BioFab Toolbox ï Software tools for biofabrication

N. Alves, P. Bártolo, N. Ferreira, M. Gaspar & A. Mateus
Centre for Rapid and Sustainable Product development, CDRsp, Polytechnic Institute of Leiria, Portugal

ABSTRACT: This is a report on the progress of the project BioFab Toolbox (PTDC/EME-CRO/120585/2010), funded by the Portuguese National Science Foundation (FCT). It consists in the development of a collection of software tools to assist the process of rapid prototyping and rapid manufacturing of new products. It is especially designed to produce scaffolds, grafts, implants and other medical applications, that usually appear under the general designation of biofabrication (biofab).

Keywords: STL file, slicing, scanning, toolpath, rapid prototyping, biofabrication;

1 INTRODUCTION

At CDRSP, many researchers are interested in the development of scaffolds and implants for medical applications using many different systems of rapid prototyping and manufacturing. The long willing of having a general platform independent method, easily adapted to new restrictions, constrains and specifications soon became an urgent necessity at each new experiment. Thus a set of routines was started to be developed, and it is still under current progress of improvement, that enables us with the flexibility and at the same time, efficiency and consistence, to manipulate each new experiment in the aim of rapid prototyping and manufacturing of scaffolds and implants for medical applications, with an increasing slope of competency. Some examples of this work can be found in [4]. At the end of this text we also display some images obtained with the software while generating the slicing, scanning and toolpath of some applications.

The collection of the available tools in the toolbox is the following one (see also flowchart of Figure 2).

- Stl2tri: transforms an STL file into a triangulation. The STL format (see figure 3 below) consists in a collection of disjoint triangular faces which is transformed into a collection of disjoint vertices together with a collection of face incidences with pointers into the collection of vertices;
- Tri2gph: transforms a given triangulation, with a specified z-coordinate, z0, into the structure of a graph with a symmetry which models the resulting contour level given by the intersection of the triangulated object with the plane z=z0. Further details on this structure are given for instance in [1];
- Gph2scn: transforms a given contour level, modeled by the structure of a graph with a symmetry, into a sequence of points, lying in the path defined by the given contour level, following a zig-zag strategy, in order to obtain a scanning toolpath for the interior of the region defined by the given input contour level, as illustrated in Figure 1;
- Scn2prn: transforms a given sequence of points, obtained by the scanning of the interior of a contour, into points with (x,y,z)-coordinates creating a file with the appropriate code, i.e. g-code, machine code, etc., accordingly to the specifications of the controllers used by the printing machine that is being used and sends it to printing.

In this text we will give some details on each one of the above steps involved in the biofabrication process, and specify some further directions for future work, that is, we will present the current status of each one of the routines outlined above indicating what is being planned for future improvements.

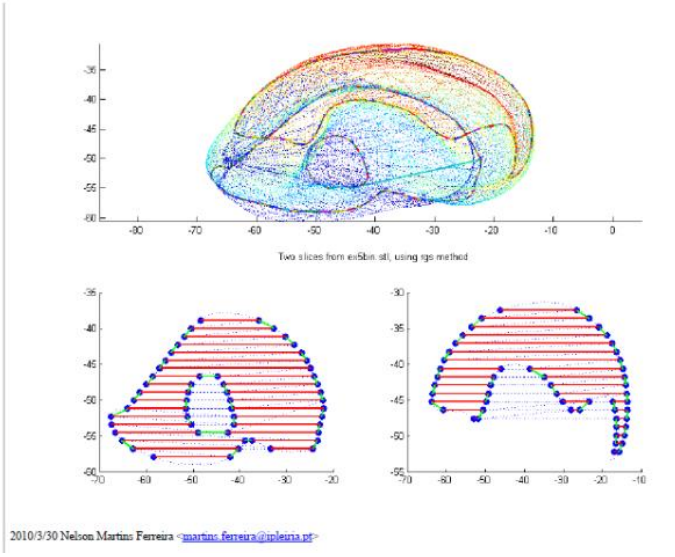


Figure 1 - Examples of output images produced by the slicing and scanning process of an STL file.

Improvements on step one include for example to read other data types such as parameterized geometric models. In step two we expect to be able to slice in non-planar geometries. In step three we expect to be able to use other strategies rather than zig-zag strategies, for instance we hope to be able to produce homotopic ones, or space filling curves such as the Peano curve. In step four we expect to control as much as possible the machine specifications, such as the velocity, the rotation, stopping times, etc.

2 READING PROCESS

The reading process consists in reading a file with geometrical information, such as a CAD model, an STL file or any other structured file with geometric primitives and/or parametric specifications with the respective instructions on how to obtain the desired modeled object to be produced. At present time we are only able to read from STL files. This is not really a serious handicap since all the other possibilities from which geometrical data are to be imported have the option to prepare an appropriate STL file. Nevertheless, in the future we plan to enhance the toolbox in order to be able to read directly from other file formats. Especially the ones that are defined by primitive geometrical entities or parameterized geometrical functions, since this will also allow us to have a better control of the geometrical information and not having to rely on interpolated triangles, avoiding in this way the errors arising from possible approximations from curved facets into planar triangular ones.

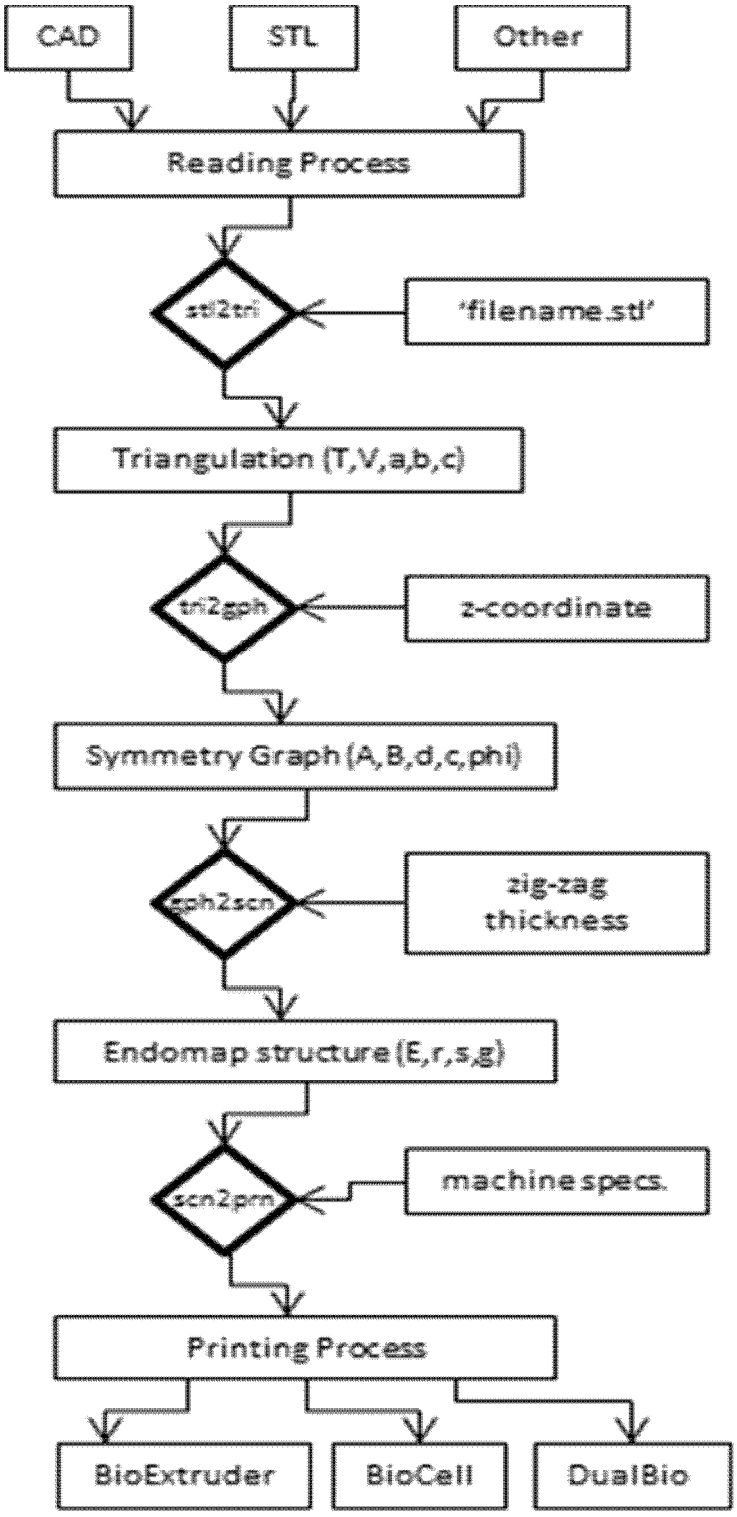


Figure 2 - Flowchart describing the tools in the biofab toolbox.

The collected information is transformed into a triangulation. That is, the collection of all triangles with specifyed triples of vertices, read from any STL file, is transformed into a list of unique vertices, V, together with a n-by-3 matrix, T, in which n is the total number of triangles, and such that each one of its lines is a triple of integer indices of the form T(l,1:3)=[i,j,k] with the meaning that the line number l in the matrix T represents the l-th triangle in the

original STL file, and it consists of the triple of vertices V(i,1:3), V(j,1:3) and V(k,1:3). Observe that each one of the lines in V contains the coordinates of a unique vertex of the form [x,y,z], obtained from the original STL file, therefore each line is unique.

This helps in separating the geometrical information V, from the topological information T. Below is the standard structure of an STL file, for the structure on the triangulation we refer to [5].

solid *name*

facet normal n_i n_j n_k +
  outer loop
    vertex $v1_x$ $v1_y$ $v1_z$
    vertex $v2_x$ $v2_y$ $v2_z$
    vertex $v3_x$ $v3_y$ $v3_z$
  endloop
endfacet

endsolid *name*

Figure 3 - Standard structure of an STL file.

3 SLICING PROCESS

For the slicing process we assume that a triangulation, i.e, an arbitrary set V (of vertices), an arbitrary set T (of triangles) and three maps

a,b,c:T----->V,

is being given, with the property that for each v in V, the collection of elements t in T for which either a(t)=v, or b(t)=v or even c(t)=v, is finite and can be ordered in such a way that t and tô share an adjacent edge every time t is the successor of tô or tô is the successor of t. This is called the star-neighborhood (see Figure 4)

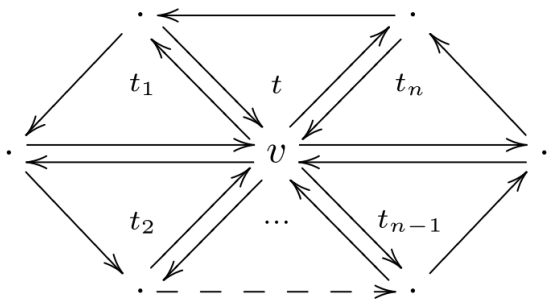


Figure 4 - Example of a star neigbourhood of a vertice in a triangulation (T,V,a,b,c).

(or umbrella) of v and guarantees that the structure (T,V,a,b,c) is the result of a triangulated 2-manifold (a surface) and it is also used in the procedure of finding the contour level for a given z-coordinate. Details on this step can be found in [2].

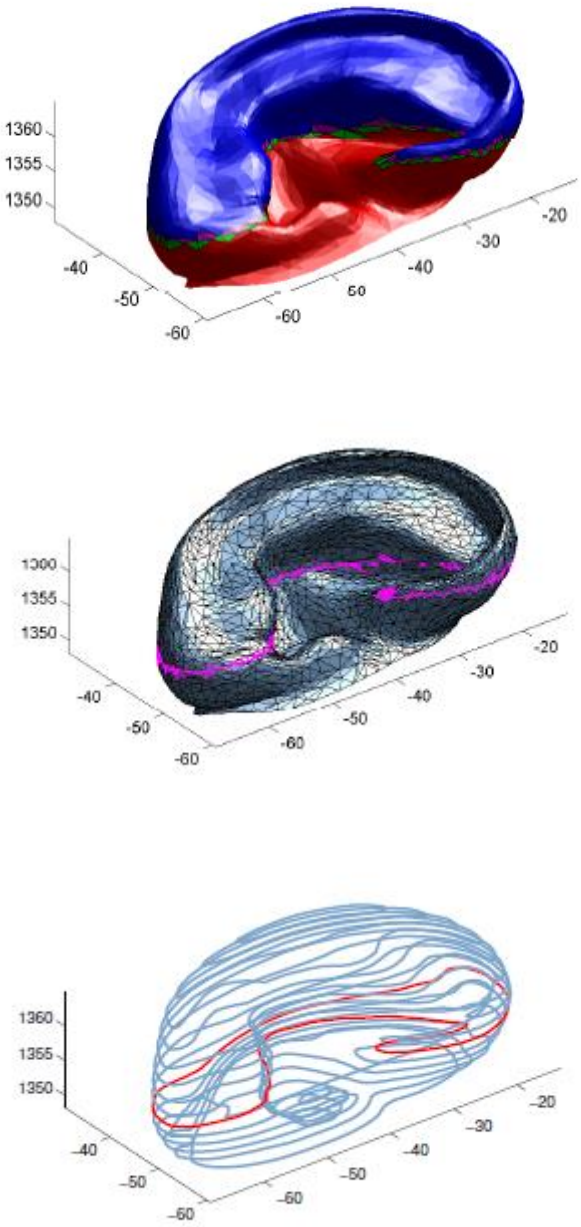


Figure 5 - An example of a triangulation with the triangles above of a given slicing plane coloured in a different color, together with contour levels.

Analysing in detail the procedure outlined in [2] we observe that the specification of the z-coordinate fixing the horizontal plane, z=z0, that intersects the 3-D triangulated body is only used in order to decide which vertices in V have z-coordinate above, under, or are lying in the plane z=z0 (see Figure 5). From this point on, the process is purely topological. This suggests the possibility to consider a slicing of the triangulated 3-D body, not necessarily by planes par-

allel to the xy-plane but rather arbitrary surfaces, as soon as they do not overlap. Observe that whatever we are saying with respect to the xy-plane can be said with respect to any other direction, by performing a simple rotation on the coordinate system. Furthermore, this will allow for instance to partition the 3-D space, not in a collection of horizontal planes, but into a collection of vertical cylinders or concentric spheres. As a result one would be able to do rapid prototyping and rapid manufacturing with systems not only in cartesian coordinates, but also in polar, cylindrical, or spherical coordinate systems.

4 SCANNING PROCESS

As explained [2] and [1], see also [3], the resulting structure of slicing a triangulation is a directed graph with a symmetry, that is a set B of points, a set A of edges connecting those points in a specifyed direction, d,c:A---->B, with d for domain or initial point and c for codomain or end point, together with a map phi:A--->A, specifying a successor for each one of the edges in A. The geometrical information is kept via an embeding B---->R^{3}, of B into the three dimensional space, induced by the fact that a point in B is the result of an intersection from a z-plane with an edge of some triangle obtained during the slicing process, hence it is equiped with geometrical information.

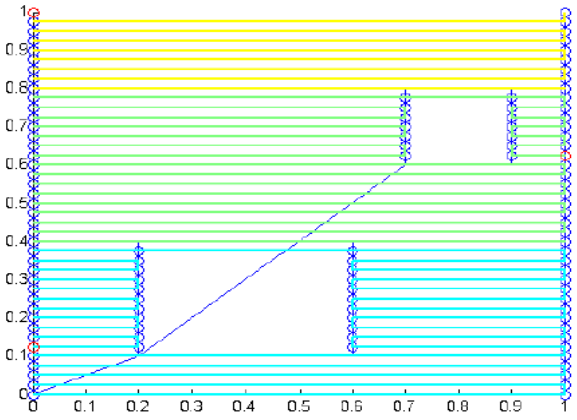


Figure 6 - An example of a toolpath generated scan with the zig-zag strategy, with discontinuities.

The only restriction on the structure of a graph with a symmetry (A,B,d,c,phi) is that phi must be a bijective map and d composed with phi is c, in other words, each edge has a unique successor. This is used in the process of calculating the zig-zag toolpath as detailed in [1]. Briefly, given (A,B,d,c,phi), together with a specified spacing h between parallel zig-zag trajectory lines to be connected, we consider the new structure (E,r,s,g) in which E is the set of points in the intersection of the specified parallel zig-zag lines and the contour resulting from the embedding of the symmetric graph into the 3-D space, as illustrated in Figure 7, with the following interpretation. The segments marked with g are connecting points crossing the interior of the region defined by the embedded contour, while r and s are connecting the points that are alternating at the boundary of the region. A finite state automaton (Figure 8) is used to describe the zig-zag strategy for the construction of the toolpath. Some variations are possible at this point, namely in the way of how to handle possible discontinuities that may occur in the process (se Figure 6). In a near future we also plan to develop other scanning strategies such as the homotopic one (the contour is homeomorphically contracted onto a single point), or the space filling curves strategy (such as Peano curves).

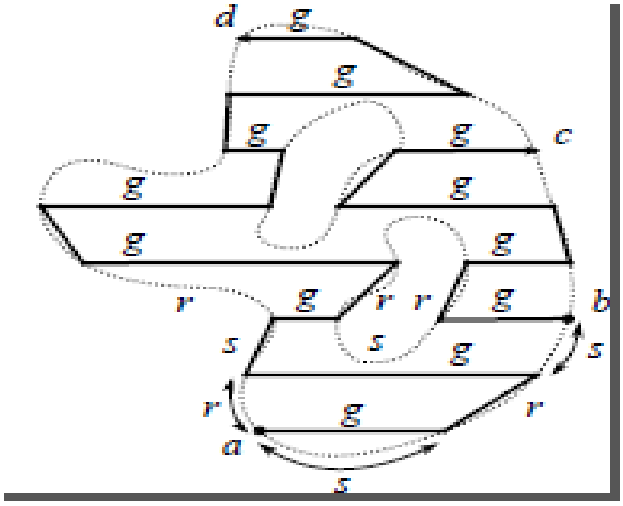


Figure 7 - An example of the endomap structure (E,r,s,g).

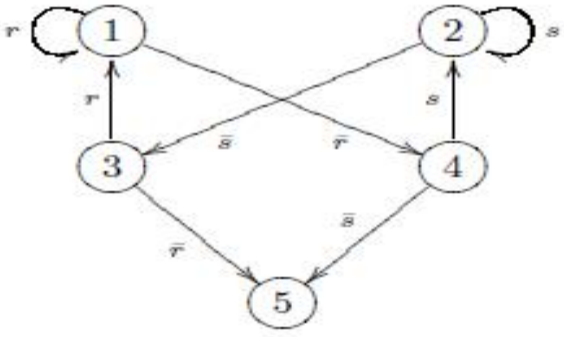


Figure 8 - Automaton used in the zig-zag strategy.

5 PRINTING PROCESS

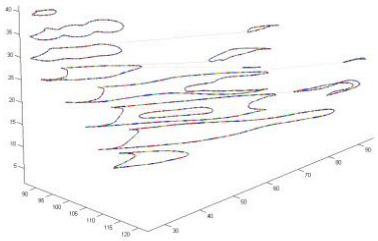
The printing process consists in generating the (specific in each case) machine code that converts the calculated toolpaths by choosen scanning strategy into impulses to be sent to the controllers in the printing system. We have already several different tools for communicating with printing machines such as BioCell, BioExtruder, DualBio, etc., developed at CDRSP.

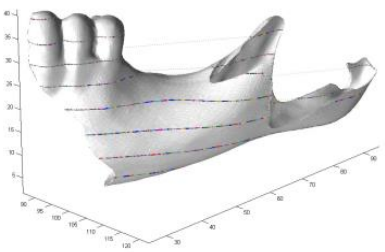
There are several issues that can be considered at this point. For instance the velocity of deposition of the material, if the machine has the ability to stop extruding material at each time a discontinuity is encountered on the precalculated toolpath, etc.

Some of the functionalities are already implemented while others are still to be developed in future work.

6 EXAMPLES

As an example we display some images obtained with the software in generating the slicing, scanning and toolpath of some applications. Our solution is based in a purely topological procedure. It uses a functional programming paradigm which enable us, on the one hand, to implement the solution in a highlevel computational system (suchasMatlab), while on the other hand, it explores the great capabilities of the modern personal computers such as time processing or working memory.





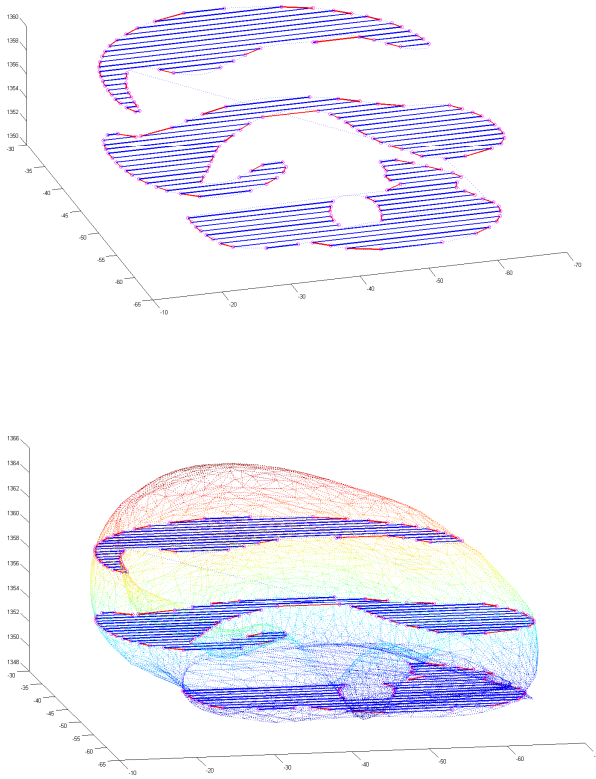


Figure 9 - An example of a triangulation with the triangles above of a given slicing plane coloured in a different color, together with contour levels.

7 REFERENCES

[1] Gaspar, Miguel B. 2011. "Sobre estratégias de varrimento contínuo para o preenchimento de curvas planas", Boletim da SPM - Encontro Nacional da SPM 2010, 0: 72 - 78.

[2] Gaspar, M.B, Martins-Ferreira, N. ï ñRobust STL processing for extrusion-based manufacturingò, Innovative Developments in Virtual and Physical Prototyping, ed. P. Bartolo et al., London: Taylor and Francis, 2012, 273 ï 279 (ISBN: 978-0-415-68418-7)

[3] Gaspar, Miguel B; Martins-Ferreira, N.. 2012. "A procedure for computing the symmetric difference of regions defined by polygonal curves", In Journal of Symbolic Computation (submitted).

[4] Innovative Developments in Virtual and Physical Prototyping: Proceedings of the 5th International Conference on Advanced Research in Virtual and Rapid Prototyping, Leiria, Portugal, 28 September - 1 October, 2011

[5] Gaspar, M.B, *BioFab Toolbox*, Master Thesis, IPL, 2010.