

METHOD AND SYSTEM FOR INDEXING AN N-DIMENSIONAL OBJECT

RELATED APPLICATION

This patent application claims priority of US provisional patent application No. 63/524,501, entitled "MÉTHODE ET SYSTÈME POUR INDEXER UN OBJET À
5 N-DIMENSIONS", filed on June 30, 2023, the specification of which is hereby incorporated by reference.

TECHNOLOGICAL FIELD

The technology relates to the field of indexing. More specifically, this patent application relates to a method and system for indexing an n-dimensional object.

10 PRIOR ART

Overview of the problem

Indexing n-dimensional objects aims to associate an index with any point having n coordinates. This point can belong to a set of points called an n-dimensional object. The coordinates are any numerical values which can be either pure spatial
15 coordinates: Euclidean coordinates or cartographic coordinates for example, or coordinates representing digital metadata or a mixture of the two. An n-dimensional object can come from various geometric or mathematical modeling or from data acquisition by physical sensor systems of various types and in particular spatial ones such as lidar sensors. In all cases, the indexing of n-
20 dimensional objects requires the establishment of a cartography of the object and a coordinate system or benchmark governing the space in which it is located. There is the possibility of using different local or global coordinate systems which communicate with each other through conversion mechanisms between systems. Once the points are located precisely in space, it is possible to begin to index
25 them.

Prior art approaches and solutions for unstructured files

For unstructured files associated with the geometry of n-dimensional objects, the simplest indexing systems are the numerical sorting systems used in the case of unstructured n-dimensional data, the points being sorted for example according to increasing values of their Euclidean coordinates in x, then in y then in z and so on. Indexing can also be carried out by sorting on parametric or spherical coordinates such as longitudes and latitudes for geolocation. Indexing is therefore, carried out by establishing a sorted table of coordinates and the index of a given set of coordinates is therefore obtained by sequentially traversing the table to obtain the index of the point in the table.

- 10 It will be appreciated by the skilled addressee that these mechanisms of searching for the indices corresponding to sets of coordinates are costly in terms of time and memory space and become prohibitive or unusable for real-time or near real-time indexing.

Prior art approaches and solutions for structured files

- 15 For structured files, with polygonal meshes for example triangulated, or meshes made up of patches of parametric surfaces of the Bezier, B-splines or Nurb's type, the indexing systems work in two stages, firstly the indexing of the meshes then subsequently the indexing of points in the cells of the mesh.

- 20 In all cases the systems return the indices in a classic way from tables of data structures with queries and sorting as previously costly in terms of time and space which is a serious drawback. There is a need for a method and system that will overcome at least one of the above-identified drawbacks.

SUMMARY

- 25 According to a broad aspect, there is disclosed a computer-implemented method of indexing an n-dimensional object comprising a plurality of points, the method comprising obtaining a mesh of the n-dimensional shape of the object, the mesh comprising at least one cell; determining a chaining traversing the at least one cell of the mesh; the chaining comprising for each cell a symbol traversing the cell and joining two vertices of the cell; for each cell: associating an SFC metacurve

with each symbol associated with two vertices; and mapping the at least one SFC metacurve onto the object using a corresponding mapping function to associate with each point of the plurality of points a corresponding index of the at least one SFC metacurve.

- 5 According to one or more embodiments, the mesh comprises a plurality of cells; the determined chaining traverses the plurality of cells of the mesh; and the mapping is performed in order to associate with each point of the plurality of points a corresponding index of the plurality of SFC metacurves.

According to one or more embodiments, the symbol is an arc.

- 10 According to one or more embodiments, the computer-implemented method further comprises providing an indication of the mapping.

According to one or more embodiments, the providing of an indication of the mapping comprises at least one of saving the indication of the mapping in a file, displaying the indication of the mapping and providing the indication of the

- 15 mapping to a remote processing device.

According to a broad aspect, there is disclosed a computing device comprising: at least one processor; a display device; an input/output interface; a memory including instructions which when executed perform a method of indexing an n-dimensional object comprising a plurality of points, the method comprising:

- 20 obtaining via the input/output interface a mesh of the n-dimensional shape of the object, the mesh comprising at least one cell; determining, by means of the at least one processor, a chaining traversing the at least one cell of the mesh; the chaining comprising for each cell a symbol traversing the cell and joining two vertices of the cell; for each cell and by means of the at least one processor:
25 associating an SFC metacurve with each symbol associated with two vertices; and by means of the at least one processor, mapping the at least one SFC metacurve onto the object using a corresponding mapping function in order to associate with each point of the plurality of points a corresponding index of the at least one SFC metacurve; and a bus for pairing said at least one processor, said

display device, said input/output interface and said memory.

According to a broad aspect, there is disclosed a computer-readable physical memory storing statements and instructions for execution by a computer, said statements and instructions comprising: encoding means for obtaining a mesh of
5 the n-dimensional shape of the object, the mesh comprising at least one cell; encoding means for determining a chaining running through the at least one cell of the mesh; the chaining comprising for each cell a symbol traversing the cell and joining two vertices of the cell; coding means for associating an SFC metacurve with each symbol associated with two vertices of each cell; and coding
10 means for mapping the at least one SFC metacurve onto the object using a corresponding mapping function in order to associate with each point of the plurality of points a corresponding index of the at least one SFC metacurve.

According to a broad aspect, there is disclosed a computer program comprising computer-executable instructions which, when executed by a computer, perform
15 the steps of the computer-implemented method disclosed above.

In accordance with one or more embodiments, there is disclosed the computer-implemented method further comprises subdividing at least one given cell of the mesh to obtain at least two sub-cells within the given cell; wherein the chaining traversing the given cell traverses each of the at least two sub-cells, the chaining
20 comprising a symbol for each of the at least two sub-cells and wherein the SFC metacurve associated with the given cell comprises at least two SFC metacurves, each of the at least two SFC metacurves being associated with a corresponding sub-cell of the given cell.

BRIEF DESCRIPTION OF THE FIGURES

25 The invention and its advantages will become apparent in greater detail in the following description, with examples given by way of illustration with reference to the appended figures.

- Figure 1 illustrates the functional diagram of the deformation pipeline;
- Figure 2 illustrates an example of the SFC metacurve deformation process;
- Figure 3 illustrates the U and W paths;
- Figure 4 illustrates a procedural topology map;
- 5 Figure 5 illustrates odd and even generation;
- Figure 6 illustrates 4x4U and 4x4W chaining;
- Figure 7 illustrates 5x5U frames;
- Figure 8 illustrates 6x6U frames;
- Figure 9 illustrates 4x4W0 friezes;
- 10 Figure 10 illustrates U-shaped rectangular cycles;
- Figure 11 illustrates 12x12W square-shaped W cycles;
- Figure 12 shows a 12x12W square-shaped W cycle;
- Figure 13 illustrates a Peano graph and its associated graph;
- Figure 14 illustrates a double Peano cycle;
- 15 Figure 15 illustrates a Peano map of serpentine metacurves;
- Figure 16 illustrates a metacurve pattern;
- Figure 17 illustrates a rectangular Gray's metacurve;
- Figure 18 illustrates SFC metacurves on generalized cylinders;
- Figure 19 illustrates Hilbert curves of revolution;
- 20 Figure 20 illustrates spiral SFCs on parametric spheres;
- Figure 21 illustrates examples of supertoroids;

Figure 22 illustrates Dupin and Klein-Jeener surfaces;

Figure 23 illustrates a hyperbolic paraboloid and a Moebius strip;

Figure 24 illustrates a diagram of the PSFC subsystem;

Figure 25 illustrates SFC applied to parametric surfaces;

5 Figure 26 illustrates the deformation of a Gray metacurve in 3 dimensions;

Figure 27 illustrates a PSFC version of the Newell teapot;

Figure 28 illustrates a PSFC version of Catmull's elephant;

Figure 29 illustrates the organization of PSFC data;

Figure 30 illustrates an interpolation of an SFC metacurve;

10 Figure 31 illustrates a Gumbo U-shaped topological map;

Figure 32 illustrates an ear traversed by a simple Peano circuit;

Figure 33 illustrates W-type patches;

Figure 34 illustrates a simple Peano circuit on the cube;

Figure 35 illustrates a PH and a triangle;

15 Figure 36 illustrates an SFC deformation in the plane;

Figure 37 illustrates a plane tiling of deformed SFCs;

Figure 38 illustrates the code for the parSPH function;

Figure 39 illustrates the code for the sfcSPH function;

Figure 40 illustrates the code for the tabSPH function;

20 Figure 41 illustrates the code for the SPH0 function;

Figure 42 illustrates the code for the SPH1 function;

Figure 43 illustrates the code for the SPH2 function;

Figure 44 illustrates the code for the SPH3 function;

Figure 45 illustrates the code for the DC0uwLL function;

Figure 46 illustrates the code for the CD0uwLL function;

5 Figure 47 illustrates the code for the PATCHuv function;

Figure 48 illustrates the code for the PATCHsfc function;

Figure 49 illustrates the code for the PTsfc function;

Figure 50 illustrates the code for the INTERPOL function;

Figure 51 illustrates the code for the SOLUTana function;

10 Figure 52 illustrates a method for indexing an N-dimensional object; and

Figure 53 illustrates a computer system adapted to implement the present technology.

DETAILED DESCRIPTION

15 It will be appreciated that there is described in particular a computer-implemented method for indexing an n-dimensional object comprising a plurality of points. It will be appreciated by those skilled in the art that indexing an n-dimensional object is of great interest in many applications, as will be explained below.

One such method is disclosed in Figure 52. While examples of 3-dimensional objects will be disclosed below, it will be appreciated by the person skilled in the art that the technology presented can easily be extended to a dimension greater than 3.
20

According to step 100 of Figure 52, a mesh of the shape in n-dimensions is obtained. The mesh comprises at least one cell. In one or more embodiments, the mesh comprises a plurality of cells.

In one or more embodiments of the method, the mesh of the n-dimensional shape is created. The creation of such mesh according to one or more embodiments is detailed below.

5 According to step 200 of Figure 52, a chain running through the at least one cell of the mesh is determined. It will be appreciated that the chaining comprises for each cell a symbol traversing the cell and joining two vertices of the cell. It will be appreciated by those skilled in the art that the chaining can be determined according to several embodiments as explained below.

10 In one or more embodiments in which the mesh comprises a plurality of cells, the determined chaining traverses the plurality of cells of the chaining.

In one or more embodiments, the symbol joining two vertices of the cell is an arc. It will be appreciated by those skilled in the art that any other symbol may alternatively be used.

15 It will be appreciated that the determined chaining is referred to as a Peano map in the following.

20 According to step 300 of Figure 52, for each cell of the mesh, an SFC metacurve is associated with each symbol associated with two vertices. It will be appreciated that the association of the SFC metacurve with each symbol associated with two vertices can be done in different ways and will present many advantages depending on the way selected.

25 According to step 400 of Figure 52, a mapping of the at least one SFC metacurve is performed on the object using a corresponding mapping function in order to associate with each point of the plurality of points a corresponding index of the at least one SFC metacurve. In one or more embodiments in which the mesh comprises a plurality of cells, a mapping of the plurality of SFC metacurves is performed on the object using a corresponding mapping function in order to associate with each point of the plurality of points a corresponding index in the plurality of SFC metacurves.

It will be appreciated that in one or more embodiments, the method further comprises subdividing at least one given cell of the mesh to obtain at least two sub-cells within the given cell; wherein the chaining traversing the given cell traverses each of the at least two sub-cells, the chaining comprising a symbol for
5 each of the at least two sub-cells and wherein the SFC metacurve associated with the given cell comprises at least two SFC metacurves, each of the at least two SFC metacurves being associated with a corresponding sub-cell of the given cell.

While this has not been disclosed in Fig. 52, in one or more embodiments the computer-implemented method further comprises providing an indication of the
10 mapping.

It will be appreciated that the indication of the mapping may be of various types as will appreciate the skilled addressee.

It will be further appreciated that the indication of the mapping is provided by the computer system implementing the method in one or more
15 embodiments. Moreover, it will be appreciated that the indication of the mapping may be provided according to various embodiments. In one or more embodiments, the indication of the mapping is saved in a file by the computer system. The skilled addressee will appreciate that the file may be of various formats.

In one or more alternative embodiments, the indication of the mapping is
20 displayed on the display of the computer system.

In one or more alternative embodiments, the indication of the mapping is provided to a remote processing device operatively connected to the computer system. It will be appreciated that the remote processing device may be operatively connected to the computer system according to various embodiments. In one or
25 more embodiments, the computer system is operatively connected to the remote processing unit at least one of a local area network (LAN), a metropolitan area network (MAN) and a wide area network (WAN). In one or more embodiments, the wide area network (WAN) comprises the Internet.

Referring to Figure 53, there is shown a computer system 1000 adapted to implement the present technology. The computer system 1000 comprises various hardware components including one or more single-core or multi-core processors collectively represented by processor 1002, a graphics processing unit (GPU) 1004, a storage disk such as an SSD 1006, a random access memory 1008, a display interface 1010, and an input/output interface 1012.

Communication between the various components of the computer system 1000 is ensured by means of one or more internal and/or external bus 1014 (e.g. a PCI bus, a universal serial bus, an IEEE 1394 "Firewire" bus, a SCSI bus, a Serial-ATA bus, etc.), via which the various hardware components are electronically coupled.

The input/output interface 1012 may be coupled to a touch screen 1016 and/or to the internal and/or external bus(es) 1014.

The touch screen 1016 may be part of the display. In one or more embodiments, the touch screen 1016 serves as the display.

The touch screen 1016 may also be referred to as display 1016. In the embodiments illustrated in Figure 53, the touchscreen 1016 comprises touch hardware 1018 and an input/output controller 1020 enabling a communication with the display interface 1010 and/or the external bus(es) 1014. In one or more embodiments, the input/output interface 1012 may be connected to a keyboard (not shown), mouse (not shown) or trackpad (not shown) enabling the user to interact with the computer system 1000 in addition to or in place of the touch screen 1016.

Depending on the implementations of the present technology, the SSD drive 1006 stores program instructions that can be loaded into the random access memory 1008 and executed by the processor 1002 and/or the GPU 1004 in accordance with the process embodiments presented herein. For example, the program instructions may be part of a library or an application.

It will be appreciated by the person skilled in the art that the computer system 1000 may be implemented as a server, a desktop computer, a laptop computer, a tablet computer, a smart phone, a personal digital assistant or any other device suitable for implementing the present technology as will be appreciated by the person skilled in the art.

Thus, it will be appreciated that there is disclosed a computing device comprising at least one processor; a display device; an input-output interface; a memory and a bus for coupling the at least one processor, the display device, the input-output interface and the memory.

- 10 It will be appreciated that the memory comprises instructions which when executed by the at least one processor perform a method for indexing an n-dimensional object comprising a plurality of points. The method comprises obtaining via the output input interface a mesh of the n-dimensional shape of the object, the mesh comprising at least one cell. The method further comprises
- 15 determining by means of the at least one processor a chaining traversing the at least one cell of the mesh; the chaining comprising for each cell a symbol traversing the cell and joining two vertices of the cell. The method further comprises, for each cell of the mesh and by means of the at least one processor: associating an SFC metacurve with each symbol associated with two vertices.
- 20 Finally, the method comprises performing, by means of the at least one processor, a mapping of the at least one SFC metacurve onto the object using a corresponding mapping function in order to associate with each point of the plurality of points a corresponding index of the at least one SFC metacurve.

- It will be appreciated that there is also disclosed a computer-readable medium for
- 25 storing statements and instructions for execution by a computer, the statements and instructions comprising encoding means for obtaining a mesh of the n-dimensional shape of the object, the mesh comprising at least one cell. The statements and instructions further comprise encoding means for determining a chaining traversing the at least one cell of the mesh; the chaining comprising for
- 30 each cell a symbol traversing the cell and joining two vertices of the cell. The statements and instructions further comprise a coding means for, for each cell of

the mesh, associating an SFC metacurve with each symbol associated with two vertices. The statements and instructions further comprise coding means for mapping the at least one SFC metacurve onto the object using a corresponding mapping function in order to associate with each point of the plurality of points a
5 corresponding index of the at least one SFC metacurve.

It will be appreciated that there is further disclosed a computer program comprising computer-executable instructions which, when executed by a computer, perform one or more embodiments of the disclosed method.

DEFORMABLE SFC METACURVES

10 ***A universal deformation pipeline***

Principle of adaptability

It will be appreciated that the idea of deforming SFCs stems from the need to adapt them to the context and geometry of the data. As the use of SFCs is extremely wide-ranging, the principle of adaptability must be able to be extended
15 to SFC metacurves traversing any mesh and Euclidean or non-Euclidean support grids in a variety of application domains. The principle of adaptability of SFCs to the application context, which for various reasons has been limited to the use of historical SFCs, in particular Hilbert curves, needs to be reviewed and extended to general metacurves.

20 **Principle of universality**

It will also be appreciated that the main feature of the SFCs, i.e. the ability to index n-dimensional parameter spaces in one dimension, is the basic principle of a universal deformation pipeline. This universal deformation pipeline makes it possible to index any object in any dimension from any composition of SFC
25 metacurves, and from any mapping function by taking any path on the object through an appropriate ordering of the metacurves.

Functional diagram

The functional diagram of the universal deformation pipeline shown in **Fig. 1** breaks down into three main phases. It will be appreciated that during these main phases certain algorithmic choices determine the deformation methods used and their suitability for indexing the deformed SFCs.

The initial phase P1, P2 is the meshing phase, which will determine the partitioning by regions of the surface or object that will subsequently be traversed by the SFC metacurves. The elementary regions of the mesh are the cells, which can be subdivided into sub-cells. The second phase, P3, is the chaining phase, which determines the order in which the SFC metacurves will traverse the objects they index. The third phase, P4, P5, P6, is the phase that determines the families of SFC metacurves chosen or created specifically to traverse the meshes. The last phase, P7, P8, P9, is the mapping phase, which maps the points of the two-dimensional metacurves to points on the n-dimensional target objects.

The mesh

It will be appreciated that meshes are essentially graphs with vertices, edges, faces, hyperfaces or volumes and so on up to n dimensions. They are made up of cells, for example quadrilaterals and triangles in two dimensions. If we're interested in using SFC metacurves to index meshes in an optimal or context-sensitive way, we need to study the possible paths that these metacurves can take to visit the set of cells with special topological graphs and maps. In all cases, it will be appreciated that the cells will only be visited once, passing through the vertices only.

Structural and non-structural meshes

The deformation pipeline begins with step P1, which determines the nature of the mesh on which the SFC will be applied. There are two types of mesh: structured and unstructured. To simplify matters, structured meshes, unlike unstructured meshes, allow for simple parameterizations. The most common surface meshes are native meshes based on triangles and quadrilaterals. These are the types of

mesh that will be used in all the examples presented below. Hexagonal meshes in particular will be treated as meshes consisting solely of quadrilaterals: dissection of the hexagon into two trapezoids, or as meshes consisting of triangles and quadrilaterals: dissection of the hexagon into one quadrilateral and two triangles.

Eulerian and Lagrangian deformations

Eulerian and Lagrangian deformation methods refer to deformation methods for which the mesh grids are essentially regular and fixed in the first case, and irregular and mobile in the second. Step P2 in the meshing phase of the deformation pipeline therefore aims to introduce deformations managed by physics and not just by purely geometric methods. The principle of potential grid deformation forms the basis of methods for adapting SFCs to physical or geometric contextual data.

Chaining and traversability-ordering issues

It will be appreciated that chaining methods aim to assemble SFC metacurves together from other metacurves or simple SFCs. In all cases, chaining strategies aim to enable bijective indexing of chained curves and metacurves, whatever the chosen dimension. Once chained, the metacurves will visit the cells in a ordered specific sequence. A traversable sequence is a spatial path without going over any cell twice. Different chaining traversability-ordering strategies exist, mainly aimed at compressing geometry, organizing cell's information according to neighborhood ("data locality") or indexing cells to enable real-time queries on the information associated with them. The complexity of traversability-ordering obviously depends on the type of mesh, which may or may not be structured. The main techniques used, for example, seek to isolate "ribbon" series of contiguous triangles in triangular meshes. They can also search for Hamiltonian paths in unstructured meshes.

SFC metacurves

Combinatorial and fractal SFCs

Combinatorial SFCs are historically a limited number of pairing functions, including the Hilbert curve. But geometrically and graphically, SFCs can also be
5 generated and described in a similar way to fractal curves, using grammars or generative languages. They then lose their on-the-fly pairing capabilities (bijective encoding and decoding of indices and point coordinates). Finally, it is possible to extend SFC theory to heuristic curves such as Hamiltonian paths that cover all points on a given grid.

10 **Dimensions and algorithms**

It will be appreciated that the characteristic of SFCs of being able to index n-dimensional parametric spaces in one dimension means that algorithms operating in n-dimensions can be directly adopted at the level of meshes, metacurve generation functions, associated pairing functions and mapping
15 functions. However, for the sake of convenience and performance, these algorithms can be specialized for optimal operation in two and three dimensions.

SFC and SFC metacurves

Gray's metacurve system was designed to describe end-to-end assemblies of SFC curves belonging to new families of pairing functions. These user-
20 configurable assemblies or linkages enable SFCs to be adapted to any rectilinear grids and to index non-regular spaces. These metacurves are themselves successions of Gray curves or metacurves whose spatial information is supplied and which are associated with automatically generated pairing functions.

Mapping

25 It will be appreciated that the mapping is the operation of matching the points of the SFC metacurves to the surfaces or objects to which they are applied to.

Types of mapping

The types of mapping encountered when applying SFCs and SFC metacurves are varied and borrowed from existing work mainly in pure mathematics, cartography and applied mathematics in the field of computer-aided design and manufacturing. Most of these types are referenced in the bibliographic notes (Sec. 6).

Surface types

The types of surfaces studied are first and foremost mathematical surfaces formalized in parametric form. These surfaces can admit several different parameterizations based on trigonometric functions or polynomial functions. Next, mapping onto surfaces or free-form shapes from computer-aided design and manufacturing systems is disclosed. Finally, the mapping of SFC metacurves onto polyhedral surfaces is disclosed. In all cases, n-dimensional approaches are studied.

Vector and pixel deformations

It will be appreciated that vector and pixel deformations are of different algorithmic natures and each has its own uses. Generally speaking, vector deformations are mainly used for their bijective indexing and cryptographic capabilities, whereas pixel deformations are more closely linked to visual identification and authentication. Furthermore, the pixel representation of SFC metacurves requires the use of polygon-filling algorithms.

An example of the deformation process

A detailed example of the SFC metacurve deformation process illustrates the operation of the deformation pipeline disclosed above. The example shown in **Fig. 2** comprises eight successive phases. The first phase {S0} consists in setting up the topological polygonal diagram of the mesh. This phase may be optional, depending on the complexity and nature of the meshes to be created. It is defined in section Ref. 2.6.

The second phase {S1} generates an Euclidean mesh in a given dimension, associated with a given coordinate system and a given grid. The third phase {S2} exploits pre-existing meshes. As for the subsequent phases, the {S3} phase consists in establishing Peano maps of the previously defined mesh. These
5 Peano maps are characterized by associated graphs and constitute the ordered paths of meshes that can be meshed at a later time. The {S4} phase consists in generating grids and maps for each cell (polygon), and the {S5} phase iterates the operation over all the cells to generate the symbolic metacurve associated with the general mesh. Phase {S6} consists in generating the real 2-dimensional
10 metacurves. Phase {S7} consists in transforming the Euclidean coordinates of the metacurves into curvilinear coordinates in the given dimension of the Euclidean grid.

Deformation interface

The deformation system interface focuses on the control of the deformation using
15 control points. The use of graphical procedures, such as naval architecture's use of wooden laths and lead pins (the control points), enable to draw complex curves without the need for equations. Avoiding the manipulation of complex mathematical deformation systems is desirable in a wide range of applications. The strategy for controlling the surfaces without using their equations - the historic
20 strategy of computer-aided design and manufacturing systems - is favored in the work presented below. Classical parametric surface interfaces based on surface patches and control points will be reformulated to apply SFC metacurves to these surfaces, and hyperbolic paraboloid control points will be used to apply these metacurves to triangles or quadrilaterals in 3 dimensions.

25 CHAINING AND PEANO GRAPHS

Definition

Planar Peano graphs, in reference to Peano's first diagram, are defined as a succession of arcs visiting the faces of the graph via the vertices. These graphs are associated with polygons limited to quadrilaterals or triangles in the examples

shown. Polygons with a number of sides > 4 will be dissected, if necessary, into quadrilaterals and triangles. The work will then be generalized to polyhedral or polytopic graphs. The properties of these graphs are fundamental for generally optimizing the indexing of mesh polygons based on paths or cycles visiting them.

- 5 From a robotic point of view, the sequence of the polygons visited enables the trajectories and paths travelled to visit the set of polygons to be minimized, and memory to be optimized for the organization of finite element-type processing. Symbolic Peano maps are defined as topological maps made up of arcs waiting to be defined by final SFC metacurves.

10 ***Chaining's Typology***

- The typology of Peano chainings is important, as it enables to draw on pre-existing linkage typologies during the meshing phase, or to select the right meshing algorithms separately. A distinction is obviously made between structured and unstructured meshes, and in both cases between deterministic and heuristic meshes. Figure Fig. 3 illustrates two Peano paths, the first with a U-shaped topology: the start and end points are adjacent, and the second with a W-shaped topology, these same points being diagonally opposed. In both cases, the paths visit the 4 polygons once and only once. Moreover, the vertices that are visited are visited only once. The same figure below illustrates, on the left, a configuration with single points visited once and, on the right, a cyclic configuration with multiple points visited more than once.
- 15
20

Structured chaining

Procedural structured chaining

- Whenever possible, a procedural topological map can be used to develop combinatorially formulated Peano chains, depending on the desired resolution of the metacurve and, in this case, its parity. The procedural topological map shown in **Fig. 4** is built by isolating 4 zones, two triangular ones plus the first horizontal line and the last vertical column.
- 25

The result is the generation of even metacurves of resolution 2α and odd metacurves of resolution $2\alpha + 1$ with $\alpha = 1..4$.

Enumerative generation of chaining

When a combinatorial algorithm for generating metacurves is not available, it is possible to resort to metacurve occurrences obtained by enumerating the set of Peano paths covering a given resolution grid. These enumerations can be performed using brute-force algorithms, or manually for low resolutions. Some enumerations for 5x5 and 6x6 square grids are illustrated in **Fig. 6** and **Fig. 7**.

Chaining with rectangular friezes

Structured chaining of SFC metacurves covering rectangular regions can be based on certain Gray's lace metacurves Ref. [109] or on $(m.a)(n.a)$ juxtapositions of square metacurves of resolution a as illustrated in **Fig. 9**. In this illustration, the friezes are made from 4x4W metacurves illustrated in **Fig. 6**. The two friezes shown are made without the need for multiple vertices.

Furthermore, in Sec. 2.6, it will be illustrated that they are directly adaptable to topological polygon meshing techniques. The rectangular friezes shown in **Fig. 10** have the particularity of being cyclic and are obtained by connecting two identical metacurves together, using the center of the rectangles as the center of symmetry.

Square cyclic chaining

Procedural constructions of square cyclic chaining are easily made from the central symmetry of a square for example from rotational symmetry from SFC metacurves rotated by $\alpha\pi/2$ with $\alpha = 0..3$. To produce the 8x8 cycles in **Fig. 11**, 4x4 W-chains borrowed from **Fig. 6** are used. To realize the 12x12 cycle of **Fig. 12** a 6x6 W-chaining is used.

Peano unstructured chainings

The search for Peano chainings is equivalent to the search for Hamiltonian paths or cycles on the graph of vertex-face adjacencies associated with the Peano graph. To create such a graph, as illustrated in figure **Fig. 13**, we need to
 5 triangulate each convex polygon of the mesh from a point inside the polygon and remove the edges of the Peano graph and its bivalent vertices. As a result, the complexity of searching for paths or circuits in a Peano graph is equivalent to that of the general search for Hamiltonian paths or cycles, i.e. NP-complete.

Rectangular SFC metacurves

10 Filling curves are historically and for certain theoretical reasons square-shaped, as is the case for Hilbert curves. The extension to rectangular curves is relatively recent, with the aim of better adapting curves to the environment they traverse, for example, to scan images.

Peano maps and Hilbert metacurves

15 Gray's metacurves have enabled in particular Bib. [113] to generalize Hilbert curves and their generation modes. Generalization to rectangular grids in the form of Hilbert strips presents no difficulties. Figure **Fig. 15** illustrates the replacement of Peano arcs by Hilbert curves of different orders. The dimensions of the rectangular grid used are $2m(2n + 1)$, as illustrated in figure **Fig. 16**.

Band Hilbert metacurves

Figure **Fig. 17** illustrates two Hilbert SFC metacurves. Other SFC metacurves can be used to traverse rectangular grids, but the locality properties of Hilbert curves are interesting to exploit for traversing certain types of surfaces, such as generalized cylinders and special cases of surfaces of revolution.

Meshing with polygonal diagrams

25 Polygonal diagrams or fundamental polygons are important topological tools for studying surfaces and their classification. They enable to move from dimension 2

to dimension 3, making them a perfect tool for applying two-dimensional SFC metacurves to surfaces. The study of structured or unstructured meshes covered by SFC metacurves can be studied before the mapping. This approach is particularly effective for constructing graphs and topological maps of Peano
5 associated with polygons, e.g. rectangular polygons.

Meshing generalized cylinders

It will be appreciated that generalized cylinders are preferred geometric descriptors in pattern recognition from discrete data, but also in free-form design. One of the advantages of generalized cylinders is their ability to model a large
10 number of classical surface families, such as surfaces of revolution. Moreover, the generalized cylinder mesh is topologically adapted to the use of grids and rectangular SFC metacurves. The block diagram in figure **Fig.** 18 describes the process of applying SFC metacurves to generalized cylinders. The first step is the selection (C0) of a toric topological polygonal diagram, followed by the generation
15 of a rectangular structured mesh (C1) to which a Peano map (C2) is associated. This map (C3) establishes the order of arcs visiting the polygons of the mesh. This map is the symbolic Peano map. Next (C4), an SFC metacurve is generated for each arc in the map. The final result (C4), a composite SFC metacurve, is finally applied to a generalized cylinder (C6). Steps C0, - - - , C5 are two-
20 dimensional, while step C6 is three-dimensional.

Example of revolution surfaces

The example described below shows the creation of a Peano map for traversing a generalized cylinder of the surface-of-revolution type with Hilbert-type SFC metacurve bands. Surfaces of revolution are generally described using
25 coordinate grids made up of meridians and parallels. Spherical, ellipsoidal and cylindrical grids are special cases. In these cases, it is often convenient to use rectangular grids rather than square ones.

By mapping the points on the given surface of revolution to the points on the striped Hilbert curve, we obtain an SFC path of the surface of revolution, as shown in figure **Fig. 19**.

PARAMETRIC SURFACES

- 5 In general, the vector-type mapping of SFC metacurves uses a point-to-point correspondence, whereas the pixel-type mapping uses a point to non-flat quadrilateral correspondence, each quadrilateral being graphically represented by two triangles.

Mapping principles

- 10 The principles of applying SFCs to surfaces are generally straightforward. An example is given of the mapping of a double-spiral SFC curve (Sec. 6) to different parameterized versions of the sphere (Fig. 20). The first three mappings are classical, using trigonometric functions. The fourth mapping at bottom right is experimental and polynomial in nature. The encoding and decoding functions of
- 15 the mappings are listed in figures **Fig. 40, 41, 42, 43, 44**. In all cases, a distinction is made between mappings without combinatorial bijective pairing and mappings preserving the bijective pairing of SFC metacurves before mapping to a surface. This distinction is important when we need to find the index of a point on a metacurve after an initial inverse mapping operation, then coding the metacurve
- 20 to provide an index from coordinates.

Some image examples

- The 6 series of surface illustrations presented here are based on direct correspondence between points on surfaces known by their parametric equations and points on SFC metacurves applied to these surfaces. The main drawback of
- 25 such an approach is the problem of non-uniformity in the distribution of points on surfaces, with sometimes significant distortions of the SFC metacurves. To overcome this problem, algorithms for conformal mappings or regular point distribution need to be integrated into the matching process. Figure **Fig. 21** illustrates the deformation of a supertoroid.

Figure **Fig. 22** shows a Dupin cyclide at the top and a Klein-Jeener surface at the bottom. Cyclide distortions are minimized when the cyclide is a perfect torus. For the Klein-Jeener bottle, metacurve distortions become difficult to control without modifying the parametric equation. As far as figure **Fig. 23** is concerned, the SFC metacurves will have minimal deformations when the hyperbolic paraboloid at the top is equivalent to a plane square, but the Möbius strip at the bottom of the figure will have the same distortion problems as the Klein-Jeener bottle.

PARAMETRIC SURFACES

This paragraph describes an approach based on geometric deformations intrinsic to the theory of parametric surfaces developed in the world of computer-aided design and manufacturing. The first advantage of this approach is to reformulate existing surface generation algorithms, enabling direct adaptation to the functionalities of current CAD/CAM software and existing digital files. The second advantage is that it enables one-dimensional indexing of parametric hypersurfaces, making it possible to introduce deformable SFCs in mesh theory in any dimension.

The block diagram of the system for two- and three-dimensional mappings is shown in figure **Fig. 24**. The system is divided into 14 phases and applies to files supplied as parametric patches or polygonal mesh files. The transition to higher two-dimensions is achieved by replacing control patches with hyperpatches and polygonal meshes with meshes made up of, for example, hyper-tetrahedra and hypercubes. The system can be broken down into several subsystems. Subsystem {P5, P6, P7, P8} corresponds to a conventional system for generating parametric surfaces of the Bezier, B-splines, Nurbs or other type, operating in control patch or interpolation grid mode. The {P0, P4} subsystem corresponds to the creation of parametric patches from polygonal meshes. The {P4} case mainly concerns the generation of planar PSFCs applied to polyhedral faces or subdivision surfaces. In this case, the points of passage can typically be the points of a bilinear grid associated with each face and therefore coplanar with the face. The {P0} case corresponds to methods for smoothing polyhedra using parametric surfaces. The subsystem {P1, P2, P3} corresponds to the topological ordering of

a model's patches, in order to identify whether any Peano graphs exist on the model. The search for these graphs may be subject to constraints, such as the search for paths from a given point to another point, or for cycles. Case {P2} corresponds to the creation of one or more topological maps of the model. Case {P3} corresponds to the determination of the orientation and direction of travel of each PSFC. Subsystem {P9, P10, P11} corresponds to the creation of Gray metacurves or non-Gray metacurves used to generate PSFCs. The set of patches for a 3D model or object will be generated in step {P12}. The mapping of distorted and colorized SFC metacurves at pixel level (case {P13}) is the last subsystem.

10 **Presentation of PSFC**

Parametric surfaces and hypersurfaces

Bases and Degrees

The indexing of SFC metacurves extends to curvilinear parametric hyper-spaces and, in particular, parametric hyper-patches. These are generally used to represent free-form shapes in CAD/CAM systems. The link between the various types of SFC metacurve indexing and the analytical representation of hyperpatches is made directly by mapping the tuple of an n-SFC point to the tuple of coordinates of the hyperpatch point in its parametric space.

Hypercurves and Bezier metacurves

Bezier curves and surfaces were among the first parametric surface models to be developed and have many qualities that have made them popular in a wide range of free-form geometric modeling applications. Their mathematical formalism lends itself perfectly well to junction with SFC metacurve theory in spaces of any dimension. The algorithmic calculation of points belonging to hypersurfaces is straightforward and is formulated by equation **Equ. 1**. Although the proposed system for applying SFC metacurves to hypersurfaces works whatever the type of parametric surface chosen, the simplicity of the Bezier formalism makes it possible to illustrate n-dimensional correspondences between hypersurfaces and n-dimensional metacurves.

$$S(u, v, w) = \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n B_i^l(u) B_j^m(v) B_k^n(w) P_{ijk}$$

(Equ. 1)

where P_{ijk} are the control points of the 3-dimensional hyperpatches and B_i^n ($i = 0..n$) are the Bernstein-based functions defined by $B_i^n(t) = \frac{n!}{i!(n-i)!} (i-t)^n t^i$.

- 5 From equation **Equ. 1**, it is possible to obtain in **Equ. 2** the formulation, from the index of point I , of the coordinates of a point $x_{0,1,2}$ on a three-dimensional DC3 metacurve of resolution a applied to the Bezier volume. Parameter b is used to control the scale factor of the metacurve in the patch it visits. Using $b < a$ avoids overlaps between different parts of the metacurve for neighboring patches.

$$10 \quad \mathfrak{S}(\mathfrak{I}) = \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n B_i^l(u) B_j^m(v) B_k^n(w) P_{ijk}$$

$$\text{where } u, v, w = \frac{a(a-b) + 2bx_{0,1,2}}{2a^2}$$

$$\text{et } x_{0,1,2} \in {}^S_a DC^3(I)$$

(Equ. 2)

- 15 Figure **Fig. 26** illustrates the deformation of a Gray metacurve into a Bezier volume.

The Bezier hyperpatch has 64 control points.

Bicubic surfaces in matrix form

- 20 Based on predefined Bernstein functions, it is also possible to use a matrix formalism for bicubic surfaces. The following formulas for calculating a point on the surface are obtained for the general parametric representation of a bicubic surface:

$$x(u, v) = U * M * G_x * M^t * V^t$$

$$y(u, v) = U * M * G_y * M^t * V^t$$

$$z(u, v) = U * M * G_z * M^t * V^t$$

(Equ. 3)

- 5 with $U = [u^3, u^2, u^1, 1]$, V^t , the transposed matrix of $V = [v^3, v^2, v^1, 1]$, M the 4x4 matrix representing the chosen polynomial basis (e.g. Bezier), M^t its transpose, G_x , G_y , G_z the 4x4 matrices formed from the x, y and z coordinates of the patch control points. Figure **Fig. 26** illustrates the indexing of a Bezier patch by an mC² curve, while figure **Fig. 27** illustrates the indexing of the Utah teapot modeled from
10 bicubic Bezier surfaces by mC² curves.

PSFCs: The junction between Parametric Surfaces (PSs) and Space Filling Curves (SFCs)

- PSFCs use the UV grids of the parametric surfaces as a support. The algorithm for calculating points belonging to the UV grid Alg. 10 is modified to calculate SFC
15 points applied to the grids. The resolution of the SFCs is given in rational form $\{a, b\}$ with $a \leq 1$. This formalism makes it possible to fix a percentage of SFC occupancy in its patch and therefore to fix the distance between adjacent PSFCs. Figure **Fig. 27** illustrates a PSFC version of Newell's teapot and figure **Fig. 28** a PSFC version of Catmull's elephant. In both cases, the topological models are
20 multiply-connected. The main parts of the teapot are thus independent and unconnected.

- The illustration of the teapot on the left shows the PSFC and the triangular mesh of the teapot. The illustration on the right shows the body of the teapot and the Peano path of the associated patches. In the case of the elephant, the figure on
25 the left illustrates the first phase in calculating the PSFCs, i.e. replacing the u, v parametrization. The second phase, illustrated by the figure on the right, consists in choosing the neighborhood distance between adjacent PSFCs. The third phase, which is also the most complex, consists in determining one of the four possible

orientations of the PSFCs in their patch and one of the two directions of travel of the SFCs, i.e. the beginning and end of the PSFC in its patch.

PSFC formalism

The final formalism of the PSFCs associated with a mesh model will therefore be made up of lists describing the different Peano paths associated with a model. These lists will be constituted from the original indices of the patches as geometrically designed. Each list constituting a path will be associated with the list of PSFC orientations in their patch. These lists, coded in quaternary form, can be replaced by a decimal number. These same lists will be associated with lists of the same length encoding the directions of the SFCs in binary form. These in turn can be replaced by a decimal number. The memory storage structure for all PSFCs associated with a mesh model is illustrated in **Fig. 29**. Typical data structures for parametric patches are shown in the inset at the bottom left of the figure.

PSFC defined by interpolation

The interpolation mechanism for parametric surfaces defined by equation **Equ. 3** consists in determining the control polyhedron of a patch from points belonging to the surface. For a bicubic patch, 16 points in the general configuration will provide the coordinates of the 16 control points. These points are calculated using the figures **Fig. 13** and **Fig. 14** functions. The advantage of using interpolation is twofold: not only can any PSFC be applied to curvilinear surfaces discretized by a mesh of points or triangles, for example, but interpolation can also be used to control the points at which the PSFC passes over the surface in question.

Trajectory optimization

The interpolation of SFC metacurves passing through a number of crossing points not only offers an interesting method of spatially deforming SFCs, but also enables to solve certain trajectory optimization problems for covering regions. In this case, the general deformation of an SFC metacurve follows the principle illustrated in figure **Fig. 30**. Starting from n^2 coplanar or non-coplanar crossing

points, the control polyhedron of the associated parametric patch of degree n is determined. This determines a double parametrization in u and v and subsequently enables us to map a previously selected SFC metacurve to the mesh in u and v .

5 ***PSFC by example***

PSFC versions of two 3D models based on Bezier patches, which have become part of the history of computer graphics are presented. These are Catmull's Gumbo elephant and Newell's teapot. Two main approaches can be used to obtain a PSFC version of the model. The first is topological and semi-manual, while the second is combinatorial and automatic, but based on a naive brute-force algorithm. Indexing these objects using the topological method involves the following steps. The first step is to determine the possible Peano graphs on the models in question. To do this, a topological map of the models, its Schlegel diagram, has to be created first.

15 **Creating a topological map**

Model parametric patches are generally described sequentially in digital exchange formats. They allow to visualize UV curves or associated triangles without worrying about any order between them or neighborhood problems. Patches can therefore be processed independently of each other, and possibly in parallel. In the context of mesh simulation or object manufacturing, on the other hand, the problem of neighboring patches is fundamental.

Peano graph search

The theory of Peano graphs is introduced in Sec. 2. The search for these graphs in different mesh types enables to order the sequence of surface patches in digital exchange formats. This traversability-ordering determines not only the topological relationships between patches, but also the optimization of mesh calculation processing and cell organization. Figure **Fig. 32** illustrates a simple Peano cycle for the PSFC traversing the patches in one ear of Catmull's elephant model. In the figure on the left, the PSFC is a double-spiral SFC. Top right shows the

topological map of the ear, with the Peano circuit created using arcs of circles. Bottom right is the graph corresponding to the topological map in the form of vertices and edges.

Heterogeneous squares of classes U, W and Z

- 5 In line with the realization of Gray or non-Gray metacurves, it is possible to define PSFCs of hybrid typologies mixed in U and W or purely U or W. Figure **Fig. 33** illustrates a purely W PSFC mesh from Catmull's model before the PSFC orientation and direction phase. In general, the simultaneous use of U and W PSFCs simplifies the search for Peano graphs. The use of Z-shaped PSFCs is of particular interest in cryptography.
- 10

POLYHEDRAL SURFACES

- Polyhedral surfaces, that are the set of polygonal faces of polyhedral, can be visited by SFC metacurves according to previously established Peano paths. The two main problems in covering a polyhedral surface are therefore the construction and possible deformation of the SFC metacurves in three-dimensional plane polygons, and the determination of the Peano chain and graph associated with the surface. The problem of covering polyhedral surfaces with SFC metacurves can be considered as a special case of parametric surfaces whose surface patches have all their control points coplanar (Bib..[61]). This is an interesting strategy to use if we wish to cover the polyhedral surface with deformed SFC metacurves.
- 15
- 20

- On the left, figure **Fig. 34** shows the Schlegel diagram of a cube whose faces are covered and visited only once by SFC metacurves. The chaining and associated Peano graph is a simple Peano cycle, with some vertices not visited. Each arc of the graph is then associated with a set of SFC metacurves covering the faces of the cube. The right-hand side of the figure shows an inside view of the cube and the resulting chaining.
- 25

Polygons and SFC metacurves

Two different approaches are used to deform SFCs in the plane. The first is to apply SFCs to non-flat or plane quadrangles, based on the parametrization of the hyperbolic paraboloid. The triangles can then be decomposed into quadrangles.

- 5 The second approach uses PSFCs with coplanar control points. The resulting PSFCs are flat and deformed in the plane of the control points. Figure **Fig. 35** shows on its left a PH traversed by a meandering SFC and on its right a triangle in space previously split into 3 quadrilaterals and traversed by 3 meandering SFCs of different resolutions.

Deformations in the plane

- Deformations of SFC metacurves from 3-dimensional control points can be adapted to structured or unstructured planar meshes in three-dimensional space. In fact, SFC deformations in any plane of space can be obtained directly by introducing coplanar control points. This property makes it possible to traverse
15 meshed plane regions from SFCs, but the principle is easily extended to higher spaces. The control points of four-dimensional hyperpatches contained in a three-dimensional subspace can therefore be used to obtain an SFC filling this subspace.

- Two complementary approaches are disclosed. One is based on control points in
20 the space of the hyperbolic paraboloid, the other one on control points of parametric surface patches. In both cases, the coplanarity of the control points in a plane ensures deformations in the plane in question.

Hyperbolic paraboloid control points

- The hyperbolic paraboloid mesh can be used to apply any SFC metacurve to a
25 non-flat or plane quadrilateral when the 4 vertices of the quadrilateral are coplanar. Equation **Equ. 4** provides the parametric expression for the point Q linearly dependent on the four points A, B, C, D.

$$Q = \frac{\rho (\alpha A + \beta B)}{\alpha +} + \frac{\sigma (\gamma C + \delta (D))}{\gamma +} (\rho + \sigma)^{-1} \quad (4)$$

The homogeneous weighting coefficients α , β , γ , δ , ρ , σ are transformed by successively posing $\beta = \alpha - 1$, $\delta = \gamma - 1$, $\sigma = \rho - 1$, $\gamma = \alpha$ then finally $\alpha = u$ and $\rho = v$ to obtain equation Ref.5 which expresses the point of travel of the quadrilateral Quv as a function of the parameters u and v.

$$Q_{uv} = vuA - v(u-1)B - u(v-1)D + (u-1)(v-1)C \quad (5)$$

Figure **Fig. 23** illustrates a general hyperbolic paraboloid in the space traversed by an SFC metacurve. Figure **Fig. 35** illustrates a plane PH on the left, and the dissection of a triangle into three planes PHs on the right, each traversed by a different SFC metacurve.

Control points and parametric surfaces

On the left, figure **Fig. 36** illustrates an SFC in the U-type plane, with the 16 control points used to deform it bicubically. Deformations of higher degree n are obtained by increasing the number of control points to n^2 and adapting the matrix formalism of equation **Equ. 3**. The figure on the right shows the folding of a W-type SFC in three-dimensional space.

Planar tessellations of SFC metacurves

Plane tessellations of SFC metacurves can be obtained by tessellating the meshes obtained from their control points. Figure **Fig. 37** illustrates a planar tessellation of deformed SFCs by tessellating their coplanar control points.

APPLICATIONS

It will be appreciated that the disclosed method for indexing an n-dimensional object is of great interest for many applications. In particular, the method can be advantageously used in any of the following applications:

1. Indexing points on a surface for physical objects and digital objects.
2. Indexing points on a volume in medical imaging to enable haptic control of soft and deformable objects and adaptive classification in MRI tanks.
3. Indexing points on a volume in additive manufacturing for filling and slicing in 3D printing.
4. Indexing of spatial databases for topological and geometric data.
5. Indexing of graphics databases for texture maps, SFC pixel display, polygon mesh compression.
6. Indexing and optimization of trajectories in mobile robotics for laser nozzles, drones (military, LED).
7. Mesh optimization in nD space in FEM for Lagrangian and HamStrand systems.
8. Target acquisition (indexing) for wireless energy transmission (laser, sun) and wire-guidance.
9. Geometric indexing and compression for photogrammetric and LIDAR data.
10. AI pattern recognition with neural SFC generation and 3D neural networks.
11. Image analysis with image scans and biometric scans.
12. Communication networks with network indexing and routing optimization.
13. Genomics with human genome unfolding coding and genome indexing and volume deformation.
14. Electronic design, with the design of deformable SFC antennas and deformed SFC circuits.

It will be appreciated by the person skilled in the art that the disclosed method offers a technical solution to a technological problem. Indeed, by proposing a method of indexing an n-dimensional object comprising a plurality of discrete points, it is possible to obtain optimization of the resources required to access
5 each of the discrete points of any n-dimensional object. Anyone familiar with the subject will appreciate that identifying discrete points in an n-dimensional object requires significant resources, both in terms of storage and information access. Using an index to identify a given discrete point instead of the coordinates of the discrete point therefore optimizes the resources of the associated device. Not
10 only memory, but also the associated processor(s) will be appreciated.

BIBLIOGRAPHICAL NOTES

Notes on SFCs

New families of SFCs isolated recently are described in the following series of works: **Bib.** [106, 112, 113, 109, 110, 107, 108, 111], then generalized at the level
15 of a composition and assembly system called Gray's metacurves **Bib.** [114]. The chaining of SFC metacurves is introduced in **Bib.** [115]. The principle of composing SFC functions is also described in **Bib.** [92]. Recursive descriptions of SFCs can be related to fractal methods. They are introduced in **Bib.** [90] or **Bib.** [95]. The work **Bib.** [18] is a comprehensive introduction to a fractal approach to
20 SFCs and their construction using a generative language. The construction of non-combinatorial SFCs using artificial intelligence techniques is described in **Bib.** [117].

Notes on adaptable SFCs

Classically, the adaptability of SFCs to given environments is achieved using
25 Euclidean grids **Bib.** [30], **Bib.** [39]. Gray's metacurves, introduced in **Bib.** [114] allow adaptation to rectilinear polygonal regions while retaining bijective pairing capabilities. **Bib.** [100].

Notes on deformable SFCs

The use and study of regular grid transformations is not new **Bib.** [1], **Bib.** [3] and are classified into projective transformations, anamorphic transformations, cartographic transformations and Lagrange transformations, among others. The latter are well suited to the deformation of regions or grids and are particularly used in mesh theory **Bib.** [12]. The strategy of favoring the deformation of grids: Lagrangian approach over the Eulerian approach of fixed grids in finite element theory is very similar to the strategy of deforming SFC grids meshing the regions they cover. The following references **Bib.** [99], **Bib.** [97], **Bib.** [96], **Bib.** [35] introduce the differences between Lagrangian and Eulerian meshes. Reference **Bib.** [35] is an introduction to the design of physical objects with structurally meshed geometry.

Notes on rectangular SFCs

Please refer to **Bib.** [54] for scanning rectangular images. However, it is necessary to distinguish between rectangular curves generated graphically by grammars or heuristic methods, and rectangular pairing curves generated combinatorically. Among the latter, Gray's metacurves introduced in **Bib.** [114] generalize filling curves to rectangular shapes and, more generally, to rectilinear polygons. Gray metacurves use new families of Gray curves such as the yaw curves described in **Bib.** [109], which lend themselves to rectangular generalizations. An algorithm for drawing rectangular SFCs is given by **Bib.** [38], then taken up and extended to 3 dimensions in **Bib.** [103].

Notes on meshes

Some meshing methods prefer quadrilateral meshes to triangular meshes and dissect triangles into 3 quadrilaterals (**Bib.** [91]). A hexagonal meshing method that can easily be converted into a triangular and quadrilateral mesh is presented in **Bib.** [65]. Obtaining planar quadrilateral meshes for architectural structures is presented in **Bib.** [64]. An overview of techniques for meshing complex shapes from quadrilaterals can be found in **Bib.** [84] as well as in **Bib.** [53] and **Bib.** [69].

For a general introduction to mesh theory, see **Bib.** [29]. For an introduction to volume mesh theory, please refer to **Bib.** [73]. For a general introduction to different types of mesh, please refer to **Bib.** [118]. For procedural meshes of the Catmull-Clark type, please refer to **Bib.** [6] and **Bib.** [48]. For the problem of mesh
5 quality based on triangles, see **Bib.** [19].

Notes on mesh traversability-ordering

Triangulated mesh traversability-ordering is studied for data compression in **Bib.** [27] and also in **Bib.** [63]. The existence of two- and three-dimensional Hamiltonian paths on grids is discussed in **Bib.** [47]. A methodology for
10 transforming unstructured triangular meshes into quadrilateral meshes and then searching for Hamiltonian paths through the quadrilaterals is presented in **Bib.** [76] and **Bib.** [75]. HAMSTRAND technologies are presented in **Bib.** [60] **Bib.** [82] and **Bib.** [94]. The topological organization of cells and their memory neighbors is presented in **Bib.** [101]. The problem of representing massive terrain models
15 using triangles is described in **Bib.** [89]. Techniques for representing triangular meshes using ribbons are described in **Bib.** [31]. SFC-based data traversability-ordering from a memory perspective is discussed in **Bib.** [68] and also in **Bib.** [46]. The use of Delaunay methods to improve mesh quality is introduced in **Bib.** [22]. In the same vein, **Bib.** [56]. Mesh traversal using B-splines is described in
20 **Bib.** [37] for compression applications. Error correction in quantum codes also gives rise to path problems on meshes mapped onto surfaces such as toric **Bib.** [36] and **Bib.** [121].

Notes on meshes and SFC

The connection between grid indexing and SFC indexing is introduced in **Bib.** [79]. Mesh or graph partitions also bring together SFC theories and Hamiltonian
25 path search (**Bib.** [49], **Bib.** [80], **Bib.** [87]).

Notes on SFC and object indexing

The indexing of any object, digital or real, is important in the field of target acquisition **Bib.** [77] in telemetry or wire-guidance.

Notes on mapping principles

Historically, the convergence between classical SFCs and their mapping to surfaces, in particular the sphere, is mainly due to cartographers **Bib.** [4]. In **Bib.** [74] we find an mapping of SFC on a sphere that respects an equidistribution
5 of points on the surface, based on an algorithm presented in **Bib.** [42]. For the first three mappings on a sphere illustrated in figure **Fig.** 20, the parametric equations are described in **Bib.** [120] based on trigonometric functions. The principle of bijective correspondence between the three-dimensional points of these surfaces and the two-dimensional points of the SFC metacurve to be
10 applied to them is well described and illustrated in **Bib.** [119]. The mapping of quadrilateral meshes, and textures is developed in **Bib.** [85].

Notes on mapping types

Mapping methods are closely associated with dedicated coordinate systems, particularly in analytical geometry and cartography. The following references **Bib.** [57], **Bib.** [78], **Bib.** [83] and **Bib.** [116] provide a comprehensive overview of
15 these methods, mainly in two dimensions. For formulations of parametric equations for surfaces in three dimensions, please refer to **Bib.** [120].

Notes on vector deformations

For an overview of vector deformations, please refer to **Bib.** [88] for an overview
20 of general techniques for surface mappings and distortions. Conformal mappings are introduced in **Bib.** [81]. For the specific study of SFC mappings with vector deformations on mathematical surfaces and three-dimensional object models, please refer to the following seminal works **Bib.** [51], **Bib.** [59] and **Bib.** [72].

Notes on graphical meshes

25 The study of spatial structures and certain types of surfaces can also be carried out graphically, using descriptive geometry or projective geometry (**Bib.** [7]). In this case, parametrization and u , v patches can be obtained graphically. Purely two-dimensional graphical techniques can also be used to generate grid patterns

close to the projections of ruled surfaces (**Bib.** [20]). A presentation of quadric ruled surfaces without the use of equations is given in **Bib.** [55].

Notes on Peano chaining

The symbolic representation of arc-filling curves is shown schematically in a publication by Peano **Bib.** [2]. In this case, arcs cross quadrilaterals, joining adjacent vertices. These arcs symbolize filling curves of variable resolution and U-shaped topology. More recently, vertex-face graphs linked to triangulated meshes have been introduced by a series of articles by Bartholdi, the main ones being **Bib.** [33], **Bib.** [41] and **Bib.** [40] with an introduction provided by **Bib.** [44] or **Bib.** [43]. One of the main interests of this work is to introduce a real strategy for indexing triangulated meshes from filling curves. The first Peano chaining studies were codified by **Ref.** [5].

Notes on polygonal diagrams

A good introduction can be found in (**Bib.** [12]). In particular, they are used in the theory of simplicial complexes **Bib.** [70] and have been introduced as design aids in **Bib.** [17].

Notes on parametric surfaces

Parametric surfaces and their equations are well documented in tutorials for symbolic calculation systems such as Maplesoft (**Bib.** [34], **Bib.** [23]), Mathematica or the Matlab environment. A compilation of examples is also provided in **Bib.** [71]. The Klein-Jeener bottle is introduced in **Bib.** [45]. The supertoroid equations are developed in **Bib.** [8] and **Bib.** [16]. Two presentations of generalized cylinders are found in **Bib.** [10] and **Bib.** [24].

Notes on Bezier hyperpatches

An introduction to parametric volumes in solid modeling is given in **Bib.** [13]. A general introduction to parametric volumes and hyperpatches is given in **Bib.** [52]. A presentation of B-splines in Mathematica code is given in **Bib.** [93]. The deformation potential of Bezier hyperpatches in haptic interfaces is described in

Bib. [86]. The principle of folding Bezier volumes into fractally generated planar SFC layers is introduced in **Bib.** [58]. Some interface problems of Bezier volumes are discussed in **Bib.** [28]. The realization of 64-control-point hyper-patches from the stacking of 16-control-point patches without self-intersections is illustrated in

5 **Bib.** [102]. References **Bib.** [62] and **Bib.** [122] provide an introduction to the problem of unfolding the human genome and its relationship with three-dimensional CFS. Volumetric medical imaging (MRI) also makes use of adaptive SFCs for voxel ordering in classification processes (**Bib.** [98]).

Notes on Bezier's modelisation

10 The theory of parametric surfaces and Bezier curve modeling is covered in **Bib.** [25]. Data on Catmull's Gumbo elephant are available in **Bib.** [61] and those for Newell's teapot in **Bib.** [32]. The problem of connecting u , v between neighboring patches (watertightness) is introduced in **Bib.** [26]. Meshing regions bounded by Bezier curves is discussed in **Bib.** [67].

15 ***Notes on trajectory optimization***

The optimization of trajectories using filling curves has been the subject of work related to the travelling salesman problem. Barthodi has already been cited as an author or co-author in **Bib.** [11], **Bib.** [9], **Bib.** [15], **Bib.** [21], **Bib.** [14] and more recently **Bib.** [66] should also be cited. Bartholdi's work favors the use of

20 Sierpinski's filling curves, considered well suited to the indexing of triangles. In **Bib.** [104] introduces SFCs to the problem of finding nearest neighbors.

ALGORITHMIC CODE

Algorithm Alg.1 parSPH

The function **parSPH** is implemented by the algorithm **Alg.** 1 in figure **Fig.** 38.

25 The function calculates the list of points on a sphere of radius R from a parametrization function indicated by the value f . The points calculated belong to a grid of resolution a . The limits of the parametrization are given by the $uMxMn$ and $vMxMn$ pairs of minmax values of the parameters in u and v . The function

returns a list of grid points in three dimensions.

Algorithm Alg.2 sfcSPH

The function **sfcSPH** is implemented by the algorithm **Alg. 2** shown in figure **Fig. 39**. The function calculates the list of points of an SFC metacurve applied to a sphere of radius R. The parameters of the function are the same as those of **Alg. 1**. The SFC metacurve is fixed in the following code (double spiral SFC) but can be generalized to any SFC metacurve accessible via a mapping table. The function returns the list of points in the three-dimensional metacurve. The double loop and parameterization of the **parSPH** function has been replaced by a single loop and parameterization.

Algorithm Alg.3 tabSPH

The function **tabSPH**, shown in figure **Fig. 40**, serves as a mapping table between the **parSPH** and **sfcSPH** functions and the various sphere parameterizations given as examples, corresponding to functions **SPH0** in figure **Fig. 41**, **SPH1** in figure **Fig. 42**, **SPH2** in figure **Fig. 43** and **SPH3** in figure **Fig. 44**. The function parameters are the same as for the **parSPH** and **parSPH** functions. The function returns v3 the triplet of coordinates of the calculated point on the sphere.

Algorithm Alg.8 DC0uwLL

The function **DC0uwLL**, shown in figure **Fig. 45**, is implemented by the algorithm **Alg. 8**. The function calculates the coordinates of a point on a level-1 Hilbert strip curve from the point's index. The function receives as input *ind* the index of the point and the pair v2 of the length and height of the curve. It returns the pair of coordinates of the given index point.

Algorithm Alg.9 CD0uwLL

The function **CD0uwLL**, shown in figure **Fig. 46**, is implemented by the algorithm **Alg. 9**. It is the coding function associated with the previous decoding function **DC0uwLL**. The function receives as input the pair v2 of coordinates of the point whose index is being sought and the pair v2ab of the length and height of the

support grid of the striped Hilbert curve.

Algorithm Alg.10 PATCHuv

The function **PATCHuv**, shown in **Fig. 47**, is implemented by the algorithm **Alg. 10**. The function calculates the coordinates of the points belonging to the grid formed by the U and V parametric curves on the surface. The function receives as input the list *Is* of the 16 control points of the patch and *n* the resolution of the UV grid. Function **BEZeval** is the function that implements Equation 3.

Algorithm Alg.11 PATCHsfc

The function **PATCHsfc**, shown in figure **Fig. 48**, is implemented by the algorithm **Alg. 11**. The function calculates the coordinates of points belonging to a sequence of SFCs traversing grids formed by U and V parametric curves on the surface. The function receives as input the list *Is* of patches, *a* the numerator of the SFC resolution expressed in rational form, *b* the denominator of the resolution, *IsOR* the list of SFC orientations for each patch, i.e. 0..3, and *IsINV* the list of path directions for each SFC, i.e. 0..1.

Algorithm Alg.12 PTsfc

The **PTsfc** function, shown in figure **Fig. 49**, is implemented by the algorithm **Alg. 12**. The function calculates the parameters *u*, *v* of the metacurve after homothetic transformation and rotation in the plane. The metacurve is introduced in the code as an example but can generally be called up externally via a hash table. The function returns the transformed parameters *ut*, *vt*.

Algorithm Alg.13 INTERPOL

The **INTERPOL** function, shown in figure **Fig. 50**, is implemented by the algorithm **Alg. 13**. The function calculates the coordinates of the control points of a bicubic Bezier patch from a list of crossing points. The function receives as input the list *IsPTSpas* of passage points and returns the list of control points.

Algorithm Alg.14 SOLUTana

The **SOLUTana** function, shown in **Fig. 51**, is implemented by the algorithm **Alg. 14**. The function calculates the coordinates of the control points of a bicubic patch by direct analytical solution. The function is called by the **INTERPOL** function for
5 each X, Y, Z coordinate.

Clauses

Clause 1. A computer-implemented method of indexing an n-dimensional object comprising a plurality of points, the method comprising:

5 obtaining a mesh of the n-dimensional shape of the object, the mesh comprising at least one cell;

determining a chaining traversing the at least one cell of the mesh; the chaining comprising for each cell a symbol traversing the cell and joining two vertices of the cell;

10 for each cell: associating an SFC metacurve with each symbol associated with two vertices; and

mapping the at least one SFC metacurve onto the object using a corresponding mapping function to associate with each point of the plurality of points a corresponding index of the at least one SFC metacurve.

15 Clause 2. The computer-implemented method claimed in clause 1, wherein the mesh comprises a plurality of cells; the determined chaining traverses the plurality of cells of the mesh; and the mapping is performed in order to associate with each point of the plurality of points a corresponding index of the plurality of SFC metacurves.

20 Clause 3. The computer-implemented method of clause 1 wherein the symbol is an arc.

Clause 4. The computer-implemented method as claimed in any one of clauses 1 to 3, further comprising providing an indication of the mapping.

25 Clause 5. The computer-implemented method as claimed in clause 4, wherein the providing of an indication of the mapping comprises at least one of saving the indication of the mapping in a file, displaying the indication of the mapping and providing the indication of the mapping to a remote processing device.

Clause 6. A computing device comprising:

at least one processor;

a display device;

an input/output interface;

5 a memory including instructions which when executed perform a method of indexing an n-dimensional object comprising a plurality of points, the method comprising:

10 obtaining via the input/output interface a mesh of the n-dimensional shape of the object, the mesh comprising at least one cell;

determining, by means of the at least one processor, a chaining traversing the at least one cell of the mesh; the chaining comprising for each cell a symbol traversing the cell and joining two vertices of the cell;

15 for each cell and by means of the at least one processor: associating an SFC metacurve with each symbol associated with two vertices; and

20 by means of the at least one processor, mapping the at least one SFC metacurve onto the object using a corresponding mapping function in order to associate with each point of the plurality of points a corresponding index of the at least one SFC metacurve; and

a bus for pairing said at least one processor, said display device, said input/output interface and said memory.

25

Clause 7. A computer-readable physical memory storing statements and instructions for execution by a computer, said statements and instructions comprising:

5 encoding means for obtaining a mesh of the n-dimensional shape of the object, the mesh comprising at least one cell;

encoding means for determining a chaining running through the at least one cell of the mesh; the chaining comprising for each cell a symbol traversing the cell and joining two vertices of the cell;

10 coding means for associating an SFC metacurve with each symbol associated with two vertices of each cell; and

coding means for mapping the at least one SFC metacurve onto the object using a corresponding mapping function in order to associate with each point of the plurality of points a corresponding index of the at least one SFC metacurve.

15 Clause 8. A computer program comprising computer-executable instructions which, when executed by a computer, perform computer-implemented method as claimed in any one of clauses 1 to 5.

Clause 9. The computer-implemented method as claimed in any one of clauses 1 to 5, further comprising subdividing at least one given cell of the mesh
20 to obtain at least two sub-cells within the given cell; wherein the chaining traversing the given cell traverses each of the at least two sub-cells, the chaining comprising a symbol for each of the at least two sub-cells and wherein the SFC metacurve associated with the given cell comprises at least two SFC metacurves, each of the at least two SFC metacurves being associated with a corresponding
25 sub-cell of the given cell.

REFERENCES

- [1] Albrechten Durer. Hierin sind begriffen vier bucher von menschlicher proportion. <https://collections.library.yale.edu/catalog/10613881>, 1528.
- [2] G. Peano. Formulario mathematico. Bocca, 1908.
- 5 [3] J Arthur Thomson. On growth and form. Nature Publishing Group UK London, 1917.
- [4] Guy M Morton. A computer oriented geodetic data base and a new technique in file sequencing. 1966.
- [5] Walter Wunderlich. Über peano-kurven. *Elemente der Mathematik*,
10 28 :1–10, 1973.
- [6] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6) :350–355, 1978.
- [7] Adrian Gheorghiu, Virgil Dragomir, and Dinu Manoil. Geometry of structural forms. Applied Science Publishers London, 1978.
- 15 [8] Alan H Barr. Superquadrics and angle-preserving transformations. *IEEE Computer graphics and Applications*, 1(1) :11–23, 1981.
- [9] Loren Kerry Platzman and JJ Bartholdi III. An $O(n \log n)$ planar travelling salesman heuristic based on spacefilling curves. Technical report, GEORGIA INST OF TECH ATLANTA PRODUCTION AND DISTRIBUTION
20 RESEARCH CENTER, 1982.
- [10] Steven A Shafer and Takeo Kanade. The theory of straight homogeneous generalized cylinders and a taxonomy of generalized cylinders. 1983.
- [11] John Joseph Bartholdi and Loren Kerry Platzman. Heuristics based on spacefilling curves for combinatorial problems in the plane. Technical report,
25 GEORGIA INST OF TECH ATLANTA PRODUCTION AND DISTRIBUTION RESEARCH CENTER, 1984.

- [12] Eric Lord and C. Wilson. The mathematical description of shape and form, 03 1984.
- [13] Malcolm S Casale and Edward L Stanton. An overview of analytic solid modeling. *IEEE Computer Graphics and Applications*, 5(2) :45–56, 1985.
- 5 [14] Dimitris Bertsimas and Michelangelo Grigniy. On the space filling curve heuristic for the euclidean traveling salesman problem. Technical report, Citeseer, 1988.
- [15] Loren K Platzman and John J Bartholdi III. Spacefilling curves and the planar travelling salesman problem. *Journal of the ACM (JACM)*, 36(4) :719–737,
10 1989.
- [16] Alan H Barr. Rigid physically based superquadrics. In *Graphics Gems III (IBM Version)*, pages 137–159. Elsevier, 1992.
- [17] Helaman Ferguson, Alyn Rockwood, and Jordan Cox. Topological design of sculptured surfaces. In *Proceedings of the 19th annual conference on*
15 *Computer graphics and interactive techniques*, pages 149–156, 1992.
- [18] Heinz-Otto Peitgen, Hartmut Jürgens, Dietmar Saupe, and Mitchell J Feigenbaum. Chaos and fractals : new frontiers of science. Springer, 1992.
- [19] L Paul Chew. Guaranteed-quality mesh generation for curved surfaces. In *Proceedings of the ninth annual symposium on Computational geometry*,
20 pages 274–280, 1993.
- [20] Dale Seymour, Linda Silvey, and Joyce Snider. Line designs : Designs and drawings. Ideal School Supply, 1994.
- [21] John J Bartholdi III. A routing system based on spacefilling curves. *Georgia Institute of*, 1995.
- 25 [22] Jim Ruppert. A delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of algorithms*, 18(3) :548–585, 1995.

- [23] Colm Mulcahy. The basic curves and surfaces of computer aided geometric design. *MAPLETECH*, 3(1) :65–73, 1996.
- [24] Nicholas HE Pillow. *Recognition of generalized cylinders using geometric invariance*. PhD thesis, University of Oxford, 1996.
- 5 [25] Michael E Mortenson. Geometric modeling. John Wiley & Sons, Inc., 1997.
- [26] Andrey A Mezentsev and Thomas Woehler. Methods and algorithms of automated cad repair for incremental surface meshing. In *IMR*, pages 299–309. Citeseer, 1999.
- 10 [27] Jarek Rossignac. Edgebreaker : Connectivity compression for triangle meshes. *IEEE transactions on visualization and computer graphics*, 5(1) :47–61, 1999.
- [28] Benjamin Schmitt, Alexander Pasko, and Vladimir Savchenko. Extended space mapping with bezier patches and volumes. In *Implicit Surfaces*, volume 99, pages 25–31, 1999.
- 15 [29] Marshall W Bern and Paul E Plassmann. Mesh generation. *Handbook of computational geometry*, 38, 2000.
- [30] Revital Dafner, Daniel Cohen-Or, and Yossi Matias. Context-based space filling curves. In *Computer Graphics Forum*, volume 19, pages 209–218. Wiley Online Library, 2000.
- 20 [31] Jihad El-Sana, Francine Evans, Aravind Kalaiah, Amitabh Varshney, Steven Skiena, and Elvir Azanli. Efficiently computing and updating triangle strips for real-time rendering. *Computer-Aided Design*, 32(13) :753–772, 2000.
- [32] Thomas V Thompson II. teapot : this is the original non-rational bezier teapot data. the data comes from martin newel and jim blinn. <https://users.cs.utah.edu/~dejohnso/models/teapotBezier>, January 2000.
- 25

- [33] John J Bartholdi III and Paul Goldsman. Vertex-labeling algorithms for the hilbert spacefilling curve. *Software : Practice and Experience*, 31(5) :395–408, 2001.
- [34] M. B. Monagan K. O. Geddes K. M. Heal G. Labahn S. M. Vorkoetter
5 J. McCarron P. DeMarco. Maple 7 programming guide. Waterloo Maple Inc, 2001.
- [35] GA Amiryants, VD Grigorive, FZ Ishmurov, A Franz, E d’Henin, and B Kaempfer. Investigations of airship aeroelasticity, 2002.
- [36] A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of*
10 *Physics*, 303(1) :2–30, jan 2003.
- [37] Anshuman Razdan, Vinodth Kumar Mohanram, and Gerald E Farin. Triangle mesh compression using b-spline curves. In *CISST*, pages 378–383. Citeseer, 2003.
- [38] Lutz Tautenhahn. Draw a space-filling curve of arbitrary size.
15 http://lutanho.net/pic2html/draw_sfc.html, 2003.
- [39] Michael Aftosmis, Marsha Berger, and Scott Murman. Applications of space-filling-curves to cartesian methods for cfd. In *42nd AIAA Aerospace Sciences Meeting and Exhibit*, page 1232, 2004.
- [40] John J Bartholdi and Paul Goldsman. Multiresolution indexing of
20 triangulated irregular networks. *IEEE Transactions on Visualization and Computer Graphics*, 10(4) :484–495, 2004.
- [41] John J Bartholdi Iii and Paul Goldsman. The vertex-adjacency dual of a triangulated irregular network has a hamiltonian cycle. *Operations Research Letters*, 32(4) :304–308, 2004.
- [42] Markus Deserno. How to generate equidistributed points on the surface
25 of a sphere. *If Polymerforschung (Ed.)*, 99(2), 2004.

- [43] Perouz Taslakian. Hamiltonian cycles in the vertex-adjacency dual. <http://cgm.cs.mcgill.ca/~athens/cs507/Projects/2004/PerouzTaslakian/introduction.htm>, 2004.
- [44] Perouz Taslakian and Godfried Toussaint. Hamiltonian cycles in sparse vertex-adjacency duals. In *14th Annual Fall Workshop on Computational Geometry*, page 30. Citeseer, 2004.
- [45] Arpam. Patrice jeener graveur sur cuivre. <http://arpam.free.fr/jeener.html>, 2005.
- [46] Guohua Jin and John Mellor-Crummey. Using space-filling curves for computation reordering. In *Proceedings of the Los Alamos Computer Science Institute Sixth Annual Symposium*. Citeseer, 2005.
- [47] William F Mitchell. Hamiltonian paths through two-and three-dimensional grids. *Journal of Research of the National Institute of Standards and Technology*, 110(2) :127, 2005.
- [48] H-R Pakdel and Faramarz Samavati. Incremental catmull-clark subdivision. In *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)*, pages 95–102. IEEE, 2005.
- [49] Stefan Schamberger and Jens-Michael Wierum. Partitioning finite element meshes using space-filling curves. *Future Generation Computer Systems*, 21(5) :759–766, 2005.
- [50] Tony Tung and Francis Schmitt. Indexation de modèles 3d par graphe de reeb multi-résolution augmenté. *Annales des Télécommunications*, 60 :1309–1336, 01 2005.
- [51] Jonathan A Quinn, Frank C Langbein, Ralph R Martin, and Gershon Elber. Density-controlled sampling of parametric surfaces using adaptive space-filling curves. In *Geometric Modeling and Processing-GMP 2006 : 4th International Conference, Pittsburgh, PA, USA, July 26-28, 2006. Proceedings 4*, pages 465–484. Springer, 2006.

- [52] Martin Samuelčík. Bézier and b-spline volumes. Projekt dizertačnej práce. Bratislava, 2006.
- [53] Alla Sheffer, Emil Praun, Kenneth Rose, et al. Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision*, 2(2) :105–171, 2007.
- [54] Jian Zhang, Sei-Ichiro Kamata, and Yoshifumi Ueshige. A pseudo-hilbert scan for arbitrarily-sized arrays. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 90(3) :682–690, 2007.
- [55] Hugues Vermeiren. Les quadriques réglées sans équations. *Bulletin de l'APMEP*, (477) :489–495, 2008.
- [56] Oleksiy Busaryev, Tamal K. Dey, and Joshua A. Levine. Repairing and meshing imperfect shapes with delaunay refinement. SPM '09, New York, NY, USA, 2009. Association for Computing Machinery.
- [57] Yasser Ebrahim, Maher Ahmed, Wegdan Abdelsalam, and Siu-Cheung Chau. Shape representation and description using the hilbert curve. *Pattern Recognition Letters*, 30(4) :348–358, 2009.
- [58] Saravana Kumar Gurunathan, Ponnusamy Pandithevan, and Appa Ambatti. Fractal raster tool paths for layered manufacturing of porous objects. *Virtual and Physical Prototyping*, 4 :91–104, 06 2009.
- [59] Jonathan Alexander Quinn. Low-discrepancy point sampling of 2d manifolds for visual computing. Cardiff University (United Kingdom), 2009.
- [60] Andrew Wissink, Aaron Katz, William Chan, and Robert Meakin. Validation of the strand grid approach. In *19th AIAA Computational Fluid Dynamics*, page 3792. 2009.
- [61] The Little Grasshopper. Quad tessellation with opengl 4.0 and patch surfaces and vertex welding. <https://prideout.net/blog/old/blog/index.html@p=49.html>, 2010.

- [62] Ana Lyons. How the human genome folds in 3-d.
<https://thetech.com/2010/03/09/lieberman-v130-n11>, 2010.
- [63] Topraj Gurung, Mark Luffel, Peter Lindstrom, and Jarek Rossignac. Lr : Compact connectivity representation for triangle meshes. In *ACM SIGGRAPH 2011 papers*, pages 1–8. 2011.
- [64] Yang Liu, Weiwei Xu, Jun Wang, Lifeng Zhu, Baining Guo, Falai Chen, and Guoping Wang. General planar quadrilateral mesh design using conjugate direction field. *ACM Transactions on Graphics (TOG)*, 30(6) :1–10, 2011.
- [65] Christian Müller. Conformal hexagonal meshes. *Geometriae dedicata*, 154 :27–46, 2011.
- [66] Yi-Chih Hsieh and Peng-Sheng You. A new space-filling curve based method for the traveling salesman problems. *Appl. Math. Inf. Sci.*, 6 :371–377, 2012.
- [67] Orest Shardt and John C Bowman. Surface parameterization of nonsimply connected planar bézier regions. *Computer-Aided Design*, 44(5) :484–e1, 2012.
- [68] Huy T Vo, Claudio T Silva, Luiz F Scheidegger, and Valerio Pascucci. Simple and efficient mesh layout with space-filling curves. *Journal of Graphics Tools*, 16(1) :25–39, 2012.
- [69] David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. Quad-mesh generation and processing : A survey. In *Computer graphics forum*, volume 32, pages 51–76. Wiley Online Library, 2013.
- [70] Peter Giblin. Graphs, surfaces and homology : an introduction to algebraic topology. Springer Science & Business Media, 2013.
- [71] Rodrigo Platte. Parametric surfaces.
<https://www.chebfun.org/examples/geom/ParametricSurfaces.html>, 2013.

- [72] Jonathan A Quinn, Frank C Langbein, Yu-Kun Lai, and Ralph R Martin. Fast low-discrepancy sampling of parametric surfaces and meshes. In *Proceedings of the Conference on the Mathematics of Surfaces XIV. IMA*, page 43, 2013.
- 5 [73] Lionel Untereiner. *Représentation des maillages multirésolutions : application aux volumes de subdivision*. PhD thesis, Université de Strasbourg, 2013.
- [74] Mark McClure. Equitably distributed curve on a sphere. <https://mathoverflow.net/posts/189338/revisions>, 2014.
- 10 [75] Jay Sitaraman and Beatrice Roget. Solution algorithm for unstructured grids using quadrilateral subdivision and hamiltonian paths. 01 2014.
- [76] Jayanarayanan Sitaraman and Beatrice Roget. Solution algorithm for unstructured grids using quadrilateral subdivision and hamiltonian paths. In *52nd Aerospace Sciences Meeting*, page 0079, 2014.
- 15 [77] Prem Anand, Ramakrishnan Pandiarajan, and P Govinda Raju. Wireless power transmission to uav using laser beaming. 2015.
- [78] Chamberlain Fong. Analytical methods for squaring the disc. *arXiv preprint arXiv :1509.06344*, 2015.
- [79] Mark McKenney. Introduction to spatial
20 indexing. http://www.cs.siu.edu/~marmcke/docs/cs490/_static/AdvancedDatabases.pdf, 2015.
- [80] Aparna Sasidharan and Marc Snir. Space-filling curves for partitioning adaptively refined meshes. *Mathematics and Computer Science*, 2015.
- [81] Alexander I Bobenko, Stefan Sechelmann, and Boris Springborn.
25 Discrete conformal maps : Boundary value problems, circle domains, fuchsian and schottky uniformization. Springer Berlin Heidelberg, 2016.

- [82] Yong Su Jung, Bharath Govindarajan, and James Baeder. A hamiltonian-strand approach for aerodynamic flows using overset and hybrid meshes. In *72nd Annual Forum of the American Helicopter Society*, 2016.
- [83] Martin Lambers. Mappings between sphere, disc, and square. *Journal of Computer Graphics Techniques Vol*, 5(2), 2016.
- [84] Marcel Campen. Partitioning surfaces into quadrilateral patches : A survey. In *Computer Graphics Forum*, volume 36, pages 567–588. Wiley Online Library, 2017.
- [85] Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. Scalable locally injective mappings. *ACM Transactions on Graphics (TOG)*, 36(4):1, 2017.
- [86] Celia Romo, Francisco A Conde, Nigel W John, and Juan Carlos Torres. Modeling deformable objects for medical training with haptic devices. In *2017 International Conference on Cyberworlds (CW)*, pages 202–205. IEEE, 2017.
- [87] Ricard Borrell, Juan Carlos Cajas, Daniel Mira, Ahmed Taha, Seid Koric, Mariano Vázquez, and Guillaume Houzeaux. Parallel mesh partitioning based on space filling curves. *Computers & Fluids*, 173 :264–272, 2018.
- [88] ETH-2520538-00L. Mappings representation and distorsion. <https://igl.ethz.ch/teaching/shape-modeling/sm2021/>, 2018.
- [89] Kavisha Kumar, Hugo Ledoux, and Jantien Stoter. Compactly representing massive terrain models as tins in citygml. *Transactions in GIS*, 22(5):1152–1178, 2018.
- [90] Arthur Masson. Space filling curves. <http://arthurmasson.xyz/space-filling-curves/>, 2018.
- [91] Julia Docampo-Sanchez and Robert Haimes. Towards fully regular quad mesh generation. In *AIAA Scitech 2019 Forum*, page 1988, 2019.

- [92] Jim Hughes and Chris Eichelberger. Space-filling curves and the art of composition. <https://www.geomesa.org/assets/outreach/FOSS4GNA2019-ComposingSFCs.pdf>, 2019.
- [93] Robert Ipanaqué, Ricardo Velezmoro, and RT Urbina. Bspline solids manipulation with mathematica. *arXiv preprint arXiv :1906.07574*, 2019.
- [94] Yong Su Jung. *Hamiltonian Paths and Strands for Unified Grid Approach for Computing Aerodynamic Flows*. PhD thesis, University of Maryland, College Park, 2019.
- [95] Doug McKenna. Hilbert curves : Outside-in and inside gone. <https://apps.apple.com/us/app/hilbert-curves/id1453611170>, 2019.
- [96] Michael Powell, Daniel Alejandro Ibanez, Timothy Jesse Fuller, Brian Neal Granzow, Thomas E Voth, and Glen Hansen. Multi-material dynamic domain topology changes in the lagrangian grid reconnection (lgr) code. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2019.
- [97] Alban de Vaucorbeil, Vinh Phu Nguyen, Sina Sinaie, and Jian Ying Wu. Material point method after 25 years : Theory, implementation, and applications. *Advances in applied mechanics*, 53 :185–398, 2020.
- [98] Unal Sakoglu, Lohit Bhupati, Nazanin Beheshti, Nikolaos Tsekos, and Lennart Johnsson. An adaptive space-filling curve trajectory for ordering 3d datasets to 1d : Application to brain magnetic resonance imaging data for classification. In *Computational Science–ICCS 2020 : 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part III* 20, pages 635–646. Springer, 2020.
- [99] Hatem Soliman, Ahmed Shash, T. El-Hossainy, and M. Abd-Rabou. Investigation of process parameters in orthogonal cutting using finite element approaches. *Heliyon*, 6 :1–10, 11 2020.

- [100] Liang Zhou, Chris R Johnson, and Daniel Weiskopf. Data-driven space-filling curves. *IEEE transactions on visualization and computer graphics*, 27(2):1591–1600, 2020.
- [101] Solal Amouyal and Alex Margolin. A novel framework for unstructured meshes with optimized cell ordering using hamiltonian paths. In *AIAA Scitech 2021 Forum*, page 1902, 2021.
- [102] Xuanyi Zhao, Jinggai Li, Shiqi He, and Chungang Zhu. Geometric conditions for injectivity of 3d bézier volumes. *AIMS Math*, 6 :11974–11988, 2021.
- 10 [103] Jakub Červený. gilbert. <https://github.com/jakubcerveny/gilbert>, 2021.
- [104] Konstantin Barkalov, Anton Shtanyuk, and Alexander Sysoyev. A fast knn algorithm using multiple space-filling curves. *Entropy*, 24(6), 2022.
- [105] Dharna Nar and Radhika Kotecha. Optimal waypoint assignment for designing drone light show formations. *Results in Control and Optimization*, 9 :100174, 2022.
- 15 [106] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques à centre. <https://hal.archives-ouvertes.fr/hal-03745896/file/Article13.pdf>, August 2022. working paper or preprint.
- 20 [107] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques à noyaux. <https://hal.archives-ouvertes.fr/hal-03745216/file/Article14.pdf>, August 2022. working paper or preprint.
- [108] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques à réflexions. <https://hal.archives-ouvertes.fr/hal-03744206/file/Article15.pdf>, August 2022. working paper or preprint.
- 25

- [109] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques en forme de lacets. <https://hal.archives-ouvertes.fr/hal-03737365/file/Article10.pdf>, July 2022. working paper or preprint.
- 5 [110] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques en forme de méandres. <https://hal.archives-ouvertes.fr/hal-03737998/file/Article11.pdf>, July 2022. working paper or preprint.
- [111] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques en forme de spirales. <https://hal.archives-ouvertes.fr/hal-03738025/file/Article12.pdf>, July 2022. working paper or preprint.
- 10 [112] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes de Gray surfaciques en forme d'escaliers. <https://hal.archives-ouvertes.fr/hal-03745141/file/Article16.pdf>, August 2022. working paper or preprint.
- 15 [113] Jean-François Rotgé. Fonctions de couplage et chaînage des courbes et metacourbes de Gray généralisant les courbes de Hilbert. <https://hal.science/hal-03773928/file/Article9.pdf>, September 2022. working paper or preprint.
- 20 [114] Jean-François Rotgé. Fonctions de couplage et chaînage des metacourbes de Gray surfaciques à rotors. <https://hal.archives-ouvertes.fr/hal-03768804/file/Article2.pdf>, September 2022. working paper or preprint.
- [115] Jean-François Rotgé. Principes de chaînage des courbes de Gray. <https://hal.archives-ouvertes.fr/hal-03759184/file/Article5.pdf>, August 2022. working paper or preprint.
- 25

- [116] Irish Times. Space-filling curves, part ii : Computing the limit function, August 11 2022. <https://thatsmaths.com/2022/08/11/space-filling-curves-part-ii-computing-the-limi>
- [117] Hanyu Wang, Kamal Gupta, Larry Davis, and Abhinav Shrivastava.
5 Neural space-filling curves, 2022.
- [118] Ricardo Camarero. Génération de maillage : introduction.
https://moodle.polymtl.ca/pluginfile.php/781056/mod_resource/content/7/frn_introduction.pdf, 2023.
- [119] Jean-François COLONNA. Hilbert and peano space filling curves and
10 beyond : From squares and cubes to surfaces (bidimensional manifolds) and
tridimensional manifolds, 2023.
<http://www.lactamme.polytechnique.fr/Mosaic/descripteurs/LesCourbesRemplissantes.01.Ang.html>.
- [120] Robert FERRÉOL. Surfaces.
15 <https://mathcurve.com/surfaces/surfaces.shtml>, 2023.
- [121] Arthur Pesah. An interactive introduction to the surface code.
<https://arthurpesah.me/blog/2023-05-13-surface-code/>, 2023.
- [122] Aiden Lab Gupi Ranganathan* and Erez Lieberman Aiden. Unfolding the genome. <https://groundworks.io/journal/projects/86>, November 2, 2021.

CLAIMS:

1. A computer-implemented method of indexing an n-dimensional object comprising a plurality of points, the method comprising:

obtaining a mesh of the n-dimensional shape of the object, the mesh comprising at least one cell;

determining a chaining traversing the at least one cell of the mesh; the chaining comprising for each cell a symbol traversing the cell and joining two vertices of the cell;

for each cell: associating an SFC metacurve with each symbol associated with two vertices; and

mapping the at least one SFC metacurve onto the object using a corresponding mapping function to associate with each point of the plurality of points a corresponding index of the at least one SFC metacurve.

2. The computer-implemented method claimed in claim 1 wherein the mesh comprises a plurality of cells; the determined chaining traverses the plurality of cells of the mesh; and the mapping is performed in order to associate with each point of the plurality of points a corresponding index of the plurality of SFC metacurves.

3. The computer-implemented method of claim 1 wherein the symbol is an arc.

4. The computer-implemented method as claimed in any one of claims 1 to 3, further comprising providing an indication of the mapping.

5. The computer-implemented method as claimed in claim 4, wherein the providing of an indication of the mapping comprises at least one of saving the indication of the mapping in a file, displaying the indication of the mapping and providing the indication of the mapping to a remote processing device.

6. A computing device comprising:

at least one processor;

a display device;

an input/output interface;

a memory including instructions which when executed perform a method of indexing an n-dimensional object comprising a plurality of points, the method comprising:

obtaining via the input/output interface a mesh of the n-dimensional shape of the object, the mesh comprising at least one cell;

determining, by means of the at least one processor, a chaining traversing the at least one cell of the mesh; the chaining comprising for each cell a symbol traversing the cell and joining two vertices of the cell;

for each cell and by means of the at least one processor: associating an SFC metacurve with each symbol associated with two vertices; and

by means of the at least one processor, mapping the at least one SFC metacurve onto the object using a corresponding mapping function in order to associate with each point of the plurality of points a corresponding index of the at least one SFC metacurve; and

a bus for pairing said at least one processor, said display device, said input/output interface and said memory.

7. A computer-readable physical memory storing statements and instructions for execution by a computer, said statements and instructions comprising:

encoding means for obtaining a mesh of the n-dimensional shape of the object, the mesh comprising at least one cell;

encoding means for determining a chaining running through the at least one cell of the mesh; the chaining comprising for each cell a symbol traversing the cell and joining two vertices of the cell;

coding means for associating an SFC metacurve with each symbol associated with two vertices of each cell; and

coding means for mapping the at least one SFC metacurve onto the object using a corresponding mapping function in order to associate with each point of the plurality of points a corresponding index of the at least one SFC metacurve.

8. A computer program comprising computer-executable instructions which, when executed by a computer, perform computer-implemented method as claimed in any one of claims 1 to 5.

9. The computer-implemented method as claimed in any one of claims 1 to 5, further comprising subdividing at least one given cell of the mesh to obtain at least two sub-cells within the given cell; wherein the chaining traversing the given cell traverses each of the at least two sub-cells, the chaining comprising a symbol for each of the at least two sub-cells and wherein the SFC metacurve associated with the given cell comprises at least two SFC metacurves, each of the at least two SFC metacurves being associated with a corresponding sub-cell of the given cell.