}_i$ , where  $\omega_i(\cdot)$  are barycentric coordinates (e.g. mean-value coordinates in 2D [HF06] or in 3D [JSW05]). Finally, the distance in interior of the mesh can be obtained using computed diffusion distances  $F_{DF}^2(\mathbf{p}_i, \mathbf{p}_j)$  and barycentric interpolation.

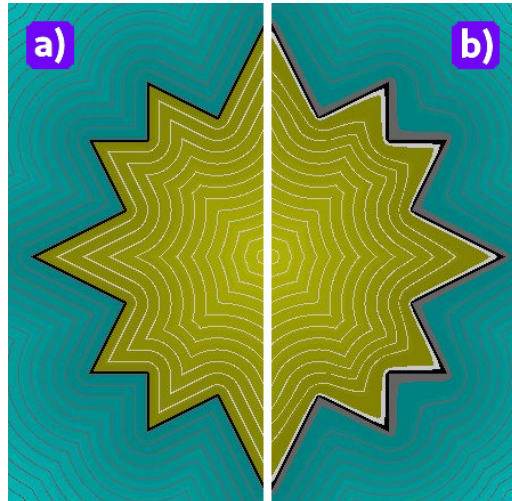
In Fig. 11 (c) we show a slice of the 'heart' object. The IDF field was computed between fixed 'source' point and other points of the mesh. One can see that the interior field is continuous, smoothly changing and following the boundary of the object. In our implementation it took 0.7 seconds to compute the tetrahedralised FRep 'heart' object. To compute the diffusion map it took 0.4 seconds and the computation of the IDF field took 115 seconds.

At this step we can apply an extrapolation operation (e.g., using a wavenumber based extrapolation [TK00]) to the obtained IDF field to propagate the distances to exterior of the object using the already computed boundary distances. This operation will allow us to make the IDF field signed at the last step of the basic algorithm.

One of the limitations of the chosen method for the IDF generation is that it was specifically developed for tetrahedral meshes [RLF09]. As a consequence, to make it compatible with other already discussed hybrid schemes, the generated tetrahedral mesh should be converted to a voxel representation and the computed IDF field interpolated in voxel centres. This method is also relatively slow to compute.

### 6.2. Step 3: Smoothing an obtained UDF

The resulting distance function  $F_{DF}(\mathbf{p})$  for SDF, ADF or IDF is unsigned and satisfies the equation (19). Having obtained UDF, we need to smooth the generated discrete field. To enforce an at least  $C^1$  continuity and essential smoothness for the obtained field, we need to use some at least  $C^1$  continuous interpolation function, e.g. B-splines or bicubic/tricubic splines [Kno00].



**Figure 13:** The demonstration of HFRep  $C^1$  continuity. a) the HFRep 'star' object that was computed with  $s_l = 0.00001$  for  $F_{sig}$ , equation (23); b) the HFRep 'star' object that was computed with  $s_l = 0.1$  for the  $F_{sig}$ , equation (23); all sharp features are smooth, i.e. the HFRep function is  $C^1$  continuous.

### 6.3. Step 4: Distinguishing between interior, boundary and exterior of the object

At the previous step we had obtained a smooth and continuous unsigned distance function  $F_{smDF}(\mathbf{p})$  defined by equation (20) that we used to compute UDF. Now, at the fourth step of the basic algorithm, we need to define the sign of UDF. To restore the sign we suggest to use a smooth step-function that depends on the values of the FRep function  $F_{FRep}(\mathbf{p})$ , defined at the first step of the basic algorithm. The step-function  $F_{st}(F_{FRep}(\mathbf{p}))$  should satisfy the following requirements:

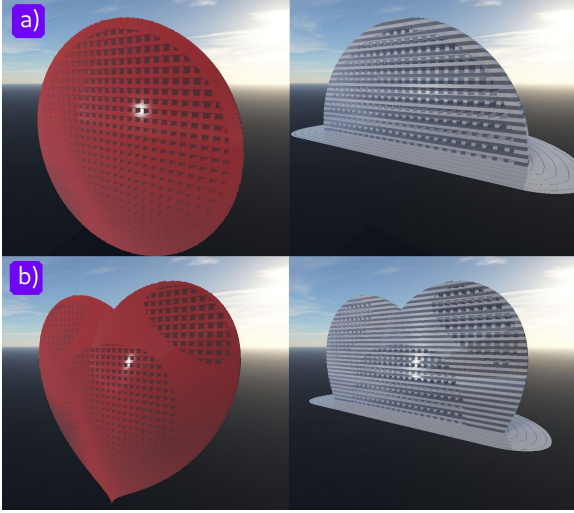
1. it is approximately equal  $-1$  when it corresponds to the exterior of the FRep object,  $F_{FRep}(\mathbf{p}) < 0$ ;
2. it should be approximately equal to  $0$  on the boundary of the FRep object,  $F_{FRep}(\mathbf{p}) = 0$ ;
3. it should be approximately equal to  $1$  inside the FRep object,  $F_{FRep}(\mathbf{p}) > 0$ ;
4. it should be at least  $C^1$  continuous everywhere in a Euclidean space  $\mathbb{R}^n$ ;
5. it should barely modify the values of UDF.

We have identified two classes of functions which satisfy these requirements. These are sigmoid functions and spline functions, particularly cubic splines with Hermite end conditions, to estimate the slopes [Kno00]. In this work we use the hyperbolic tangent sigmoid function [VMR\*88] (see Fig. 12). By controlling slope parameter  $s_l$ , it is possible to get nearly step-function behaviour around zero:

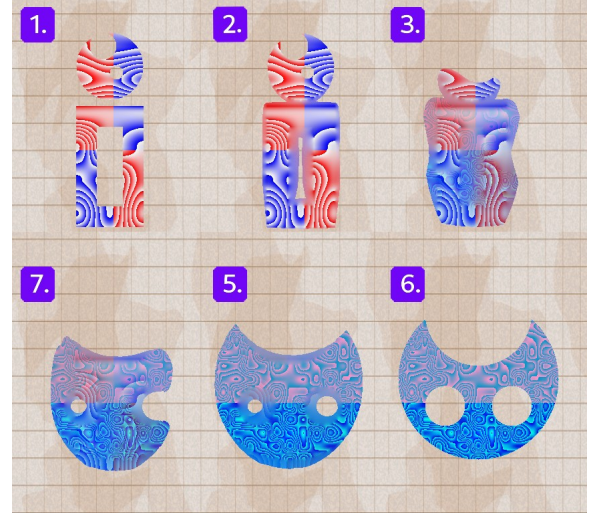
$$F_{sig}(x) = \frac{r}{1 + \exp(-2x/s_l)} - \frac{r}{2}, \quad \forall x \in \mathbb{R} \quad (23)$$

where  $r$  controls the range of the  $F_{st}(x)$  along y-axes. We need to set





**Figure 14:** The illustration of the HFRep heterogeneous object based on the FRep and SDF representations with incorporated microstructure. a) the rendered HFRep 'sphere' object using the sphere-tracing method (left) and its isolines (right); b) the rendered HFRep 'heart' object using sphere-tracing method (left) and its isolines (right);



**Figure 15:** The illustration of the metamorphosis between two HFRep textured objects using the STB and STTI techniques. The texturing was made using procedural noise functions. Supplementary video: figure15.m4v

## 877 7.1. Microstructures

In Fig. 14 we demonstrate how microstructures in interior of the  $O_{HFRep}$  object are implemented. The microstructures were defined as incorporated infinite slabs in interior of the 'sphere' and 'heart' objects using set-theoretic operations (18). The infinite slabs were defined according to [PFV\*11] as follows:

$$S(\mathbf{p}) = \sin(\mathbf{v} \odot \mathbf{p} + \boldsymbol{\phi}) + \mathbf{l}; \quad (24)$$

where  $S(\mathbf{p}) \geq 0$  is a vector function, with components defined as a set of slabs orthogonal to either X or Y or Z-axes,  $\mathbf{v}$  is a frequency vector, with components defined as the distance between parallel slabs along one of the axes,  $\mathbf{p}$  is a point  $\mathbf{p} \in X$ ,  $\boldsymbol{\phi}$  is a phase vector, with components defined as the position of slabs on one of the axes with respect to the origin and  $\mathbf{l}$ ,  $-1 < l_i < 1$  is a threshold vector that together with frequency parameters controls the thickness of each slab. Then the basic algorithm was applied to the obtained function to compute the HFRep objects with microstructures. In our implementation it took 14 seconds to generate the FRep field for the object with microstructures shown in Fig. 14(b). The computation of the distance transform took 16 seconds and the smoothing of the computed unsigned distance took 3 seconds.

Implementation was done using C++ and OpenGL. The HFRep geometric shape was computed as a scalar field which was stored in a 3D texture. Then it was passed into a fragment shader for assigning a single colour attribute and rendered using the sphere-tracing method.

## 896 7.2. Procedural textures

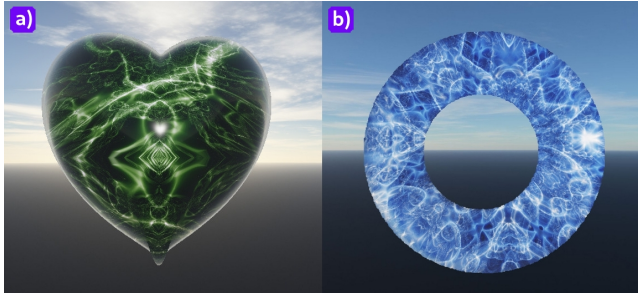
We can specify attributes as simple procedural textures. Fig. 15 shows two heterogeneous HFRep objects  $O_{H_v, HFRep}$  with coloured wooden textures that were obtained using a procedural function

852 parameter  $r = 2$  to make the function (23) be defined in the interval  
853  $(-1, 1)$  along the y-axes.

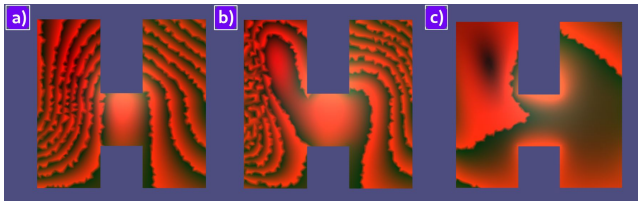
854 The continuity of the HFRep function can be visualised as it is  
855 shown in Fig. 13. In Fig. 13 (a), we show half of the 'star' object  
856 that was generated with  $s_l = 10^{-5}$  to follow the step-function shape  
857 as close as possible. In Fig. 13 (b), we show half of the 'star' object  
858 that was generated with  $s_l = 0.1$  to smooth the isolines shape. We  
859 can see that the  $C^1$  continuity of the generated distance field is pre-  
860 served and the obtained geometric shape of the object is watertight.

## 861 7. Dealing with attributes in HFRep framework

862 In this section we show how we practically work with HFRep het-  
863 erogeneous objects in terms of their attributes. In section 5 we  
864 have outlined the basic algorithm for generating HFRep attribute  
865 functions. However, there is no universal approach for dealing with  
866 HFRep object attributes because of their widely various nature. In  
867 this section we show how the proposed framework works for some  
868 representative attributes, namely, microstructures, colour and mater-  
869 ial attributes. We show the microstructures (Fig. 14), two hetero-  
870 geneous objects with defined procedural marble material parameter-  
871 ised by the SDF distance (Fig. 16), an HFRep 2D 'H' object  
872 with three different parameterisations of the procedural wood func-  
873 tion by the IDF distance (Fig. 17), a heterogeneous model of the  
874 COVID-19 virus cell (Fig. 18) and two models of metamorphosis  
875 dealing with a dynamic (time-variant) smooth transition from one  
876 HFRep object to another (Figs. 15 and 19).



**Figure 16:** Two heterogeneous HFRep objects a) 'heart' and b) torus with differently parameterised and coloured marble materials.



**Figure 17:** The HFRep 'H' object that is textured using procedural function 25 modelling the 'wood' texture. This function was differently parameterised (a), (b), (c) by computed HFRep based on FRep and IDF for the given object.

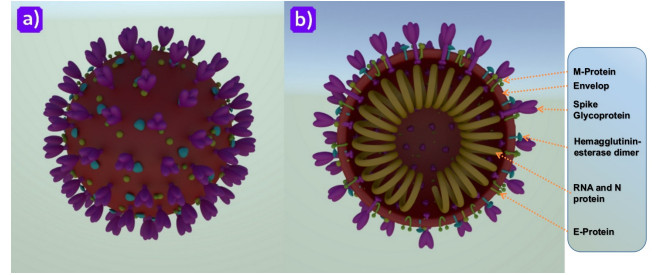
$f_{wood}(\mathbf{p})$ . This function is constructed using hash table  $htab(\mathbf{p})$  allowing for random sampling of the position values  $\mathbf{p}$  multiplied by the frequency  $v$ . The procedural function for the wood can be defined as follows:

$$\begin{aligned} g(\mathbf{p}) &= htab(\mathbf{p} \cdot v) \cdot c; \\ f_{wood}(\mathbf{p}) &= g(\mathbf{p}) - int(g(\mathbf{p})); \end{aligned} \quad (25)$$

where  $c > 1$  is a constant,  $g(\mathbf{p})$  is a noise function,  $int(g(\mathbf{p}))$  is an integer part of the function  $g(\mathbf{p})$  output value. To parameterise  $f_{wood}(\mathbf{p})$  by the distance, we assign the distance values to the frequency parameter  $v$ .

Then a simple segmentation of the geometric shape of the objects was done (see Fig. 15,(1)). We split the shape into four regions and assign colours using the obtained HFRep distance function  $F_{HFRep}(\mathbf{p})$  and procedural function  $f_{wood}(\mathbf{p})$  that defines the texture of the wood. The generated objects were used as inputs for 2D heterogeneous metamorphosis on the basis of the space-time blending (STB) method to handle geometry transformation and the space-time transfinite interpolation (STTI) to handle colour transformation [TAF\*20]. This example was implemented using C++ and OpenCV.

In Fig. 17 we show three textured 'H' HFRep objects. The textures for these objects were generated using three different IDF based parameterisations of the procedural function for the wood. In Fig. 16 we show two 3D HFRep objects 'heart', (a), and 'torus', (b), with procedurally defined marble material with different parameterisation by the HFRep distance.



**Figure 18:** The COVID-19 cell model obtained as a combination of the HFRep and SDF functions. a) exterior of the virus cell; b) interior of the virus cell.

### 917 7.3. Voxel based attributes

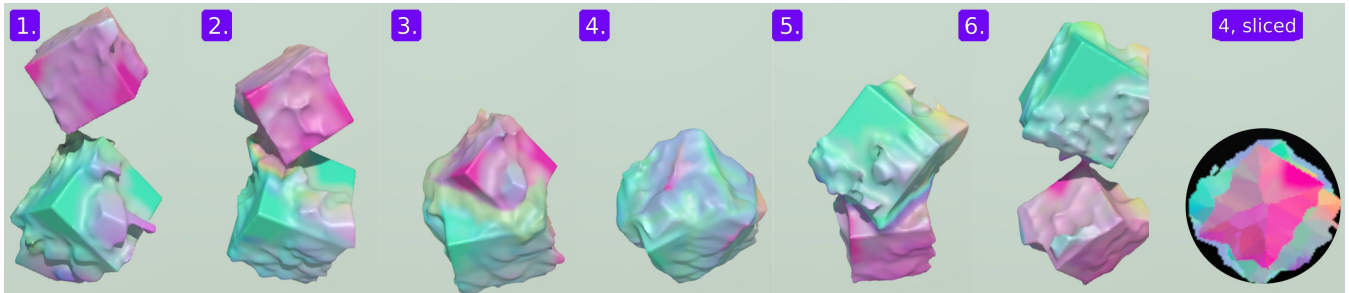
918 In this subsection we discuss how HFRep objects with voxel-based  
919 attributes can be defined. Two following examples were implemented  
920 in SideFX Houdini using the OpenVDB library.

921 In Fig. 18 we show a 3D model of the COVID-19 cell that was  
922 obtained using 207 set-theoretic operations. In Fig. 18, (b), we can  
923 see the interior structure of the COVID-19 cell [MG20]. The central  
924 part representing the RNA and N-protein was defined using SDF  
925 that was further combined with the HFRep spherical shell of the  
926 cell. The M-protein was also defined as a combination of the SDF  
927 arc and two HFRep spheres. The rest of the elements were defined  
928 using HFRep. Each element is mono-coloured and colours are  
929 assigned per-voxel.

930 Fig. 19 demonstrates a 3D metamorphosis between two heterogeneous  
931 objects that are a combination of the HFRep and SDF defined objects  
932 [TAPA20]. This example served as one of tests for the '4D Cubism'  
933 project [CMPA18]. The input and target objects are two SDF cubes  
934 spaced from each other. These input shapes were segmented using  
935 an octree data structure to make it possible a local faceting and  
936 distortions. Two colours were assigned to them per-voxel. Then  
937 different HFRep and SDF 'cubist' features were assigned to selected  
938 areas of the two basic SDF cubes, which were coloured per-voxel  
939 as randomly chosen colours from the specified range. Then we apply  
940 the same combination of methods as we have discussed before for  
941 the 2D metamorphosis. The generated colour and geometric shape  
942 transformations happen simultaneously and interconnectedly. In  
943 Fig. 19, 4 (sliced), we show how the interior of the object is  
944 transformed during the 3D metamorphosis process.

### 945 8. Conclusions

946 In this work we have introduced a theoretical and practical framework  
947 for modelling volumetric heterogeneous objects on the basis of  
948 a novel unifying functionally-based hybrid representation called  
949 HFRep. First, we have identified four conventional representational  
950 schemes related to scalar fields of different kinds, namely FRep,  
951 SDF, ADF and IDF, suggested a formalisation of those approaches  
952 and described their advantages and drawbacks. This has allowed  
953 us to formulate the requirements for a unifying hybrid representation.  
954 The defining functions in the core of HFRep are continuous and  
955 have a distance property everywhere in a Euclidean space.



**Figure 19:** Metamorphosis between multiple coloured objects using the STB and STTI techniques. Colours of the initial objects are procedurally defined per voxel. Supplementary videos: *figure19.m4v* and *figure19\_slice.m4v*.

956 They also have several other useful properties. We have defined the 995  
 957 mathematical basics of the representation and developed an algo- 996  
 958 rithmic procedure allowing to generate HFRep objects in terms of 997  
 959 their geometry and attributes.

960 To make our approach practical, we have provided a detailed 1000  
 961 description of the main steps of the algorithm and identified some 1001  
 962 problematic issues associated with them. This has required employ- 1002  
 963 ing a number of techniques of different nature, separately and in 1003  
 964 combination. Some of these techniques were already described in 1004  
 965 literature, others had to be improved or developed. In particular, a 1005  
 966 new FIM algorithm for solving the eikonal equation on hierarchical 1006  
 967 grids has been developed. 1007  
 1008

968 To show how the proposed framework works, we have illustrated 1009  
 969 the algorithmic process with a number of implemented examples, 1010  
 970 including those that deal with colour, material and microstructure 1011  
 971 attributes in the interior of functionally-defined shapes in the con- 1012  
 972 text of time-variant modelling. 1013

973 While the boundary representation will remain the main and pre- 1014  
 974 vailing instrument for geometric modelling, we believe that the 1015  
 975 functionally-based representations generalising a well-established 1016  
 976 implicit modelling approach, are becoming more important in the 1017  
 977 context of some modern applications. Hopefully, HFRep that em- 1018  
 978 braces advantages and circumvents drawbacks of FRep, SDF, ADF, 1019  
 979 IDF will find its applications. 1020  
 1021  
 1022

980 Future work will be concerned with developing operations over 1023  
 981 HFRep objects in the context of different applications, especially 1024  
 982 related to physical simulation, additive manufacturing and visual 1025  
 983 effects. In technical terms, we aim to develop an HFRep field ex- 1026  
 984 trapolation procedure for IDFs and beyond the computing domain. 1027  
 985 One of the interesting directions will be the introduction of the at- 1028  
 986 tribute definition in the interior of the volumetric object using the 1029  
 987 diffusion-based IDFs. We also consider a further generalisation of 1030  
 988 the FIM method for 3D hierarchical grids. 1031  
 1032  
 1033

## 989 References

990 [ABGA04] ALLEGRE R., BARBIER A., GALIN E., AKKOUCHE S.: A 1037  
 991 hybrid shape representation for free-form modelling. In *Proceedings of* 1038  
 992 *the Shape Modeling International 2004* (2004), SMI '04, IEEE Com- 1039  
 993 puter Society, pp. 7–18. 3 1040  
 994 [AKK\*02] ADZHIEV V., KARTASHEVA E., KUNII T., PASKO A., 1041

SCHMITT B.: Cellular-functional modeling of heterogeneous objects. In *Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications* (2002), SMA '02, ACM, pp. 192–203. 3

[BDN16] BARCLAY J., DHOKIA V., NASSEHI A.: Additive manufacturing simulation using signed distance fields. In *Sustainable Design and Manufacturing 2016* (2016), Setchi R., Howlett R. J., Liu Y., Theobald P., (Eds.), Springer International Publishing, pp. 435–444. 5

[BKW\*18] BADER C., KOLB D., WEAVER J. C., SHARMA S., HOSNY A., COSTA J., OXMAN N.: Making data matter: Voxel printing for the digital fabrication of data across scales and domains. *Science Advances* 4, 5 (2018). 2, 3

[Bom67] BOMAN J.: Differentiability of a function and of its compositions with functions of one variable. *MATHEMATICA SCANDINAVICA* 20 (1967), 249–268. 9

[Bow95] BOWYER A.: Svlis set-theoretic kernel modeller: Introduction and user manual. Information Geometers. 2

[BST04] BISWAS A., SHAPIRO V., TSUKANOV I.: Heterogeneous material modeling with distance fields. *Computer Aided Geometric Design* 21, 3 (2004), 215 – 242. 3, 5, 7

[BVG19] BÁLINT C., VALASEK G., GERGÓ L.: Operations on signed distance functions. Conference: The 11th Conference of PhD Students in Computer Science, pp. 1–12. 5

[CLO6] COIFMAN R. R., LAFON S.: Diffusion maps. *Applied and Computational Harmonic Analysis* 21, 1 (2006), 5 – 30. Special Issue: Diffusion Maps and Wavelets. 6, 13

[CMPA18] CORKER-MARIN Q., PASKO A., ADZHIEV V.: 4d cubism: Modeling, animation, and fabrication of artistic shapes. *IEEE Computer Graphics and Applications* 38, 03 (2018), 131–139. 16

[CT00] CHEN M., TUCKER J. V.: Constructive Volume Geometry. *Computer Graphics Forum* (2000). doi:10.1111/1467-8659.00464. 3

[CWW13] CRANE K., WEISCHEDEL C., WARDETZKY M.: Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans. Graph.* 32, 5 (2013), 152:1–152:11. 6

[Dan80] DANIELSSON P.-E.: Euclidean distance mapping. *Computer Graphics and Image Processing* 14, 3 (1980), 227 – 248. 12

[DCL\*08] DENG J., CHEN F., LI X., HU C., TONG W., YANG Z., FENG Y.: Polynomial splines over hierarchical t-meshes. *Graph. Models* 70, 4 (2008), 76–86. 6

[dGGV08] DE GOES F., GOLDENSTEIN S., VELHO L.: A hierarchical segmentation of articulated bodies. In *Proceedings of the Symposium on Geometry Processing* (2008), SGP '08, Eurographics Association, p. 1349–1356. 13

[DTD\*15] DOUBROVSKI E., TSAI E., DIKOVSKY D., GERAEDTS J., HERR H., OXMAN N.: Voxel-based fabrication through material property mapping: A design method for bitmap printing. *Computer-Aided Design* 60 (2015), 3 – 13. Material Ecology. 3



- [DZ10] DELFOUR M., ZOLSIO J.-P.: *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*. Society for Industrial and Applied Mathematics, 2010. 4
- [FPRJ00] FRISKEN S. F., PERRY R. N., ROCKWOOD A. P., JONES T. R.: Adaptively sampled distance fields: A general representation of shape for computer graphics. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (2000)*, SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., pp. 249–254. 2, 5, 6, 7
- [FSP15] FRYAZINOV O., SANCHEZ M., PASKO A.: Shape conforming volumetric interpolation with interior distances. *Comput. Graph.* 46, C (2015), 149–155. 3, 7
- [GFO18] GIBOU F., FEDKIW R., OSHER S.: A review of level-set methods and some recent applications. *Journal of Computational Physics* 353 (2018), 82 – 109. 2, 3
- [GM19] GÓMEZ J. V., ÁLVAREZ D., GARRIDO S., MORENO L.: Fast methods for eikonal equations: An experimental survey. *IEEE Access* 7 (2019), 39005–39029. 12
- [HDEE14] H. DYER R., E. EDMUNDS D.: *From Real to Complex Analysis*. Springer International Publishing, 2014. 3
- [HF06] HORMANN K., FLOATER M. S.: Mean value coordinates for arbitrary planar polygons. *ACM Trans. Graph.* 25, 4 (2006), 1424–1441. 14
- [HL09] HILLER J., LIPSON H.: Design and analysis of digital materials for physical 3d voxel printing. *Rapid Prototyping* 15 (2009), 137–149. 3
- [JBS06] JONES M. W., BAERENTZEN J. A., SRAMEK M.: 3d distance fields: a survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics* 12, 4 (2006), 581–599. 1, 2, 12
- [JSW05] JU T., SCHAEFER S., WARREN J.: Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.* 24, 3 (2005), 561–566. 14
- [JW08] JEONG W., WHITAKER R.: A fast iterative method for eikonal equations. *SIAM Journal on Scientific Computing* 30, 5 (2008), 2512–2534. 12, 13
- [Kar99] KARCZMARCZUK J.: Geometric modelling in functional style. In *Proc. of the III Latino-American Workshop on Functional Programming, CLAPP'99 (1999)*, pp. 8–9. 5
- [KBDH99] KUMAR V., BURNS D., DUTTA D., HOFFMANN C.: A framework for object modeling. *Computer-Aided Design* 31, 9 (1999), 541 – 556. 3
- [KDB16] KOSCHIER D., DEUL C., BENDER J.: Hierarchical hp-adaptive signed distance fields. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (2016)*, SCA '16, Eurographics Association, pp. 189–198. 5, 6
- [KKL15] KIM J.-H., KIM C.-H., LEE J.: A hybrid sdf for the detailed representation of liquid–solid mixed surfaces. *Computer Animation and Virtual Worlds* 26, 5 (2015), 527–536. 3, 5
- [Kno00] KNOTT G. D.: *Interpolating Cubic Splines*, vol. 18 of *Progress in Computer Science and Applied Logic*. Birkhäuser Basel, 2000. 14
- [KT07] KOU X., TAN S.: Heterogeneous object modeling: A review. *Computer-Aided Design* 39, 4 (2007), 284 – 301. 1
- [LEM\*17] LIVESU M., ELLERO S., MARTÍNEZ J., LEFEBVRE S., ATTENE M.: From 3D Models to 3D Prints: An Overview of the Processing Pipeline. *Computer Graphics Forum* (2017). 1
- [LFAB14] LEI S., FRANK M. C., ANDERSON D. D., BROWN T. D.: A method to represent heterogeneous materials for rapid prototyping: the matryoshka approach. *Rapid Prototyping Journal* 20, 5 (2014), 390–402. 1, 2
- [LFSL20] LI B., FU J., SHANG C., LIN Z.: Review of heterogeneous material objects modeling in additive manufacturing. *Visual Computing for Industry, Biomedicine, and Art* 3, 6 (2020). 1
- [LL92] LEYMARIE F., LEVINE M.: Fast raster scan distance propagation on the discrete rectangular lattice. *CVGIP: Image Understanding* 55, 1 (1992), 84–94. 10, 12
- [LRF10] LIPMAN Y., RUSTAMOV R. M., FUNKHOUSER T. A.: Biharmonic distance. *ACM Trans. Graph.* 29, 27 (2010), 27:1–27:11. 3
- [LRL11] LIU Y.-S., RAMANI K., LIU M.: Computing the inner distances of volumetric models for articulated shape description with a visibility graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 12 (2011), 2538–2544. 6
- [LSD04] LAEHYUN KIM, SUKHATME G. S., DESBRUN M.: A haptic-rendering technique based on hybrid surface representation. *IEEE Computer Graphics and Applications* 24, 2 (2004), 66–75. 3
- [MG20] MOUSAVIZADEH L., GHASEMI S.: Genotype and phenotype of covid-19: Their roles in pathogenesis. *Journal of Microbiology, Immunology and Infection* (2020). doi:10.1016/j.jmii.2020.03.022. 16
- [NKI\*18] NGO T. D., KASHANI A., IMBALZANO G., NGUYEN K. T., HUI D.: Additive manufacturing (3d printing): A review of materials, methods, applications and challenges. *Composites Part B: Engineering* 143 (2018), 172 – 196. 3
- [NMK\*06] NEALEN A., MÜLLER M., KEISER R., BOXERMAN E., CARLSON M.: Physically based deformable models in computer graphics. *Computer Graphics Forum* 25, 4 (2006), 809–836. 1
- [PASS95] PASKO A., ADZHIEV V., SOURIN A., SAVCHENKO V.: Function representation in geometric modeling: concepts, implementation and applications. *The Visual Computer* 11, 8 (1995), 429–446. 2, 4, 5, 10, 11
- [PASS01] PASKO A., ADZHIEV V., SCHMITT B., SCHLICK C.: Constructive hypervolume modeling. *Graphical Models* 63, 6 (2001), 413 – 442. 3, 7
- [PFV\*11] PASKO A., FRYAZINOV O., VILBRANDT T., FAYOLLE P.-A., ADZHIEV V.: Procedural function-based modelling of volumetric microstructures. *Graphical Models* 73, 5 (2011), 165 – 181. 15
- [PS12] PATANÈ G., SPAGNUOLO M.: Smi 2012: Full local approximation of scalar functions on 3d shapes and volumetric data. *Comput. Graph.* 36, 5 (2012), 387–397. 7
- [PT92] PAYNE B. A., TOGA A. W.: Distance field manipulation of surface models. *IEEE Computer Graphics and Applications* 12, 1 (1992), 65–71. 5, 11
- [Req80] REQUICHA A. G.: Representations for rigid solids: Theory, methods, and systems. *ACM Comput. Surv.* 12, 4 (1980), 437–464. 2
- [RLF09] RUSTAMOV R. M., LIPMAN Y., FUNKHOUSER T.: Interior distance using barycentric coordinates. In *Proceedings of the Symposium on Geometry Processing (2009)*, SGP '09, Eurographics Association, pp. 1279–1288. 6, 7, 10, 13, 14
- [RMD11] REINER T., MÜCKL G., DACHSBACHER C.: Smi 2011: Full paper: Interactive modeling of implicit surfaces using a direct visualization approach with signed distance functions. *Comput. Graph.* 35, 3 (2011), 596–603. 11
- [RRSS16] REGLI W., ROSSIGNAC J., SHAPIRO V., SRINIVASAN V.: The new frontiers in computational modeling of material structures. *Computer-Aided Design* 77 (2016), 73 – 85. 2
- [RSST01] RVACHEV V. L., SHEIKO T. I., SHAPIRO V., TSUKANOV I.: Transfinite interpolation over implicitly defined sets. *Computer Aided Geometric Design* 18, 3 (2001), 195 – 220. 12
- [Rva82] RVACHEV V. L.: *Theory of R-functions and Some Applications*. Naukova Dumka, 1982. (rus). 10, 11
- [Sch97] SCHECHTER E.: Chapter 25 - fréchet derivatives. In *Handbook of Analysis and Its Foundations*, Schechter E., (Ed.). Academic Press, San Diego, 1997, pp. 661 – 687. 4
- [SFA\*15] SANCHEZ M., FRYAZINOV O., ADZHIEV V., COMNINOS P., PASKO A.: Space-time transfinite interpolation of volumetric material properties. *IEEE Transactions on Visualization and Computer Graphics* 21, 2 (2015), 278–288. 12
- [SG17] SHARMA G. K., GURUMOORTHY B.: A hybrid approach to define and represent material distribution in heterogeneous objects. *Computer-Aided Design and Applications* 14, 1 (2017), 70–82. 3

- 1168 [SRGB14] SOLOMON J., RUSTAMOV R., GUIBAS L., BUTSCHER A.:  
1169 Earth mover's distances on discrete surfaces. *ACM Trans. Graph.* 33, 4  
1170 (2014), 67:1–67:12. [2](#), [7](#)
- 1171 [Sul15] SULLIVAN A.: Hybrid adaptively sampled distance fields, 2015.  
1172 Patent number: 9122270, United States, US patent. [3](#)
- 1173 [TAF\*20] TERESHIN A., ADZHIEV V., FRYAZINOV O.,  
1174 MARRINGTON-REEVE F., PASKO A.: Automatically controlled  
1175 morphing of 2d shapes with textures. *SIAM Journal on Imaging*  
1176 *Sciences* 13, 1 (2020), 78–107. [16](#)
- 1177 [TAFP19] TERESHIN A., ADZHIEV V., FRYAZINOV O., PASKO A.: Hy-  
1178 brid Function Representation with Distance Properties. In *Eurographics*  
1179 *2019 - Short Papers* (2019), Cignoni P., Miguel E., (Eds.), The Euro-  
1180 graphics Association, pp. 17–20. [2](#)
- 1181 [TAPA20] TERESHIN A., ANDERSON E., PASKO A., ADZHIEV V.:  
1182 Space-time blending for heterogeneous objects. In *Eurographics 2020*  
1183 *- Short Papers* (2020), Wilkie A., Banterle F., (Eds.), The Eurographics  
1184 Association. [9](#), [16](#)
- 1185 [TF18] TANG Y., FENG J.: Multi-scale surface reconstruction based on  
1186 a curvature-adaptive signed distance field. *Computers Graphics* 70  
1187 (2018), 28 – 38. *CAD/Graphics* 2017. [5](#), [6](#)
- 1188 [TK00] TAM C. K., KURBATSKII K. A.: A wavenumber based extrap-  
1189 olation and interpolation method for use in conjunction with high-order  
1190 finite difference schemes. *J. Comput. Phys.* 157, 2 (2000), 588–617. [14](#)
- 1191 [TP11] TSUKANOV I., POSIREDDY S. R.: Hybrid method of engineer-  
1192 ing analysis: Combining meshfree method with distance fields and col-  
1193 location technique. *Journal of Computing and Information Science in*  
1194 *Engineering* 11 (2011), 031001:1–031001:9. [3](#)
- 1195 [TSNI10] TAKAYAMA K., SORKINE O., NEALEN A., IGARASHI T.:  
1196 Volumetric modeling with diffusion surfaces. In *ACM SIGGRAPH Asia*  
1197 *2010 Papers* (2010), no. Article 180 in SIGGRAPH ASIA '10, Associa-  
1198 tion for Computing Machinery. [2](#)
- 1199 [VMR\*88] VOGL T. P., MANGIS J. K., RIGLER A. K., ZINK W. T.,  
1200 ALKON D. L.: Accelerating the convergence of the back-propagation  
1201 method. *Biological Cybernetics* 59, 4 (1988), 257–263. [14](#)
- 1202 [WYJ\*11] WANG J., YANG Z., JIN L., DENG J., CHEN F.: Parallel and  
1203 adaptive surface reconstruction based on implicit pht-splines. *Computer*  
1204 *Aided Geometric Design* 28, 8 (2011), 463 – 474. *Solid and Physical*  
1205 *Modeling* 2010. [1](#), [2](#), [3](#), [5](#), [12](#)
- 1206 [You83] YOUNG I. T.: Image analysis and mathematical morphology, by  
1207 j. serra. academic press, london, 1982, xviii + 610 p. \$90.00. *Cytometry*  
1208 4, 2 (1983), 184–185. [5](#)
- 1209 [YYW12] YUAN Z., YU Y., WANG W.: Object-space multiphase im-  
1210 plicit functions. *ACM Trans. Graph.* 31, 4 (2012). [2](#)
- 1211 [I01] ĐURIKOVIČ R., CZANNER S., INOUE H.: Growth animation of  
1212 human organs. *The Journal of Visualization and Computer Animation*  
1213 12, 5 (2001), 287–295. [5](#)