Design for 4D printing: Modeling and computation of smart materials distributions

Germain Sossou, Frédéric Demoly, Hadrien Belkebir, H. Jerry Qi, Samuel Gomes, Ghislain Montavon

Version of Record: https://www.sciencedirect.com/science/article/pii/S026412751930512X
Manuscript_38ef11c61c18e235e120bd1ab45195ee

Design for 4D printing: Modeling and Computation of Smart Materials Distributions

Germain Sossou[1], Frédéric Demoly*[1], Hadrien Belkebir[1], H. Jerry Qi[2], Samuel Gomes[1], Ghislain Montavon[1]

[1] ICB UMR 6303 CNRS, Univ. Bourgogne Franche-Comté, UTBM, 90010 Belfort, France
[2] G.W.W. School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

*corresponding author.

Abstract

The material complexity allowed by additive manufacturing (AM) has made smart materials (SMs) processing easier than usually, giving birth to the so called 4D printing (4DP). This has expanded further the design space around AM. Yet, for this design space to be embraced by designers, there is the need to make SMs modeling and simulation easier, especially in conceptual design. Previously, efforts have been dedicated to a voxel-based modeling and simulation framework for SMs showing how important are material distributions (MDs) when it comes to design for 4DP. Here, a twofold contribution is made to design for 4DP. First a computational tool embodying the previously developed theoretical framework is introduced. This tool – VoxSmart – harnesses the power and the convenience of the graphical algorithm editor Grasshopper® within the CAD software Rhinoceros® to SMs modeling and simulation. The tool basically allows for an easy simulation of any MD. Given a source shape and a target shape of the same part, a set of materials (conventional/smart) and a stimulus, finding the right MD that yields the appropriate transformation upon exposure to the stimulus is quite challenging. This is the core of the second contribution. An adaptive compliant wing of a micro unmanned aerial vehicle is presented as case study.

1. Introduction

In the wake of the emergence of additive manufacturing (AM) from a rapid prototyping process to a manufacturing process, there is another breakthrough in manufacturing: 4D printing (4DP). This involves the use of smart materials (SMs) and possibly conventional materials as "inks" for 3D printing. Owing to the stimuli-responsiveness of the SMs, the so printed item is imbued with the ability to change, hence its 4th dimension: time. Literature has highlighted 4D printed parts which react to a wide spectrum of stimuli such as heat [1-5], electricity [6], moisture [7], light and magnetic field [8] by changing shape and others whose color changes in response to heat (as shown in Fig.1(b)) or when strained [9]. Currently 4DP is rightly the subject of huge research efforts [10, 11] regarding the manufacturing aspects (exploration of new AM processes and materials, characterization, etc.). However very little work is done to support designers in considering 4DP in their concepts.

In 4DP, through the use of SMs as raw materials in AM, what is sought is basically to embed a smart behavior within a structure. This is done in such a way that the material is/becomes the mechanism [12], and a passive source of energy (available in the environment or supplied internally) is what moves the mechanism to produce the desired/designed behavior. Questions that may then arise include: can one single SM be sufficient to produce a desired behavior? If no, what other materials should be combined to it? How can SMs be 'mixed' to produce a behavior? Etc.

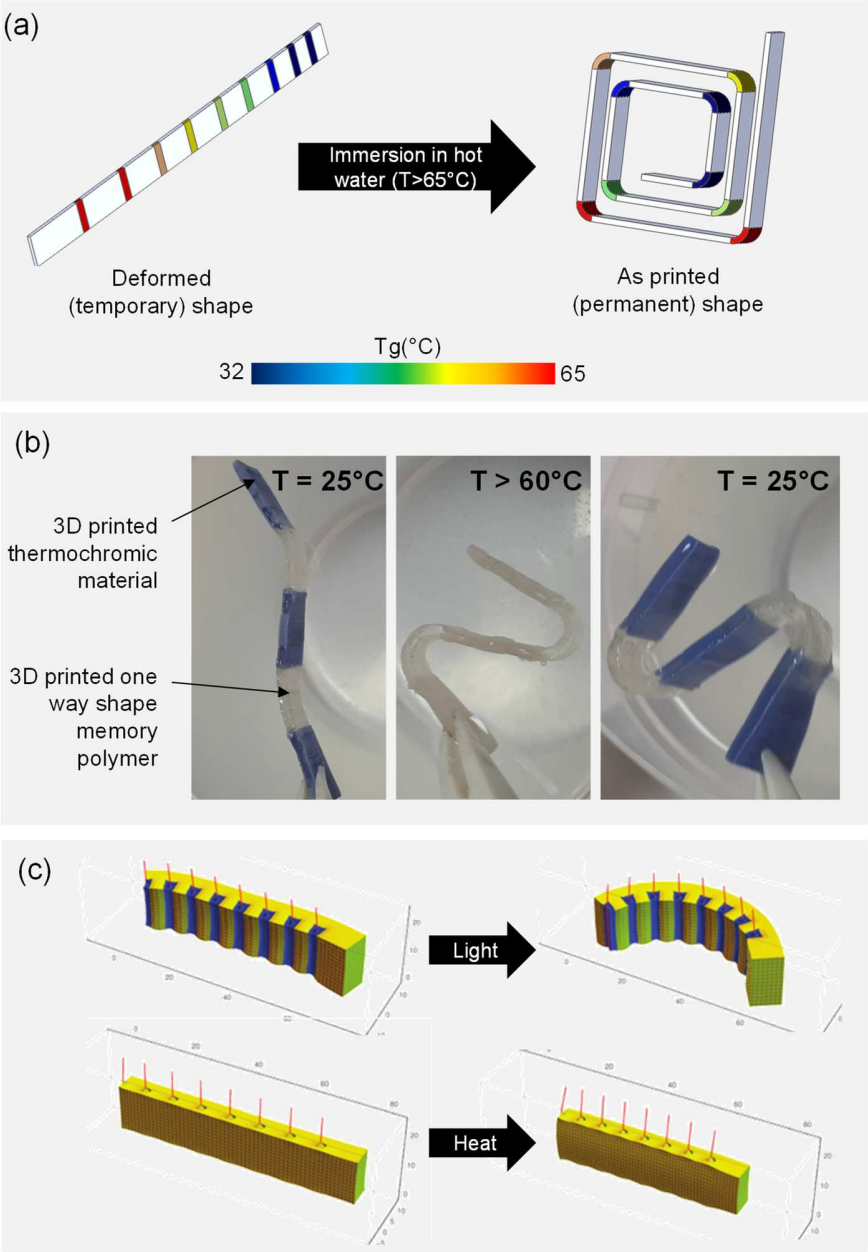


Fig.1 - Functional smart material distributions: (a) shape memory polymers distribution triggering a sequential behavior (adapted from [3]) – (b) 3D printed distribution of thermochromic polymer and shape memory polymer – (c) Hypothetical distribution combining heat-responsive hydrogel and photo-sensible fibers (adapted from [13])

Answers to these questions may be found by analyzing the following examples. The helical part presented in [3, 14] is made of an inert polymer material and many other shape memory polymers (SMPs), which differ by their glass transition temperature (Tg), a specific temperature at which shape recovery is triggered. In this part, depicted in Fig.1(a), the material distribution (MD) is such that the part behavior is sequential: only the hinges are made of SMP, in such a way that their glass transition temperatures increase outwards. This MD is such that – at a temperature higher than all the Tg – the hinges' shape recoveries (from the deformed straight shape to the permanent bent shape) occur sequentially, ensuring thus a successful shape recovery of the whole part. Another example, shown in Fig.1(b), is a part printed

in our laboratory with a SMP and a thermochromic material. Using the environment's medium (water) as heat provider, the MD is such that the thermochromic material turns totally white when the shape recovery of the SMP sections is complete, and it turns back to blue when cold indicating that the SMP material is rigid again. One more example whose MD is finer, more intricate and less intuitive than the previously described, is the one presented in [13], and depicted in Fig.1(c). This hypothetical part combines heat-responsive hydrogel and photo-sensible fibers. When the fibers' material is uniformly distributed in the gel, heat and light all have the effect of uniformly contracting (no bending) the gel, but when the fibers are used and the composite is not fixed on a surface, it contracts like an accordion under the effect of heat and bends when illuminated. These examples have prompted us to posit the idea that a 4D printed item owes its functionality mainly to the specific MD that it is made of; in other words one MD equates to one concept. MD is then of high importance when it comes to design the material-is-the-mechanism [12] like structures. This still holds even when SMs are not involved: a specific distribution of the same (conventional) material with void can yield unconventional behavior, which is well demonstrated by the so-called metamaterials [15, 16]. Such SMs-based examples support the idea that SMs combined with other materials lead to smart behaviors more complex than when taken alone. In a nutshell, considering particularly the case shown in Fig.1(c), designing the right distribution that can provide precisely controlled motion and shape change may be a challenging task. Paramount to designing for 4DP is the design of a MD made of smart and conventional materials. Designers, which are more likely to find innovative 4DP-based concepts, may not be enough equipped to embrace this new design freedom.

One of the milestones in the road towards 4DP's adoption among designers is the capability to rapidly model and simulate any MD, especially in the conceptual design phase. The main reason behind this need for a rapid evaluation is that a MD equates a concept as shown in the above discussion. Current design tools that can be deemed as dedicated to 4DP do not support this vision. Most of these tools seem to be dedicated at the simulation and design of origami structures, a situation fully encapsulated by the statement: "Specialized software is needed to design specific folds, creases, and patterns, and to demonstrate the sequence of folding." [17]. While we acknowledge that origami design [18] is quite a challenging task, these structures do only represent a niche of 4DP. In addition, in the case of 4DP-based origamis, even when the way an origami would fold is fully designed, issues related to how material would be distributed to achieve its folding sequence are still to be solved. We posit that designing a MD for an application can be made through any of the following manners:

- Distributions reuse: using a predefined distribution (whose behavior leads to the desired state change) and apply it to one of the states;
- *3D painting*: an explorative approach consisting in patterning freely the materials and simulating the so obtained distribution until the outcome meets the desired change;
- Distribution computation: in a topology optimization-like [19] manner, computing the appropriate MD to achieve an expected change given the available materials.

Here the attention is focused on the last two manners, especially for the case of SMs whose response induces a change of shape. As the existing approaches for determining SMs distribution are rather *ad hoc* (except a few such as [20]), this research work aims to provide a methodological approach to this problem along with a supporting computational tool to be used in the early phases of the product design process. In an initial research effort dedicated to Design for 4D printing (DF4DP) [21], we introduced a voxel-based theoretical framework for modeling and simulating SMs. The modeling scheme has been demonstrated to be able to simulate the behavior of conventional materials and shape changing SMs. Theses SMs include piezoelectric material, electro-/magneto-/photostrictive materials and hydrogels. Building on this research initiative, we introduce a design tool – called VoxSmart – embodying the proposed framework. It is a tool

based on the graphical algorithm editor Grasshopper® (GH), an add-on of the commercial CAD software Rhinoceros© (RH). The purpose is to make the modeling and simulation scheme usable in a form that benefits a large spectrum of designers. Furthermore and most importantly a methodology is provided to compute MDs based on the modeling framework. As such the reader is advised to refer to [21] for the theoretical foundations behind VoxSmart and the proposed MD computation engine.

2. Introduction to VoxSmart

The proposed modeling scheme introduced in [21] for SMs simulation has been made tangible by a computational design tool. The framework has been implemented in the RH [22] add-on Grasshopper® (GH). RH is a CAD explicit modeling software allowing a seamless design of complex shapes. GH is a graphical algorithm editor (without the need of any script) that allows form generation (in RH environment) and virtually any computation. A graphical algorithm is typically a collection of components (running each a computation) connected by wires, which are the data flowing through them. What makes the GH computation engine virtually infinitely expandable is the possibility to develop plugins for specific tasks (e.g. design, simulation, even manufacturing control). There is a rapidly growing GH community of users (namely designers) and plugins developers. The shape complexity allowed by RH and GH has prompted us to develop a GH plugin for materializing our SMs modeling and simulation scheme: VoxSmart; the plugin has been scripted in the programming language C#.

2.1 Components

Fig. 2 - The VoxSmart plugin components

VoxSmart includes six categories of components: Voxel Edition, Material Edition, Boundary Conditions Definition, Stimulus Definition, Simulation and Distribution Computation, as shown in Fig. 2. The Voxel Edition category gathers components that are used to construct the voxel model of the object (which can be imported from RH – which reads most of the CAD format such as STEP or STL – or generated in GH) to be simulated. A voxel model can be constructed by specifying an origin and voxels' counts along x, y and z or by voxelizing an input geometry (as illustrated in Fig. 3); in both cases a voxel size (which can be seen as a resolution) must be specified. A voxel object can also be edited by adding/removing voxels and extruding one of the object's faces along any of the positive or negative x, y and z directions. In the Material Edition category, there are components for creating a conventional material (defined with a Young and a shear moduli), components for creating SMs and components for assigning these materials to specific regions within the voxelized object. The Stimulus Definition category gathers components for defining stimuli including: heat, electric field, magnetic field, and light. More details can be found about the major components in the appendix.

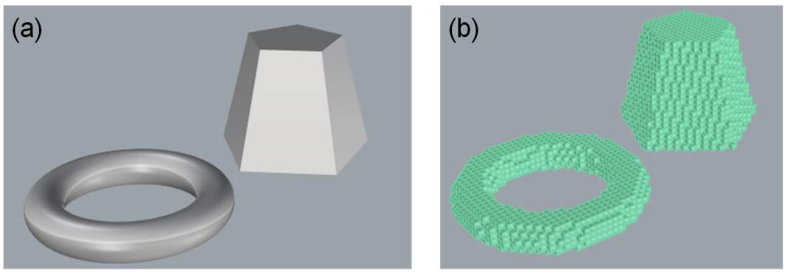


Fig. 3 - Voxelization: (a) Initial shapes - (b) Voxelized shapes with VoxSmart.

2.2 **Workflow of a VoxSmart definition**

The typical workflow for a VoxSmart definition is shown in Fig. 4. Groups of components have been highlighted (purple background) for the sake of explanation. The definition is the one that has been used to simulate the smart hydrogel valve printed in [5] and whose modeling and simulation with the proposed modeling scheme have been presented in [21]. First the object's geometry is defined (Fig. 4(a)) with native GH components (it could have been imported as well), then the geometry is voxelized (Fig. 4(b)). To avoid any simulation error related to voxel without any material, components for creating a voxel model require as input a material that is initially assigned to all the voxel. Subsequent material assignments to a voxel simply overwrite the initially assigned material. In a third step SMs are assigned (Fig. 4(c)) to regions of the voxelized object; these regions are specified as other geometries that can be defined within GH or in RH. Then boundary conditions are applied to the model. Finally, after specifying the stimulus the SM is sensitive to, the model is simulated.

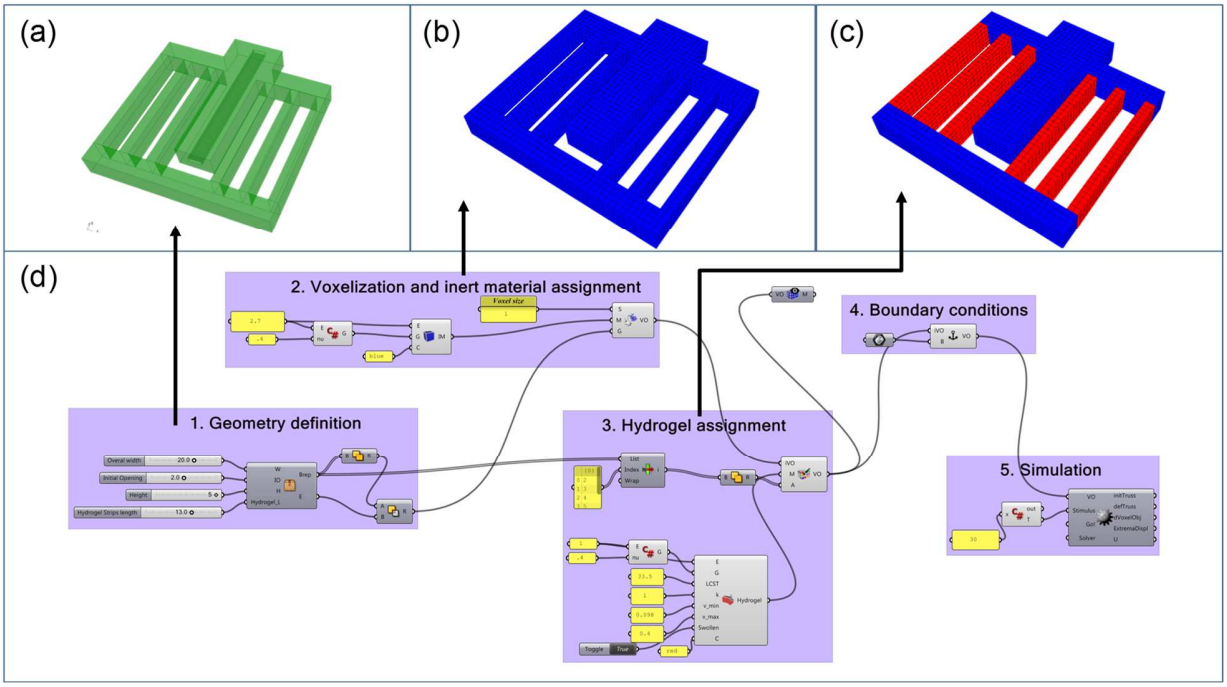


Fig. 4 - Typical VoxSmart workflow: (a) geometry definition – (b) Voxelized geometry with a homogeneous (inert) material distribution – (c) Voxelized model with a heterogeneous material distribution including the hydrogel actuating sections – (d) The whole modeling and simulation GH definition.

3. Distributions computation

With the voxel-based SMs simulation engine in place, along with the embodying tool VoxSmart, a step towards empowering designers to rapidly simulate a SM-based object has been taken. This encompasses the explorative approach of designing with SMs, the approach we termed as *3D painting* where the possibility is given to freely pattern any MD in a geometry and get how such geometry would behave. However, when a specific shape change is needed for an application, finding a right distribution achieving it may not be intuitive especially when one has limited knowledge of SMs and what strain mismatch between two dissimilar materials can lead to. Some kind of automation may then ease the design problem by at least providing a starting point, and thus by saving multiple iterations time. This is the purpose of this section.

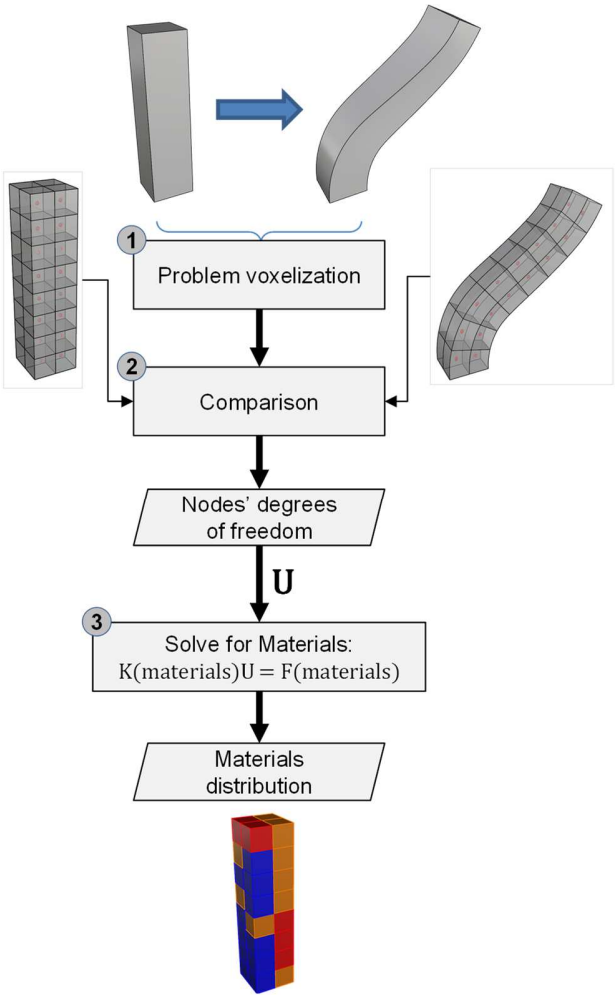


Fig. 5 - Methodology for materials distribution computation.

3.1 Problem formulation

Roughly speaking the problem of computing a distribution may be formulated as how can the SMs be spatially mixed (with the granularity allowed by the voxel-based model) to an inert (conventional) material in a source shape, so that this latter – upon exposure to the stimulus – deforms into a target shape?

More specifically the inputs of the problem are:

1. A predefined source shape S.
2. A predefined target shape T. This shape must be topologically equivalent to S. To ensure such equivalence, T should ideally be obtained by a combination of topology preserving transformations applied to S.
3. An inert conventional material that S is originally totally made of.
4. A finite number of different non-programmable shape changing SMs[1] of the same type (e.g. piezoelectric materials with different d_{33} coefficients).
5. A stimulus state which is the stimulus the specified SMs are sensitive to. "State" here refers to how the stimulus field has to be at the moment the object is expected to be of shape T.

The solution to the problem yields a spatial arrangement of the SMs within shape S. Referring to our modeling scheme this will be a distribution of the different materials making up the whole shape.

3.2 Method for computing a distribution

The solution to the problem described above is found through three steps (as shown in Fig. 5) which are delineated in the following paragraphs.

3.2.1. Problem voxelization

First of all, the source shape S is voxelized with a chosen resolution. In addition to the generated voxels (which are all in the form of a simple box mesh), two sets of geometrical entities are also to generate:

- The underlying frame's nodes: these are voxels' centers in the (non-deformed) shape S. These will be used to compute the nodes displacements;
- For each voxel, a set of two unit lines that are initially aligned with the *y* and *z* direction. The voxel's center along with these two lines define a plane that will be used to record the rotation degrees of freedom (DOFs), as illustrated in Fig. 6.

The so derived voxelized source shape S is then morphed into the desired target shape yielding a voxelized target shape T, in which individual meshes (the voxels) have been deformed and the sets of points and lines have been moved to different locations. The lines are also morphed into curves. This morphing process is manually done, using GH components that allow for an accurate morphing of an object. Nevertheless the morphing can be done otherwise, with another software.

[1] As explained in [21], such materials are all the shape changing SMs except the shape memory materials.

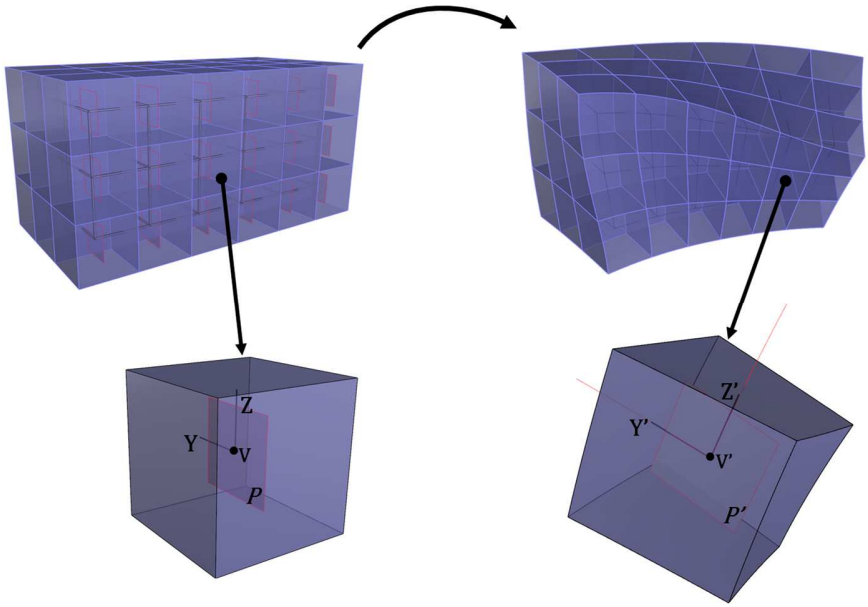


Fig. 6 - Rotational DOFs' tracking between source and target shapes.

3.2.2. **Comparison**

A comparison is made between the voxelized source shape and the target one in order to get the underlying frame nodes' DOFs (the **U** vector as introduced in Equation 6 of [21]). The required displacements are readily computed by finding the vectors translating the voxels' centers. Getting the nodes rotational DOFs is made by planes comparison. An initial plane P is generated at the center of a voxel (denoted V) using this latter and the unit lines (aligned along *y* and *z)* introduced in the problem voxelization. These lines will be denoted Y and Z. In the deformed state, the voxel center is moved to V' and Y and Z are morphed into curves. Let Y' and Z' denote the tangent to these curves at V' (see Fig. 6). Here V', Y' and Z' are used to generate the deformed plane P'. P and P' are then moved so that they intersect. Using quaternions operations, the rotations about *x*, *y*, and *z* that lead from P to P' are computed and these are taken as the required rotational DOFs.

At the end of this stage, all the frame's DOFs required for the shape change S → T are known and are stored in a vector $\boldsymbol{U_{target}} = [u_{x_1}, u_{y_1}, u_{z_1}, \theta_{x_1}, \theta_{y_1}, \theta_{z_1}, u_{x_2}, u_{y_2}, \cdots]$.

3.2.3. **Material distribution computation**

The distribution is to be derived as a solution to an inverse problem. In the forward scheme, we go from a MD making up the initial shape, stimulus and possibly loads are applied and then the global stiffness equation ($K\boldsymbol{U} = F$) is solved to get the deformation. In this problem what is sought is somehow the opposite: the deformation ($\boldsymbol{U_{target}}$) is given as known quantities (based on source and target shape) and given a stimulus, we determine the material properties of each voxel.

More specifically, the goal is to find a MD that minimizes the difference between the DOFs $\boldsymbol{U_{target}}$ and the DOFs $\boldsymbol{u}$ computed (for the same voxelized object) with that distribution. Again, in our problem formulation any voxel's material is chosen from a set of materials including one conventional material and a finite number of other SMs of the same type. As such a MD can be represented as an array of integers indicating which materials are the voxels made of. This is illustrated in Fig. 7 with a set of two materials.

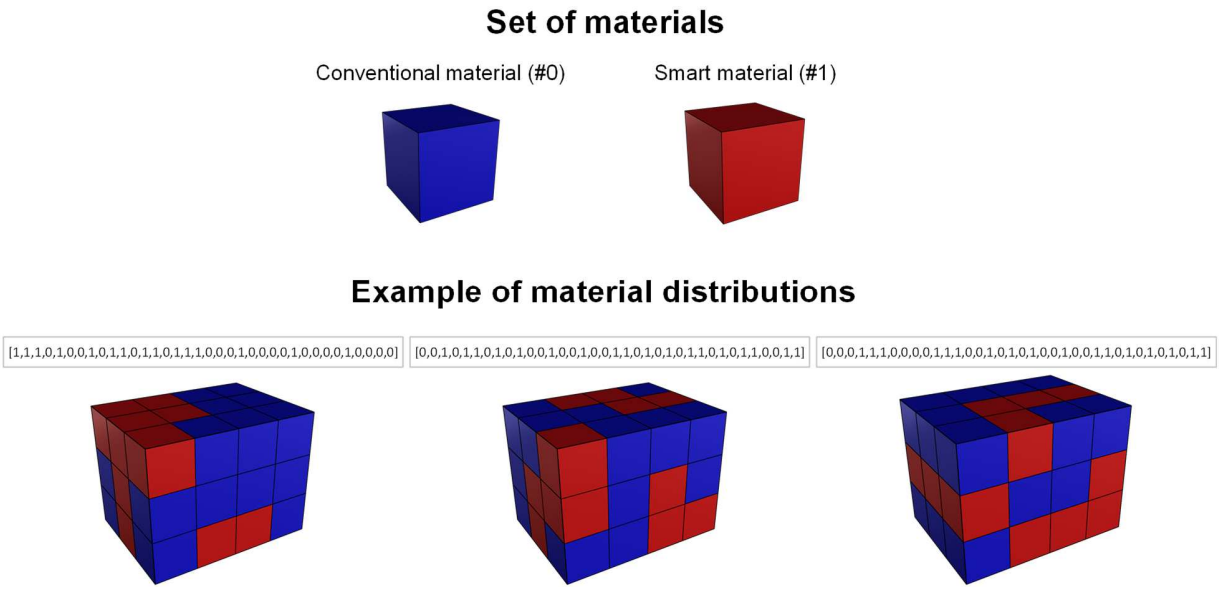


Fig. 7 - Illustration of material distribution representation for a 4 x 3 x 3 voxels object. The smart material count is limited to one.

In such setting, the material properties are not allowed to vary continuously; the problem is actually an integer-constrained optimization problem. Besides the objective (which is a scalar measure of the difference between desired DOFs and the computed ones) is not a linear function of the MD. These conditions preclude any gradient-based optimization methods from solving the problem efficiently. We then elected to use a stochastic method, namely genetic algorithm (GA) to solve the problem. GA is indeed efficient at solving such nonlinear integer-constrained problems (as demonstrated for instance in [23]).

Using the GA terminology, an individual (or a genome) is a MD (as shown in Fig. 7) whose genes are the entries of the array representation. As fitness function we used the squared norm of the vector formed by the difference between $\boldsymbol{U_{target}}$ and $\boldsymbol{u}(individual)$: $\boldsymbol{fitness} = \left\|\boldsymbol{U_{target}} - \boldsymbol{u}(individual)\right\|^2$.

On a computation aspect, the power of the Matlab® [24] GA tool box has been harnessed to solve the problem. A semi-automatic interoperability between VoxSmart and Matlab® has been created. Basically a VoxSmart component has been developed to handle the input problem: source shape, target shape, set of materials, boundary conditions and stimulus state. The component then computes $\boldsymbol{U_{target}}$ and stores it, along with the other input, in a matrix form. All the constructed matrices are finally packed in a .mat file, which is subsequently loaded in Matlab®. A script has then been written in Matlab® to handle the input problem and run the optimization problem with the built-in GA toolbox.

4. Illustrations

4.1 Verification case

The accuracy of the proposed scheme to find a known distribution has first been gaged. Basically DOFs computed for the deformation of a known distribution have been fed into the algorithm as $\boldsymbol{U_{target}}$. The

case is the one shown in section 4.2 of the first paper [21], for the hydrogel actuator. The actuator's distribution was made of a conventional material and hydrogel. The number of generations was limited to 150 and for each generation crossover rate was set to 0.94 and elites count (number of best individuals of a generation that survive till the next generation) was set to 10 (which is a pretty good figure for 100 individual generations). Only two candidate materials (conventional and hydrogel) were given as input to the algorithm.

The results for the fitness function over the 150 generations are shown in Fig. 8. The determined MD and the original one are shown in Fig. 9(a). The deformations resulting from both are shown in Fig. 9(b). The fittest individual scores 9.5, which is higher than the theoretical best fitness: 0. Nevertheless, on the one hand (the square root of) this residual is to be distributed over the 2258 DOFs making up the whole problem, which is not significant: average of 0.064 mm of difference with the right displacement and average 3.7 degree of difference with the right rotation. On the other hand, deformation computed for the found optimum is very close to the desired deformation as shown in Fig. 9. This is an indication that the problem of MD computation is not one with a unique solution. A finding which can be ascertained by the fact that the problem: Find K and F, such that: $KU = F$ does not have a unique solution (consider for instance the simple case: $K \times u = F$, where K, u and F are numbers). Furthermore, it shows that the distribution computation engine is focused on the behavior (the shape change) of the distribution rather than on the distribution itself.

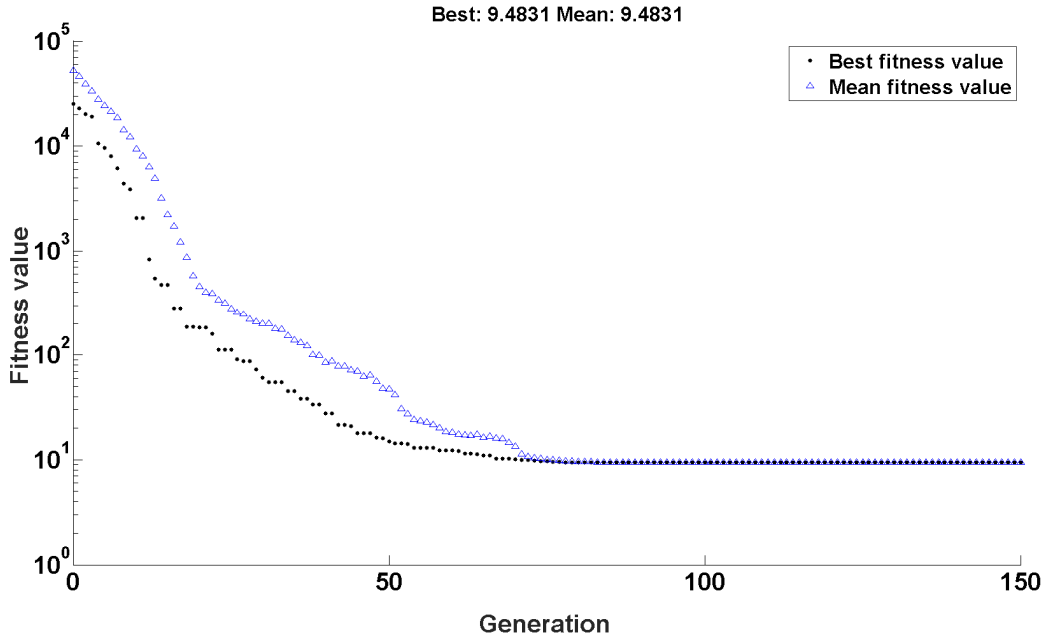


Fig. 8 - Fitness plotting over the generations

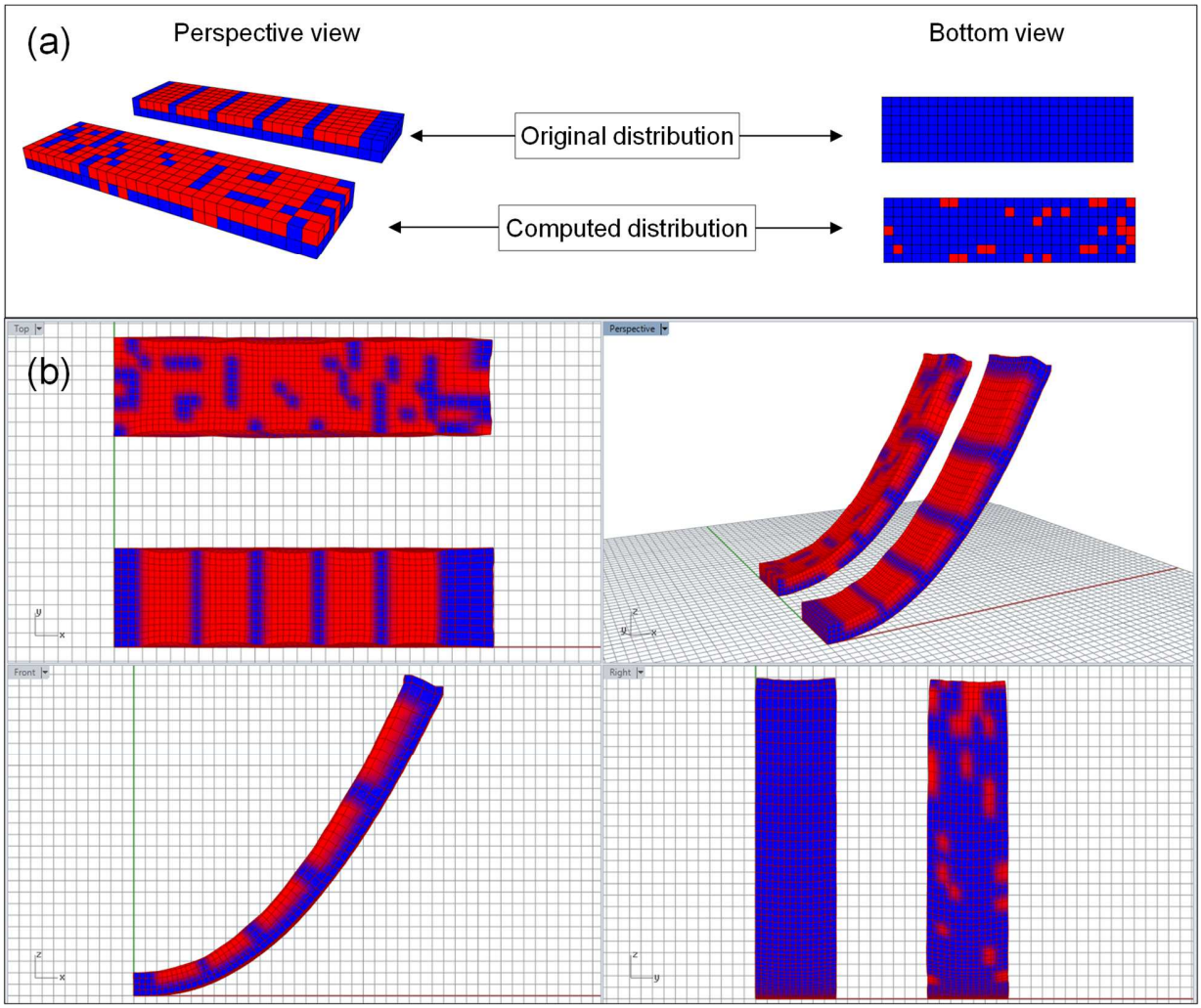


Fig. 9 - (a) Original and computed distributions - (b) Rhinoceros® viewports of the deformations as computed by VoxSmart

It can be stated that the GA method yields pretty reasonable solutions to the MD computation problem.

4.2 Design of an adaptive compliant wing

Here, we consider a more complicated deign case of an aerospace use case, namely an adaptive compliant wing as shown in Fig. 10. In order to prevent a whole wing from stalling at the same time, wings are usually slightly twisted (a feature usually referred to as wing twist). These wings are twisted from root (near the fuselage) to tip. In most case this twist is downward from root to tip. In such configuration, angle of incidence at the tip is always lower than the one at the root, thus tip – where most of the control surfaces (especially those responsible of pitch) are located – stalls quite later than root allowing the pilot to adapt the pitch angle before the wing totally stalls. Wing twist is conventionally a fixed feature. Flight phases such as take-off are where stalling is most likely to occur, while in cruise phases the probability of such event is pretty lower. Therefore, not all the lift that can be generated by a wing is achieved with a fixed twist wing. With a wing that can be twisted at will (e.g. at take-off) and let straight at cruise for instance, there could be a higher lift at cruise and hence less energy consumption. Such adaptive compliant wings are conventionally manufactured with an underlying actuating truss which morphs the

wing's skin. An alternative that could be made possible by 4DP is a bulk wing made of (light) conventional and smart materials in such a way that *the material is the mechanism* [12].

We elected to consider the case of a single spar wing. The wing span is of 200mm, typical of a micro unmanned aerial vehicle (MUAV). The problem has been formulated with as source shape the straight wing and target shape the twisted wing as presented in Fig. 10. The spar crosses all the vertical sections' centroids, and the so defined axis is used as the twist axis.

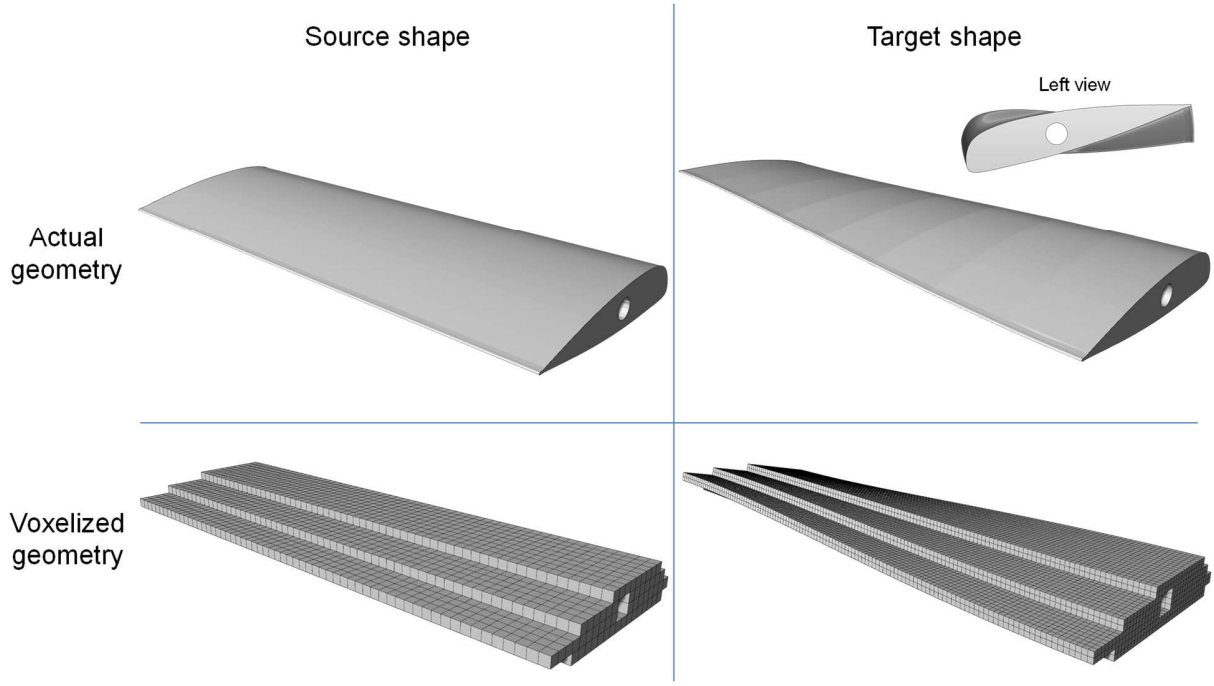


Fig. 10 - Wing twist problem formulation

As SMs we considered polymeric magnetostrictive materials. These are obtained in composite form [25] with magnetostrictive fillers dispersed in a polymeric matrix. A carbonyl iron/silicone [26] composite has been demonstrate to exhibit a strain of almost 10% at saturation. We then consider a set of three materials including a hard silicone (as the conventional material) and two hypothetical magnetostrictive composite material; the properties are outlined in Table 1.

Table 1 - Materials properties for the twist wing case

	E (MPa)	G (MPa)	λ_s (%)
#1 Silicone	30	10.07	
#2 Magnetostrictive composite 1	30	10.07	-5
#3 Magnetostrictive composite 2	20	6.7	-10

Fig. 10 presents the source and target shape both in actual geometry and voxelized geometry. The target shape is such that the twist angle is of 15° at the wing tip.

With these settings, the return distribution is shown in Fig. 11(a). All the magnetostrictive materials were considered to be at saturation. The deformed wing with the determined distribution is shown in Fig. 11(b).

With this distribution, the actual twist angle at tip is 12°. While the overall twisted configuration is achieved by this MD, the undulations caused by the local deformations are likely to reduce the airworthiness of the plane. Nevertheless this MD could be used as a starting point to a detailed design of the wing. In addition, in the MD computation problem, constraints may be set to enforce homogeneous regions, and thus to reduce the local deformations.

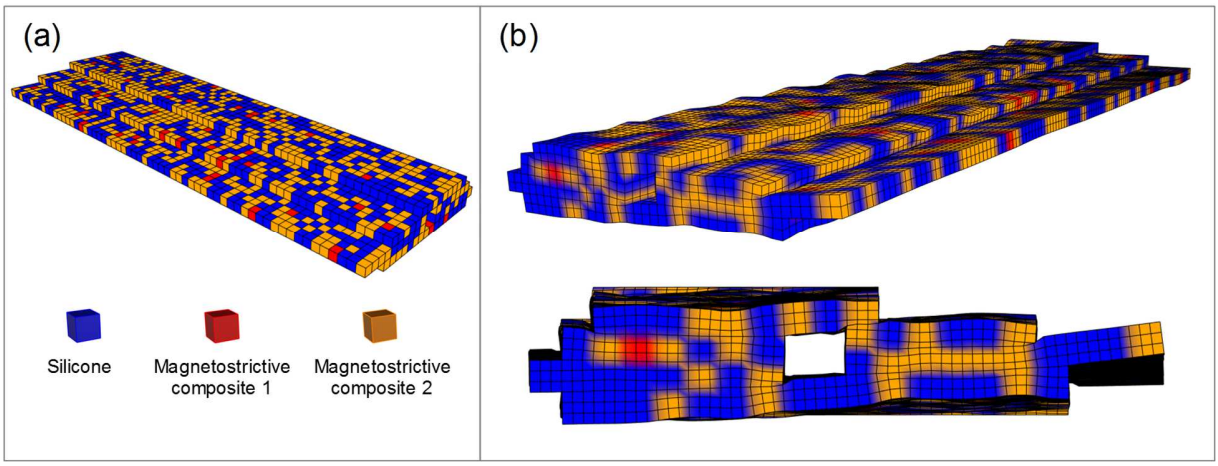


Fig. 11 - (a) Determined distribution – (b) Perspective and side view of the deformed wing when the magnetostrictive materials reach saturation

5. Conclusions and future work

4DP is being given huge research interests regarding its physical aspects (manufacturing, materials, etc.). For ensuring 4DP adoption and implementation in industry, research efforts regarding design aspects as well must be invested. In this article a contribution in that regards has been made; it is intended to empower designers when designing for 4DP. Our voxel-based SMs modeling and simulation framework – as introduced in [21] – has been materialized by a GH plugin named VoxSmart. This tool allows for an easy definition and simulation of any MD on a voxel basis. As such any heterogeneous object with a MD including SMs and conventional materials can be simulated with the tool. Main questions of interest to those interested in designing or printing a 4DP object include (but are not limited to): how a given MD of the part would behave? Which MD for the part would suit a desired post-printing change? With the VoxSmart application, answers to these questions could easily be found. As such, consideration of 4DP by designers is expected to rise. Besides, with the growing Grasshopper® community more innovative 4DP concepts are expected to emerge. Another question that may be of interest to designers is how such derived voxel-based distribution can actually be printed. Technically any additive manufacturing (AM) machine capable of multimaterial printing can print any MD, provided the required materials are available in a form that the machine can process. As shown in [27] many AM techniques are capable of multimaterial printing, be them bespoke or commercially available. Worth highlighting is the PolyJet® technique which has been demonstrated [28, 29] to be able to handle voxel-based MD.

There are nevertheless room for improvement and extension to our proposal, in addition to those mentioned in [21] (effects such as collision detection, friction, gravity, etc.). Regarding distribution computation, the stimulus itself could also be optimized (along with the materials) to achieve the desired target shape, provided that the stimulus is not a design constraint. In the same vein, the way MD is computed can be improved: currently the exploration is made voxel-wise, however a more efficient

approach could be to explore the design space group of voxels-wise, for instance a constraint can be set so that materials must be the same within a 3D Manhattan distance of 2, 3, etc. voxels from specific set points. Voids may also be used in the search for a MD as shown for instance in [30]. Other constraints on materials may also be imposed, such as a conventional material at desired regions.

Acknowledgements

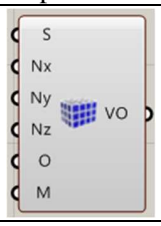
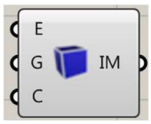
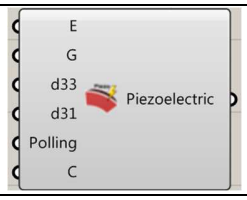
This research activity is part of much larger project in the field of design for 4D printing. The authors would like to thank the Ministère de l'Enseignement Supérieure et de la Recherche, the French "Investissements d'Avenir" program, project ISITE-BFC (contract ANR-15-IDEX-0003) as main financial supports of this research program, and S.mart Franche-Comté network for their participation.

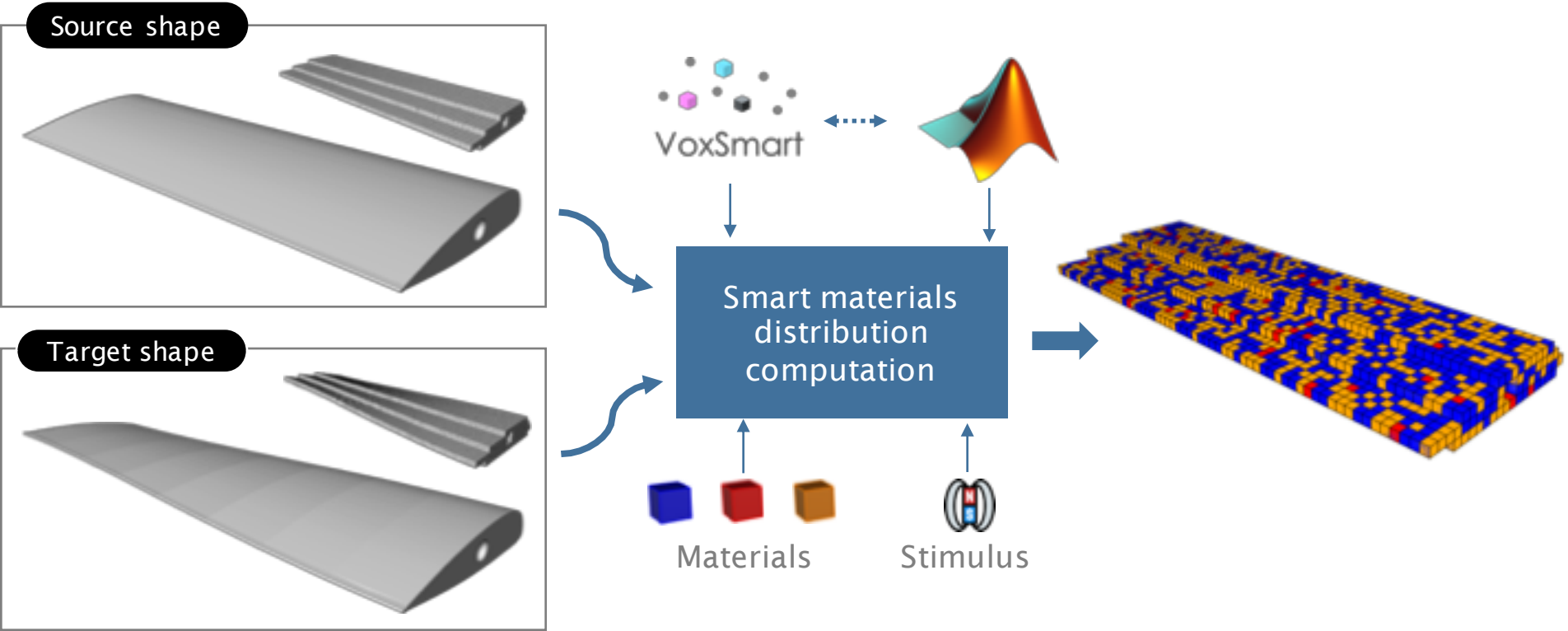
References

1. Ge, Q., H.J. Qi, and M.L. Dunn, *Active materials by four-dimension printing.* Applied Physics Letters, 2013. **103**(13): p. 131901.
2. Zarek, M., et al., *3D Printing of Shape Memory Polymers for Flexible Electronic Devices.* Advanced Materials, 2015: p. n/a-n/a.
3. Mao, Y., et al., *Sequential Self-Folding Structures by 3D Printed Digital Shape Memory Polymers.* Scientific Reports, 2015. **5**: p. 13616.
4. Bodaghi, M., A.R. Damanpack, and W.H. Liao, *Self-expanding/shrinking structures by 4D printing.* Smart Materials and Structures, 2016. **25**(10): p. 105034.
5. Bakarich, S.E., et al., *4D Printing with Mechanically Robust, Thermally Actuating Hydrogels.* Macromolecular Rapid Communications, 2015. **36**(12): p. 1211-1217.
6. Woodward, D.I., et al., *Additively-manufactured piezoelectric devices.* physica status solidi (a), 2015. **212**(10): p. 2107-2113.
7. Raviv, D., et al., *Active Printed Materials for Complex Self-Evolving Deformations.* Scientific Reports, 2014. **4**: p. 7422.
8. Caputo, M.P., et al., *4D printing of net shape parts made from Ni-Mn-Ga magnetic shape-memory alloys.* Additive Manufacturing, 2018. **21**: p. 579-588.
9. Zrinyi, M., J. Rosta, and F. Horkay, *Studies on the swelling and shrinking kinetics of chemically crosslinked disk-shaped poly(vinyl acetate) gels.* Macromolecules, 1993. **26**(12): p. 3097-3102.
10. Momeni, F., et al., *A review of 4D printing.* Materials & Design, 2017. **122**: p. 42-79.
11. Kuang, X., et al., *Advances in 4D Printing: Materials and Applications.* Advanced Functional Materials, 2019. **29**(2).
12. Noumenon. *The material is the mechanism*. 03/21/2018]; Available from: http://noumenon.eu/#&panel1-12.
13. Kuksenok, O. and A.C. Balazs, *Stimuli-responsive behavior of composites integrating thermo-responsive gels with photo-responsive fibers.* Materials Horizons, 2016. **3**(1): p. 53-62.
14. Yu, K., et al., *Controlled Sequential Shape Changing Components by 3D Printing of Shape Memory Polymer Multimaterials.* Procedia IUTAM, 2015. **12**: p. 193-203.
15. Ion, A., et al., *Metamaterial Mechanisms*, in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 2016, ACM: Tokyo, Japan. p. 529-539.
16. Jiang, Y. and Q. Wang, *Highly-stretchable 3D-architected Mechanical Metamaterials.* Sci Rep, 2016. **6**: p. 34147.

17. Pei, E. and G.H. Loh, *Technological considerations for 4D printing: an overview.* Progress in Additive Manufacturing, 2018. **3**(1): p. 95-107.
18. Suh, N.P., *Axiomatic Design: Advances and Applications*. 2001: Oxford University Press.
19. Bendsøe, M.P. and O. Sigmund, *Topology Optimization: Theory, Methods, and Applications*. 2 ed. 2004: Springer-Verlag Berlin Heidelberg. XIV, 370.
20. Maute, K., et al., *Level Set Topology Optimization of Printed Active Composites.* Journal of Mechanical Design, 2015. **137**(11): p. 111402.
21. Sossou, G., et al., *Design for 4D printing: A voxel-based modeling and simulation of smart materials.* Materials & Design, 2019: p. 107798.
22. Associates, R.M., *Rhinoceros*. 2017.
23. Yokota, T., M. Gen, and Y.-X. Li, *Genetic algorithm for non-linear mixed integer programming problems and its applications.* Computers & Industrial Engineering, 1996. **30**(4): p. 905-917.
24. MathWorks, *MATLAB*. www.mathworks.com/products/matlab.html.
25. Elhajjar, R., C.-T. Law, and A. Pegoretti, *Magnetostrictive polymer composites: Recent advances in materials, structures and properties.* Progress in Materials Science, 2018. **97**: p. 204-229.
26. Diguet, G., E. Beaugnon, and J.Y. Cavaillé, *Shape effect in the magnetostriction of ferromagnetic composite.* Journal of Magnetism and Magnetic Materials, 2010. **322**(21): p. 3337-3341.
27. Vaezi, M., et al., *Multiple material additive manufacturing – Part 1: a review.* Virtual and Physical Prototyping, 2013. **8**(1): p. 19-50.
28. Bader, C., et al., *Data-Driven Material Modeling with Functional Advection for 3D Printing of Materially Heterogeneous Objects.* 3D Printing and Additive Manufacturing, 2016. **3**(2): p. 71-79.
29. Vidimče, K., et al., *OpenFab: a programmable pipeline for multi-material fabrication.* ACM Transactions on Graphics, 2013. **32**(4): p. 1.
30. Hiller, J.D. and H. Lipson, *Multi material topological optimization of structures and mechanisms*, in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. 2009, ACM: Montreal, Québec, Canada. p. 1521-1528.

Appendix: presentation of a few VoxSmart components

	Component		Description	Input	Output
Voxel Edition	Cubic object		Generates a voxel object in the form of a regular 3D grid	• S: voxel size • N*i*: number of voxel along the *i* direction • O: origin, this is the position of the voxel with the lowest coordinates values M: material	VO: a voxel object
	Voxelizer		Voxelizes an object of any shape	• S: voxel size • M: material • G: geometry to be voxelized in a Brep form	VO: a voxel object
Materials	Inert material		Generates an inert material object	• E: Young modulus • G: shear modulus • C: color	IM: an inert material object
	Piezoelectric material		Generates a piezoelectric material object	• E: Young modulus • G: shear modulus • d33 and d11: piezoelectric coefficients • Polling: polling direction • C: color	Piezoelectric: a piezoelectric material object
	Material painter		Assigns materials to specific regions	• iVO: a voxel object • M: list of material object • A: list of regions of the voxel object	VO: a voxel object
Boundary conditions	DOFs		Assigns prescribed degrees of freedom (displacement or rotation) to voxels within specific regions	• iVO: a voxel object • D: Prescribed displacement in the form of a vector • R [optional]: Prescribed rotation in the form of a vector • A: list of regions of the voxel object	VO: a voxel object
	Fixing		Fixes voxels within a region. All the degrees of freedom of these voxels are set to zero.	• iVO: a voxel object • B: A region	VO: a voxel object
Distribution computation	Distribution		Handles the distribution computation problem and packs all the input in a file that is read by Matlab	• VO: voxel object in the initial state • Materials • Stimulus • Target: the prescribed DOFs • Path: full path to Matlab file in which all the data are stored.	


Source shape
Target shape
VoxSmart
Smart materials distribution computation
Materials
Stimulus