



HAL
open science

DynBioSketch: A tool for sketching dynamic visual summaries in biology, and its application to infection phenomena

Pauline Olivier, Tara Butler, Pascal Guehl, Jean-Luc Coll, Renaud Chabrier,
Marie-Paule Cani, Pooran Memari

► To cite this version:

Pauline Olivier, Tara Butler, Pascal Guehl, Jean-Luc Coll, Renaud Chabrier, et al.. DynBioSketch: A tool for sketching dynamic visual summaries in biology, and its application to infection phenomena. Computers and Graphics, 2024, 122, pp.103956. 10.1016/j.cag.2024.103956 . hal-04682753

HAL Id: hal-04682753

<https://hal.science/hal-04682753v1>

Submitted on 30 Aug 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DynBioSketch: A tool for sketching dynamic visual summaries in biology, and its application to infection phenomena

Pauline Olivier^{a,*}, Tara Butler^a, Pascal Guehl^a, Jean-Luc Coll^c, Renaud Chabrier^b, Pooran Memari^a, Marie-Paule Cani^a

^aLIX, Ecole Polytechnique, CNRS, IP Paris, Palaiseau, France

^bAtelier Renaud Chabrier

^cUniv. Grenoble Alpes, INSERM, CNRS, Institute of Advanced Biosciences, Grenoble, France

ARTICLE INFO

Keywords:

Sketch-based modeling
Interactive geometric modeling
Sketch-based animation
Narration

ABSTRACT

Having simple methods of illustration is essential to scientific thinking. To complement the abstract sketches regularly used in cell biology, we propose DynBioSketch, an easy-to-use digital modeling and animation tool, enabling biologists to resort to less simplified representations when necessary without having to call professional artists. DynBioSketch is an interactive sketching system dedicated to the design and communication of biological phenomena at the cellular scale that can be illustrated in a few minutes of animation. Our model integrates 3D modeling, pattern-based design of 3D shape distributions, and sketch-based animation. These elements can be combined to create complex scenarios such as the infection phenomenon on which we focus, allowing a narrative design adapted to communication between researchers or educational applications in biology. Our results, along with a user study conducted with biology researchers, highlight the potential of DynBioSketch in enabling the direct design of dynamic visual summaries that convey relevant information, as shown in our infection case study. By bridging the gap between abstract representations used by experts and more illustrative depictions, DynBioSketch opens a new avenue for communicating biological concepts.

1. Introduction

Biological phenomena inherently exhibit complexity, as a series of events or chemical reactions disrupt an initially stable environment, triggering transformations toward the next stable state. Biologists observe and experiment to study the causes and consequences of such events on an environment. Despite recent advancements in data visualization, 2D or 3D imagery merely represents events from the actual data, presenting challenges in comprehension and control. Consequently, biologists often resort to simpler representations, ranging from 2D sketches to 3D animations, to complement their understanding of a phenomenon or to communicate about it. While extremely useful, abstract, conceptual sketches may only be understood by experts. At the other end of the spectrum, designing full 3D, animated illustrations typically requires collaboration with a professional artist, a time-consuming process that often involves a series of back-and-forth discussions. As a result, many cell biologists are in demand for user-friendly creation tools that allow them to design simplified yet expressive digital illustrations of their phenomenon of interest, appropriate to communicate both among peers and in educational contexts.

While current illustration tools are useful for scientific dissemination and education, they often lack the ability to facilitate sketching in volumetric environments. Such environments involve interactions between 3D shapes and dynamic state changes, posing a challenge for representation. Existing methods dealing with illustration at the cellular level have mainly focused on modeling vascular networks in 3D using sketching conventions, or have explored blood circulation to represent vessel pathologies. However, none of these approaches offers a complete system for creating and animating a complex environment in which cells can navigate and evolve dynamically.

In this paper, we focus on the cellular scale, particularly examining the infection phenomenon. This particular choice enables us to tailor our objectives accordingly. Specifically, we aim to design a schematic representation of what occurs under the skin during the infection phenomenon. Our concept involves creating a stable environment composed of collagen fibers and vessels within which cells navigate. Subsequently, a cut triggers the immune response to infection, with white cells deforming and then catching and removing the bacteria while red cells escape through the cut. To facilitate understanding and exploration of this phenomenon, we introduce an interactive sketching system for 3D modeling, animation, and narration. This system is specifically designed to meet the needs of biologists for effectively communicating and exploring this par-

*Corresponding author: Tel.: +33-625850213
e-mail: paulinehnoivier@gmail.com (P. Olivier)

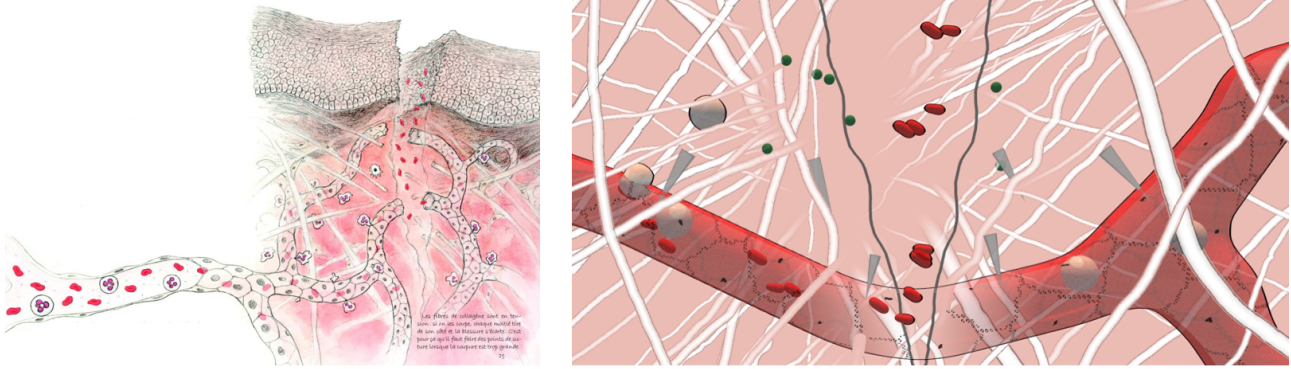


Fig. 1: An immunology illustration from the magazine *Globule* [1], © Renaud Chabrier (left) served as inspiration for a dynamic visual summary (right), i.e., an animation triggering narrative events, sketched using our system.

1 ticular phenomenon

2 This project was initiated through a collaboration with a bi-
 3 ologist and a professional artist to help us conduct both a pre-
 4 study on depiction and narration in biology. The latter con-
 5 firmed the need for an authoring tool to help biologists represent
 6 their understanding of the dynamic phenomenon of infection in
 7 a simplified way. Our solutions build on their feedback. While
 8 we use existing solutions for generating implicit surfaces from
 9 a sketch [2, 3], our main contributions are:

- 10 • A novel sketch-based modeling method to design distribu-
 11 tions of 3D shape, enabling the modeling of dense 3D en-
 12 vironments such as biological tissues with fiber networks
 13 in which cells evolve;
- 14 • A solution for sketch-based animation, applicable to both
 15 2.5D and 3D settings that builds on standard schematic
 16 conventions in biology to allow the specification of mo-
 17 tion and deformation;
- 18 • An interface for specifying and controlling temporal nar-
 19 rative scenarios with a series of events triggered over time;
- 20 • A final user study with 11 expert biologists, enabling us to
 21 validate the relevance of our tool and identify in which use
 22 cases 2.5D vs. 3D illustrations are the most relevant.

23 Throughout this paper, the illustrations focus on the challeng-
 24 ing case of immunology. We took inspiration from the drawing
 25 in Fig. 1 left, expressing the human response to infection at
 26 the cell level, and created before this work by one of our ex-
 27 pert co-authors. Using our tool to create a visual summary of
 28 the dynamic phenomenon involved, we achieved similar visual
 29 quality (see Fig. 1, right), while enabling the design of anima-
 30 tions triggered by narrative events positioned along a timeline.
 31 This provided a proof of concept of the applicability of our sys-
 32 tem in a biological context.

33 Lastly, note that the first version of this work appeared in a
 34 conference paper focusing on 2.5D animation [4]. The present,
 35 fully rewritten paper extends this work by introducing a 3D so-
 36 lution for synthesizing and animating sub-shapes within an im-
 37 plicitly defined container, enabling us to respectively generate
 38 and animate blood cells within 3D branching vessels. It also
 39 includes the final user study, lacking in the first paper, which

enabled us to discuss 2.5D versus 3D solutions with experts
 in biology. Let us note that although these experts were not
 directly involved in the development process of the tool, their
 feedback helped us to improve some aspects of the interface and
 our resulting toolbox.

2. Related work

Designing a tool for sketching visual summaries of biolog-
 ical phenomena, such as the one in Fig 1, requires the ability
 to model 3D shapes, and shape distributions, and to design ani-
 mations. It also requires studying previous illustration and nar-
 ration tools used in biology. We review these different topics
 below, referring the reader to [5, 6, 7, 8] for comprehensive tu-
 torials and surveys on illustrative rendering techniques and their
 applications in biology and anatomy.

3D modeling from a 2D sketch. Sketch-based modeling tech-
 niques range from methods that take full sketches and annota-
 tions as input to those that allow users to interactively design
 a 3D model through a series of interleaved sketching and nav-
 igation steps. Among them, a promising approach to sketching
 biological shapes is using implicit surfaces, well known for
 their ability to represent smooth, organic-looking 3D shapes.
 Bernhardt et al. proposed in Matisse [2] a general solution
 for the progressive sketching of such implicit, organic-looking
 shapes: they compute the medial axes of 2D regions succes-
 sively painted by the user under different viewpoints to use them
 as skeletons and generate 3D implicit shape parts progressively
 blended to form a single shape. Zanni et al. later improved this
 method with Scale Invariant Integral Surfaces (SCALIS)) [3]
 to capture fine shape details. Less general solutions were ex-
 plored to represent specific biological structures such as muscu-
 loskeletal structures [9] or vascular systems [10, 11, 12]. While
 it relies on sketching conventions used for teaching anatomy,
 the method of Pihuit et al. [10] is limited to a single sketch in-
 put representing a network of vessels. Closer to our goals is
 the work by Saalfeld et al. [11, 12] presenting interactive sys-
 tems aiming at creating and editing vascular systems to explain
 pathologies. While their first method [11] was restricted to 2D
 representation and relied on fluid simulation for the blood flow,

their second solution [12] was semi-immersive but required significant user intervention to adjust the weights of the Metaballs composing the vessel and to define blood flows.

In this work, we reuse the general solution from [2, 3] for building volumetric shapes from a sketch.

Synthesizing shape distributions from a sketch. Although extensive research focused on example-based texture synthesis [13], only a few methods tackled the synthesis of discrete vector textures. We focus on the latter, as the most likely to be extendable to the sketch-based creation of 3D shape distributions such as fiber networks in a volume representing some human tissue. Note that all the work discussed here only tackled 2D distribution synthesis. Moreover, while they showed promising results for texture synthesis, deep learning methods [14, 15, 16] are primarily image-based and have not yet been generalized to handle vector shape distributions effectively.

Landes et al. [17] was the first one to extend example-based vector texture synthesis, so far restricted to point distributions [18, 19, 20, 21], to the synthesis of shape arrangements. They simplify the input shapes into proxy geometries before computing spatial relationships between them, such as distances and relative orientations. Then, they use this analysis to synthesize a visually similar output texture. However, their method is limited to bounded shapes, in addition to being not real-time, which would make the creation of fiber networks difficult. Subsequently, Roveri et al. [22] presented an example-based distribution synthesis method handling both bounded and unbounded shapes. Their model decomposes shapes into point samples encoded in a functional representation and relies on neighborhood matching and energy optimization for the synthesis. Like other neighborhood-based texture synthesis methods, their approach requires input patterns with a sufficient amount of repetitions. Lastly, Olivier et al. [23] proposed a method for extending a 2D sketch representing a pattern-based vector texture while keeping the part drawn by the user unchanged. Their method constructs a hierarchy of support structures used to capture and reproduce multi-scale alignments in real time.

Our solution builds on the work of Olivier et al. [23], enabling us to achieve the 3D design of arbitrary shape distributions (including anisotropic distributions and unbounded shapes) in real time. We extend the former method to a new problem: the generation of 3D distributions from a single, 2D input sketch.

Sketch-based animation. Using sketches to design animations has already been investigated in the context of scientific illustrations, mostly to depict fluids. Zhu et al. [24] relied on sketches, manipulation, and fluid simulation to design an interactive method for animating fluid systems, enabling visualization and conveying information about cardiac abnormalities or surgical interventions. Kazi et al. proposed Draco [25] an interactive system to create animated illustrations that combine data samples and motion properties using kinetic textures. This was later extended by Kitty [26], allowing the specification of functional relationships between entities. Milliez et al. introduced Hierarchical Motion Brushes [27] to brush animated contents

onto a 3D scene, supporting coarse-to-fine animation thanks to content reuse at different levels of detail. However, it required some precomputation and lacked methods for processing distributions, which the authors later addressed using hierarchical spatio-temporal clustering [28]. Closer to our concerns, the Energy Brushes system [29] focuses on passive, secondary motion. Users generate local energy patterns by sketching strokes and defining flow particles. The energy brushes represent the direction of energy and forces that are continuously emitting particles along the sketched trajectory.

As in Draco, Kitty, and Energy Brushes, our system enables users to draw arrows to define velocity fields while we employ gradient triangles to specify force fields. We share the idea of continuously emitting elements similar to their flow particles. As in Energy Brushes, these elements can be sketched by the user, but in contrast, we augment arrows by morphing and implicit deformers to capture more general types of deformations.

Storytelling in biology. Biological visualization was extensively studied for many years, including for illustrative abstraction [30] and multi-scale visualization [31]. Subsequent work explored context-based visual preferences of static data [32, 33] and the importance of animated data relative to static one, as in [34] for complex bio-molecular configurations. In parallel, the dynamic nature of biological phenomena triggered a strong interest in interfaces focused on narrative design, whether for general data [35, 36, 37, 38], medical visualization [39, 40, 41, 42], or the narrative illustration of molecular phenomena [43, 44]. More recently, a few software specifically targeted this problem. CellPaint [45, 46] is a user-friendly interactive digital tool to create illustrations of molecular structures such as cells and viruses. Moleculumentary [47] enables the asynchronous communication of design ideas by controlling mesoscale animations of 3D models.

The work we present here is complementary: rather than reusing existing 3D animated content in the context of narration and addressing their differences of scale, we target the interactive, sketch-based creation of animated 3D content to support new narration ideas. In contrast to previous work, all narrative steps do not need to be known beforehand. Instead, we aim to create an interactive tool enabling biologists to illustrate their vision of their phenomena of interest and explore new narrative options.

3. Biological Context and Design Choices

3.1. Preliminary study with biology researchers

To better understand how biologists represent and communicate the phenomenon they study, we conducted a preliminary study with four cell biology researchers specialized in different subtopics. The participants included: a Research Director with 35 years of experience in cancer research and experimental therapeutics, a Professor with 20 years of expertise in cell mechanics, another Research Director with 26 years of experience in cell polarity and division, a Research Director holding 38 years of expertise in membrane and cytoskeleton dynamics, particularly within the context of breast cancer. They explained

to us in simple terms the main steps of their studies, employing visual analogies or visual vocabulary when needed, (e.g., “cutting similarly to scissors”, “cells hooking onto collagen fibers to squeeze through”). They also highlighted the characteristics of their phenomenon of interest in terms of shape types, behaviors, and constraints. We observed that they all presented their studied phenomenon from a narrative angle. In this narrative framework, an initial animated environment undergoes evolutionary changes in response to specific triggering events. Despite clear oral explanations and some sketches or 2D figures as visual aids (see Fig. 2), there was a notable lack of visual support to effectively represent the dynamic events they described. This study highlighted the inadequacy of using only static 2D figures for biologists, as they fail to capture the dynamic, 3D nature of phenomena aligned with their mental vision. This observation served as the driving force for the foundation design of the framework we propose.

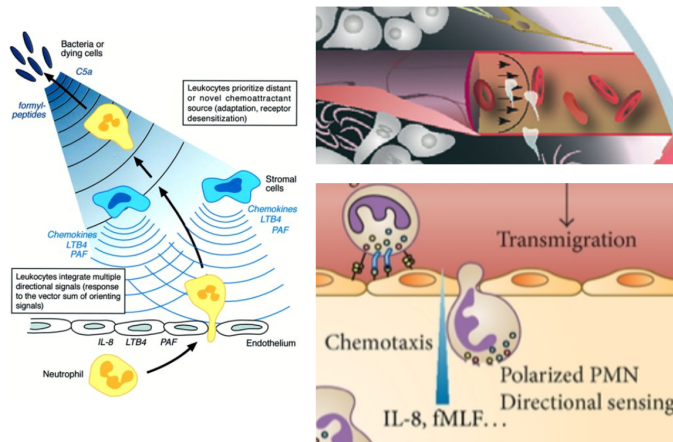


Fig. 2: Examples of 2D illustrations in cell biology that emphasize the use of schematic visual elements: on the left, ([48]), arrows illustrating the trajectory of a cell; in the top right corner, ([49]), streamlines depicting the flow within a vessel; in the bottom right corner, ([50]), a blue gradient triangle indicating the attraction of cells outside the vessel.

3.2. Schematic visual vocabulary in cell-biology

Our biologist collaborator enriched this study by formalizing the schematic vocabulary frequently used in biological illustrations to describe dynamic phenomena (see Fig. 2):

Arrows (left) depict the kinematic trajectory of an element, giving both direction and speed of displacement;

Streamlines (top right) indicate flow within a vessel using three arrows followed by a straight line for homogeneous flows or by a curved line for non-uniform flows (for example, allowing for a lower speed near the vessel membrane due to friction);

Gradient triangles (bottom right) are used to represent the difference of gradient in a medium, (e.g., pressure gradient or chemical gradient) denoting the attraction of elements.

3.3. Findings and design goals for DynBioSketch

Building upon this comprehensive preliminary study and the accompanying illustrations, we have categorized the types of 3D modeling tools, animation mechanisms, and rendering techniques to address in the development of our interactive authoring tool:

- **Variety of shapes observed:** volumetric elements such as organic shapes (e.g., vessel, cells, and their interiors); larger surfaces or volumes, particularly those unbounded at the observation scale, (e.g., skin); and linear shapes like fibers. We noticed that organ surface representation often requires the inclusion of textures, such as Voronoi-type tessellations for cell boundaries.
- **Shape distributions:** applicable to unbounded shapes (like a field of fibers) and nested bounded ones (like cells within a vessel), which also underlines the need for models capable of extrapolating a user-sketched input into a 3D distribution.
- **Animation:** need to sketch global flows, individual trajectories, and control shape deformations. In biological representation, precise control of animation synchronization is essential for building narrative scenarios. Sequences of triggering events can define such scenarios and effectively convey the dynamics of the phenomenon.
- **Expressive rendering:** based on transparency and local shape cuts to effectively represent nested environments and movements within the current element.

3.4. Methodology and technical contributions

Drawing inspirations from visual representation and narrative techniques in biology, we conceptualized DynBioSketch, as a new creative medium that seamlessly integrates the progressive modeling of a 3D biological environment with the development of animated narratives. In this interactive sketch-based modeling framework, biologists can convey their understanding of a phenomenon. Additionally, this innovative approach serves as an accessible entry point for non-expert users to explore biological phenomena. In both cases, users initiate the process by designing an initial, stable environment and subsequently triggering its temporal transformation. Let us note that by a stable 3D environment in an animated interface, we refer to a scene configuration that remains reliable and consistent while being animated or not. This setup allows users to iterate and refine environment designs or animation ideas quickly using simple sketches.

As illustrated in Fig. 3, we centered this work around the infection phenomenon. In implementing our prototype, we focused on the most important of our design objectives, aiming to achieve the right balance between simplicity and richness of representation. We proposed solutions that enable users not only to model individual 3D shapes from a sketch but also to express bounded and unbounded distributions of 3D shapes (Sec. 4). In line with this approach, our sketch-based animation mechanisms go beyond rigid motion, facilitating the specification of deformations. This functionality is complemented

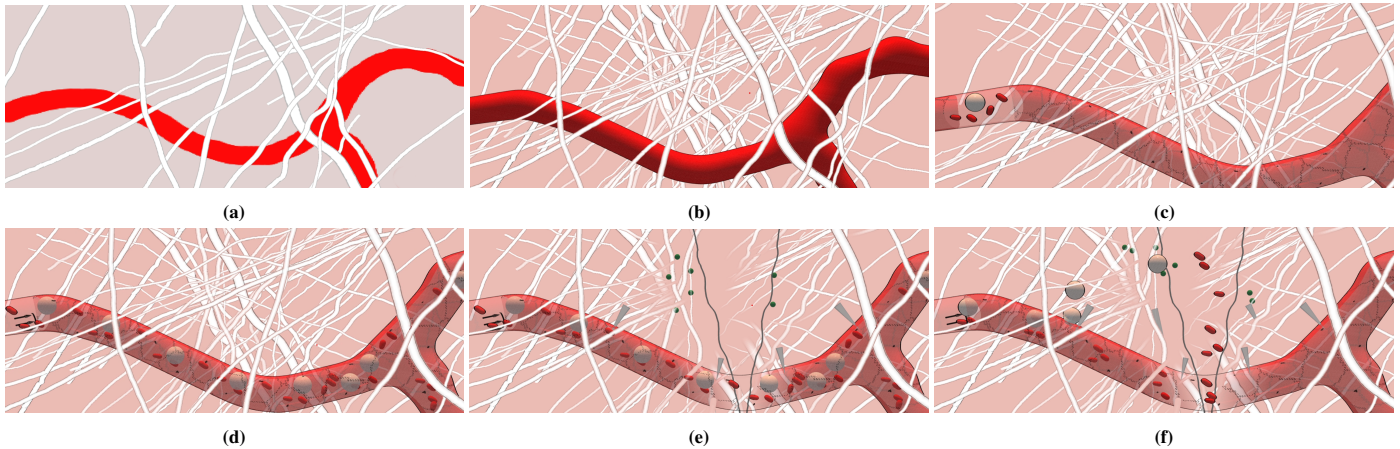


Fig. 3: Overview of *DynBioSketch*: from creating the 3D biological environment (a-c) to designing a dynamic scenario (d-f).

by a timeline that allows users to navigate and trigger narrative events (see Sec. 5). Throughout the creative process, users benefit from expressive visualization thanks to the combination of 3D shading, contouring, transparency, and textures (see Sec. 6). We have designed all the methods mentioned above as real-time solutions, which is mandatory for their seamless integration into an interactive authoring tool.

Case study. To highlight the usability of our system to create an animated biological story, we selected the bacterial infection phenomenon as our case study: this phenomenon is initiated by a cut in the tissue, impacting the skin, the underlying fibers, and the vascular network, which causes blood cells near the cut to flow through it and allows external elements like bacteria to enter the environment. Consequently, white cells, as key players in the immune response, are activated to migrate from the vessel to the cut area, contributing to the wound healing process. This migration involves their initial attraction to the vessel's inner membrane, followed by a rolling motion along this endothelium. Ultimately, such cells escape by undergoing shape deformation to pass through small openings between the endothelial cells of the membrane. We selected this particular example to exemplify our technical contributions in this article, as it features an illustration by a professional artist. Biologists have also validated this illustration, establishing it as an authoritative visual reference that showcases the capabilities of our system. Fig. 3 provides a comprehensive overview depicting: (a) the input 2D sketch, (b) the resulting 3D environment, (c) an illustrative example showcasing cell distribution within a vessel, along with the rendering of its surface, (d) the resultant distribution of cells and the use of the flow tool to specify their movement, (e) the cutting of tissues, triggering an infection resulting in the arrival of bacterial cells (in green) and white cell activation using gradient triangles and (f) the deformation of white cells as they cross the vascular membrane, attracted by the bacteria, while red cells escape through the cut.

4. Creation of a 3D biological environment

This section introduces three new tools that enable users to progressively create a stable 3D environment, in which the tar-

geted biological phenomenon will later be triggered. Although stable over time, the 3D environment can include animated elements, such as blood cells following a stationary flow predefined within a vessel.

4.1. Volumetric background tissues from a 2D sketch

Our inspiration figure (see Fig.1, left) emphasizes that biological environments usually feature distributions of 3D elements in the background (e.g., the field of collagen fibers around the vessel in our case study).

Our goal is to generate a 3D distribution of elements from a 2D input sketch, allowing users to navigate within this generated 3D environment. Our concept involves allowing users to sketch a small representative sample of what they want to visualize from the front. Subsequently, we extend this sample in 2D by a given ratio k before immersing the expanded 2D distribution in 3D by introducing an extended depth of h . The resulting 3D volume of material can be replicated with adjusted rendering to simulate an infinite environment.

The 2D expansion of the user-sketched vector texture follows the methodology outlined by Olivier et al. [23]. This approach efficiently computes a larger 2D distribution of strokes through a multi-resolution alignment analysis, seamlessly extending the initially sketched sample. Specifically, vector sketched strokes are efficiently organized into a hierarchy of linear or curved support structures, each characterized by a main direction and a ribbon containing clustered sub-structures. During synthesis, the existing supports in the exemplar are progressively extended and duplicated from top to bottom of the hierarchy, incorporating some randomization to align with the larger output space. Clones of the input strokes are ultimately distributed, with suitable jittering, along the lowest-level structures (refer to [23]).

In the second stage, we determine depth parameters for each stroke to expand the generated 2D vector texture into a 3D distribution. Our approach relies on a common perceptual assumption regarding general views: the alignment of strokes in the 2D exemplar intentionally represents a 3D alignment rather than arising solely from the projection of a specific viewpoint. Consequently, we immerse clustered strokes with similar depths through the following two steps.

For each cluster, we initially define a midpoint's depth d_e and a slope angle δ relative to the sketching plane (XY). These parameters are selected randomly within the ranges of $[0, h]$ for d_e (where h is the depth of the output domain) and $[-\pi/4, \pi/4]$ for the slope (a range informed by perceptual studies [51]). For more details, we let the user refer to the model of Olivier et al. [23].

To simulate an infinite field of 3D strokes, duplicating the resulting volume along the depth axis is crucial. We configure these duplicated instances to overlap randomly in depth, enabling intricate arrangements. To achieve this, we replicate the initial volume and shift each instance in the depth direction by a factor of $l \cdot h$, where l is an integer in the range of $[1, \infty]$. This process generates a virtually infinite field of anisotropic elements. The visualization of these elements is guided by two perceptual assumptions. Firstly, distant objects gradually fade away, implying that strokes in deeper layers appear less prominent. Secondly, we conceptualize the 3D space with a toroidal topology in depth, meaning that objects exceeding a certain depth "wrap-around" and reappear at the front. By incorporating these perceptual assumptions, we can effectively represent multiple layers and create a visually coherent volumetric display (see Fig. 4).

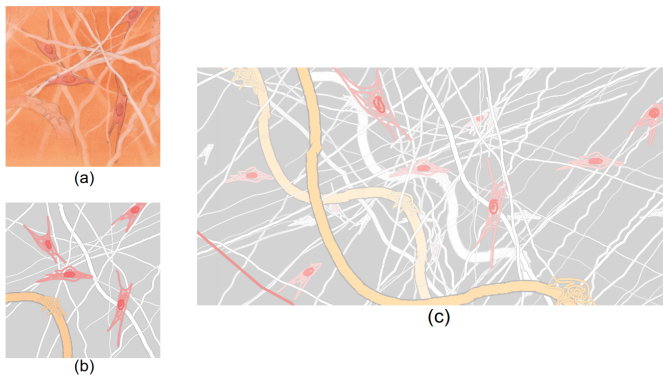


Fig. 4: (a) Depiction of cells navigating through a 3D arrangement of fibers in a biological context; (b) 2D sketch input inspired by (a); (c) Output generated by our method for synthesizing the 3D fiber and shape distribution.

4.2. Sketch-based modeling of nested shapes distributions

Since we target the creation of organic shapes such as vessels, cells, and internal cellular structures, we rely on an existing sketch-based implicit modeling methodology, borrowed from Matisse [2] to ease their creation: users alternate navigation and painting 2D regions in a plane facing the camera. The medial axis of each painted region is extracted and simplified into a graph of segments, augmented with radii information. This graph serves as a skeleton to generate a scalar field modeling a density of matter around it and progressively decreasing with distance. A specific isosurface defines the target implicit surface. Note that while each sketch results in a flat skeleton, and thus an implicit surface with a flat silhouette, sketching such shape parts from different viewpoints and then blending their fields through summation allows the creation of arbitrary 3D shapes. In our implementation, we use the field defined in

SCALIS [3] to better match the sketched contours than in the original Matisse system.

In addition to sketching isolated organic shapes, the user is invited to sketch distributions of sub-shapes nested within a larger one (see Fig. 5). This can be useful, for instance, to populate a vessel with blood cells. To this end, we developed a second extension of the method of Olivier et al. [23] dedicated to distributions of bounded shapes within a container.

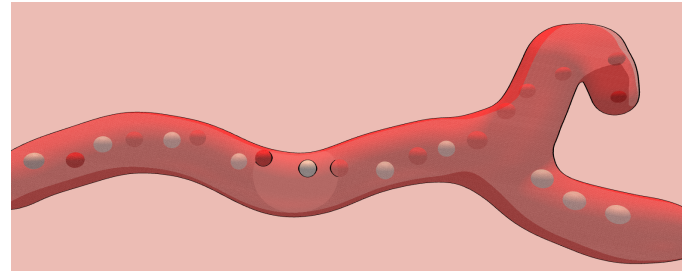


Fig. 5: From an exemplar to a distribution of cells in a 3D vessel.

The user first selects as *container* any volumetric implicit shape previously generated from a sketch. Using a camera facing the plane where a flat part of the skeleton is defined, the user selects a specific section of the container (made transparent) and creates a sample of the desired distribution. This can be done by directly sketching sub-shapes within the container or by instantiating some existing shapes. Note that this input distribution exemplar should only include bounded shapes entirely within the container. We then propose a 2.5D solution for analyzing the exemplar and extending it within the entire container: we analyze the exemplar in the skeletal plane by taking into account the 2D projections of the sub shapes, and after distribution synthesis in 2D, the depth of the newly generated sub-shapes is computed from the original depths in the exemplar, as we detail in the following paragraphs.

Adapting the original 2D distribution analysis and synthesis method from Olivier et al. [23] to free-form planar regions is challenging.

As the synthesis is limited to the interior of the free-form container, we cannot directly extract shape alignments and their directions by a successive grouping of straight lines. However, the scalar field defining the implicit container is an excellent basis for grouping. In our case, we call "quasi-aligned" the shapes positioned around the same isovalue and we define their support structure by the corresponding isoline curve in the scalar field rather than by a straight line. As in the original method, support structures are grouped recursively, with ribbons expressing the variance of the elements they represent. For that, we use the difference of isovalues as clustering distance. These simple adjustments to the original method allow us to apply the existing algorithms to analyze the free-form input sample and then extend the distribution to the full 2D projection of the container shape.

It is important to note that isolines do not require explicit computation during the synthesis phase. Instead, we iteratively instantiate the next shape to generate along a given isoline using a series of small displacements from the previous position. Each step involves a translation along the tangent direction fol-

lowed by a projection along the field gradient back to the target isovalue. We define the tangent direction as the cross-product between the field gradient and the depth direction, and the step size by the local curvature. In addition, a slight perturbation is introduced in the new position to avoid significant repetition during the synthesis phase. After extending the planar distribution to the entire container, the generated clones for each sub-shapes return to their original depth, with some extra jittering, which allows us to disperse them throughout the 3D container.

We expanded our synthesis model into 3D to accommodate more intricate containers. The primary challenge lies in managing the transition in the depth direction when moving between different 3D branches. To tackle this issue, we leverage our definition of trajectories (detailed in Section 4.3) along the isoline direction, which enables us to calculate the subsequent positions in a container, where the generated clones follow a spiraling path around an isosurface. To introduce additional variation to this trajectory, we analyze the local cross-section of the isosurface surrounding the clone's position and rotate this position around the cross-section, introducing variations to the trajectory. The degree of rotation is determined randomly within predefined angle ranges. The position of the clone shape serves as the basis for defining its local frame and the direction of its isoline, which in turn determines the subsequent position for the next (clone) shape to create.

Figure 5 illustrates the method's robustness in generating distributions of sub-shapes within containers with 3D branches, facilitated by the implicitly defined isosurfaces.

4.3. Creating stationary animations

The last step in creating a representative biological environment is to enable animation. We focus here on background motion that remains stationary over time, as opposed to the unique animated events discussed in Section 5.3. The sketch-based tools presented next enable users to define velocity fields that guide the motion of sub-shapes within a container, such as blood particles animated within a vessel. To this end, we use the flow to directly set the velocity of kinematic sub-shapes, although using it to exert viscous frictional forces on physically-based objects would also be straightforward.

In the following, we take the hypothesis of elongated containers such as blood vessels. Implicit shapes being closed surfaces, we truncate the extremities of such a container in the flow direction and use a toroidal topology to allow the sub-shapes that reach the upstream extremity to be re-generated downstream. During the animation, we detect collisions with the container and re-project sub-shapes in the field's local gradient direction to prevent unwanted leakage of sub-shapes across the container surface. Note that collision detection and response between sub-shapes can be neglected, given that they follow the same velocity field and are usually located at different depths.

To allow comparisons between 2.5D and 3D illustrations in the user study, we developed two alternative versions of the flow animation system, presented next: the first one is specific for flat-silhouette containers, which were sketched from a single viewpoint and therefore have a planar skeleton, enabling us to use a planar flow; whereas the second focuses on more general

flows on 3D arbitrary containers that may bend out of the plane facing the camera.

3-Arrow flow tool for 2.5D illustrations. In the following, we call **2.5D container** the input implicit container drawn in a plane facing the camera, and f the field used to define it. To create a 2.5D animated illustration, we compute a 2D flow used to animate the planar coordinates of the sub-shapes within the container while their depth is constant.

We inspire from conventions in biological illustrations to propose a fast way to design a global flow spanning the whole container: the user sketches three arrows to indicate the desired direction and strength of the flow and adds either a straight or a curved line near their extremities indicating a possible reduction of velocity, mimicking frictional effects, near the container's surface (see Fig. 6).

To extend this locally-defined flow to a velocity field spanning the entire container, we take the following hypothesis: we orient the flow along the isolines of the container's field f , with a constant velocity along each isoline. Moreover, although the sketched flow may be small relative to the size of the container, we assume that the flow extends over its entire cross section, with a maximal value at the middle and a minimal one close to the surface, which enables us to define the flow as follows. We extract the flow maximum velocity and lateral damping from the user's sketch. We then compute the norm of the velocity field from changes of isovalues of f within the container, as illustrated in Figure 6. We also register the orientation of the flow relative to the container's skeleton by projecting the central arrow to the closest central skeletal segment. The local velocity at any position P within the container is then defined as a vector perpendicular to the local gradient direction, oriented as to agree with the flow direction propagated along the skeleton, using the norm registered for the isovalue $f(P)$ of the flow.

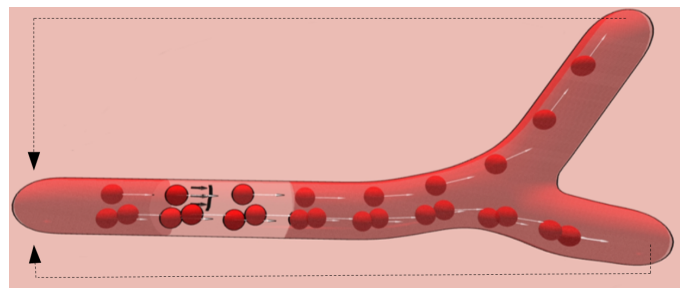


Fig. 6: Use of the triple arrows tool (in black, within the vessel) to create a velocity field within a 2.5D container. The latter, a closed implicit surface, is given a toroidal topology to bring back downstream (long arrows) all the objects advected beyond a predefined upstream limit.

Frame-based flow tool for 2.5D and 3D illustrations. To enable the animation of sub-shapes within arbitrary 3D containers in addition to 2.5D containers, we provide a second method for defining stationary flows. We address the lack of parameterization of implicit surfaces by leveraging the local frames defined by the field gradient direction, tangent, and binormal of the container's isosurfaces. Note that contrary to the 3-arrow flow tool, the frame-based tool is not limited to the design of a flow that

follows the container's isosurfaces since we can define locally contractive flows that guide the sub-shapes closer to the container's skeleton or dispersive flows that push them towards its outer surface, as well as a combination of both.

The input is a 3D container, implicitly defined as an isosurface of a field f , within which the user has defined sub-shapes. The user then uses an arrow tool to sketch one or several samples of flow velocity vectors within the container shape in a plane facing the current camera:

- If the user sketches only one single arrow, the latter is used to define a steady flow field within the whole container;
- Otherwise, the arrows are interpreted as flow velocity samples at different isovalues, enabling linear interpolation in between (see Figure 7).

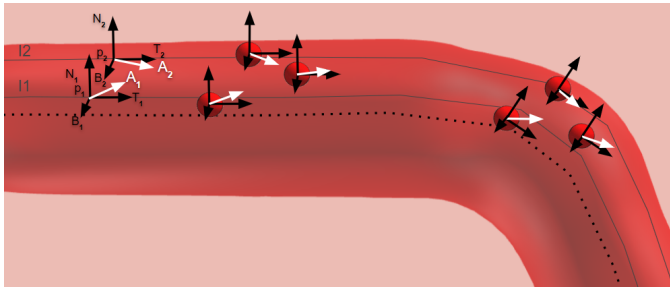


Fig. 7: The user sketches two arrows, respectively A_1 and A_2 (in white), to specify a flow with dispersive behavior at the center and compressive behavior near the surface of a container. Both arrows indicate a motion to the right at different speeds. We store each arrow A_i in the local frame defined by the tangent (T_i), the normal (N_i), and the binormal (B_i). These arrows are combined to define velocity vectors (white) anywhere in the container (see the red sub-shapes), using interpolation according to the local isovalue.

An arrow $\mathbf{A} = (A_x, A_y, A_z)$ sketched at a point P , is decomposed into local coordinates (V_T, V_N, V_B) inside a local frame (T, N, B) aligned with the isosurface passing through P , and computed as follows: N is the normal to the local isosurface computed from the gradient of f , B is binormal to the isosurface defined as the unit vector pointing to the camera, $T = N \times B$ is a unit tangent vector to the isosurface. Therefore, V_T defines the speed of motion along the isosurface of isovalue $f(P)$, while V_N defines the speed of motion in the field gradient direction, enabling the definition of more general flows such as dispersive or contractive ones.

In the case of a unique arrow, we reuse the coordinates (V_T, V_N, V_B) to set the local velocity vector at any point P within the container's shape while being interpreted in a new local frame (see the black arrows near sub-shapes in Figure 7). Conversely, if the user sketched several velocity vectors \vec{A}_i at different points P_i , each vector is decomposed as before and attached to the isovalue $f(P_i)$. This allows for the linear interpolation of new velocity vectors in local coordinates. Let P a point inside the container, such that $f(P) \in [f(P_i), f(P_j)]$, we define $\lambda(P) \in [0, 1]$ and then $\vec{A}(P)$ as:

$$\lambda(P) = \frac{f(P) - f(P_i)}{f(P_j) - f(P_i)}$$

$$\vec{A}(P) = (1 - \lambda(P))\vec{A}_i + \lambda(P)\vec{A}_j$$

For points outside the range of field values of the sketched velocity samples, we use the nearest sample to compute $A(P)$.

During the animation, the sub-shapes' motion is guided by the flow computed using these interpolated velocity vectors in local-frame coordinates and then re-expressing them in the world frame given the new local frame (T, N, B) —which may have rotated—at the current sub shape's position. The sub-shape moves along the resulting velocity direction. Note that in the case of flow specified to remain parallel to the isosurfaces ($V_N = 0$ everywhere), precision can increase by reprojecting each sub-shape, at each timestep, to their initial isosurface along the local normal direction.

Cyclic-frames versus Flow-frames: Computing consistent local frames during animation is essential for the above method to work efficiently.

Upon creation, we initialize the sub-shape's local frame at its centroid by computing (T, N, B) as above, using the camera's view direction for setting B . At each time step i , the sub-shape is moved to its new position P_i , computed from the local frame and interpolated flow velocity at P_{i-1} . We then use the new normal direction and the previous binormal direction to update the local frame:

$$\vec{N}_i(\mathbf{x}) = -\vec{\nabla}f(\mathbf{x})$$

$$\vec{T}_i(\mathbf{x}) = \vec{N}_i(\mathbf{x}) \times \vec{B}_{i-1}(\mathbf{x})$$

$$\vec{B}_i(\mathbf{x}) = \vec{T}_i(\mathbf{x}) \times \vec{N}_i(\mathbf{x})$$

This algorithm directly generates *cyclic-frames*, i.e., frames used to compute periodic motions of sub-shapes turning within a container, e.g., along isosurfaces. Indeed, the normal vector always points outwards, so even a global flow defined using a single arrow is interpreted as two different orientations at the top and bottom of the container. To define directional flows spanning the entire cross-section of the container (such as the flows of blood cells within a vessel), the local frame must be transformed into a *flow-frame*, as follows: any sub-shape initialized below the centroid of the container in the flat drawing region facing the camera has its local frame defined using $-N$ instead of N , which in turn orients T back to the consistent flow direction.

See Figure 8 and the second supplemental video for a visual comparison between animations created using cyclic vs. flow frames. Our method ensures *temporal coherence*. It is robust to containers of arbitrary topological genus (top), possibly with branchings (bottom), including 3D containers drawn from different viewpoints, which lack a fully connected skeleton. Note that due to the recomputation of local frames, sub-shapes typically follow a twisting motion within such containers, as if they were forced back in by a collision with the outer surface. In our experiments, this was considered a positive feature of the method.

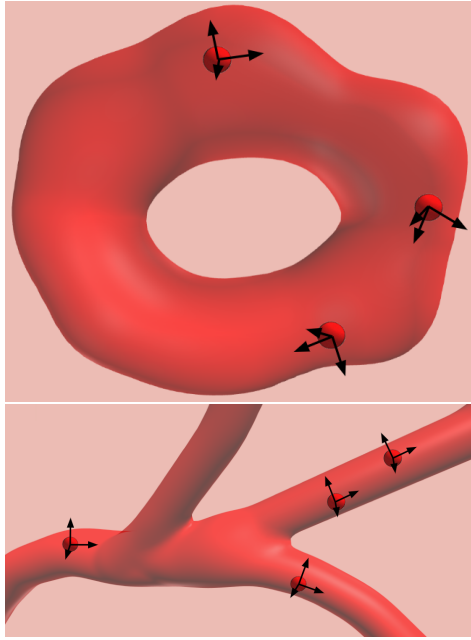


Fig. 8: We provide two categories of local frames: cyclic frames (top) defined by the local normal direction and enabling the generation of cyclic motions along isosurfaces, versus flow-frames (bottom) where the normal axis is flipped (i.e., point to the inside instead of the outside) for particles initialized below the center-line of the container. Then, all the particles passing through a given cross-section move in the same direction.

5. Design of the main biological phenomenon

At this stage, the users have created the initial, stable animated 3D environment as a basis for the phenomenon they want to represent. Users can now design their phenomena of interest by placing triggering events along a timeline and sketching the resulting consequences in the 3D scene.

5.1. Triggering events

In our scenario, the triggering event is the incision made in the environment, which affects the tissue, fibers field, and vessel network. We maintain real-time reactivity after the cut by using expressive rendering techniques (See Sec. 6 for the implementation details). Such incision disrupts blood flow as neighboring cells migrate through it. We describe this behavior by assigning cells an elliptical trajectory towards the center of the incision, followed by random oscillations as they leave the environment through the cut. Moreover, bacteria can emerge in the environment. Users can introduce new bacterial shapes using our Organic Shape Creation tool (see Sec.4). These shapes exhibit localized random movement, reinforcing their living appearance. Such a triggering event is marked on the timeline as an important event and is essential for narrative scenarios (see Sec. 5.3).

5.2. Expressing the consequences

Setting new events in the narrative dynamic scenario implies some consequences on the environment that need to be modeled and integrated into our system and along the timeline. We let users specify the following consequences:

Sketching motion. We propose a series of sketch-based animation tools to let users express the consequences of this event on the environment. To seamlessly integrate animation design into our sketch-based creative system, we draw on visual vocabulary derived from the analysis of biological illustrations (see Section 3). This approach enables us to establish a direct link between the familiar schematic representations in biology and the motion or deformation predicted in our system. While descriptions are initially provided in the plane facing the camera and applied to shapes at a constant depth, these movements and deformations can extend to a full 3D animation. We introduce motion design tools from different viewpoints for users to combine their effects. We propose two sketch-based animation tools to express such consequences: arrow and gradient triangles.

The user initiates a trajectory by sketching a single arrow from the designated target object (see Fig. 9). We meticulously sample this trajectory based on the drawing time, ensuring that the speed of the gesture directly influences the speed of the subsequent movement. The displacement between two consecutive points along the timeline is then incorporated to streamline the synchronization of various motions. This approach enhances the fluidity and coherence of the animation as a whole.

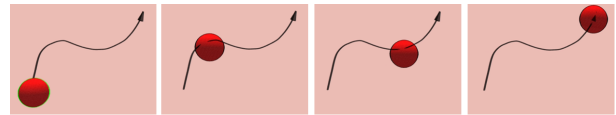


Fig. 9: The cell is following the user's arrow.

On the other hand, the user can position gradient triangles in the 2D plane facing the camera to specify directional force fields due to chemical gradients (see Fig. 1, right). These forces are applied to neighboring shapes of a given category with their intensity decreasing with distance. In our case study, we use such gradients to model the chemical procedure that attracts white blood cells to the site of infection. Additionally, we introduced a novel representation of the attractive forces exerted by cells escaping the vessel toward bacteria using an implicit type of gradient triangle. We chose an ellipse-based path directed toward the nearest available bacteria to model these forces.

Specifying shape deformation. Enabling shape deformation and potentially altering their topology through fusion or fission poses a significant challenge in biological animation. For example, in our specific case study, the deformation of white blood cells is essential to let them navigate tight spaces between vessel membrane cells and thread their way through complex fibers. To meet this challenge, we have designed two mechanisms that allow users to precisely specify and control the desired deformations: shape morphing and shape deformation through a hole.

For shape morphing, we employ sketch-based modeling to define the initial and final shapes, positioning them at specific moments along the timeline. To generate the intermediate shapes, we utilize a morphing algorithm based on Minkowski sums [52], as illustrated in Fig. 10.

The alternative model proposed here is specific to shape deformation through a hole. Considering a target shape T tasked

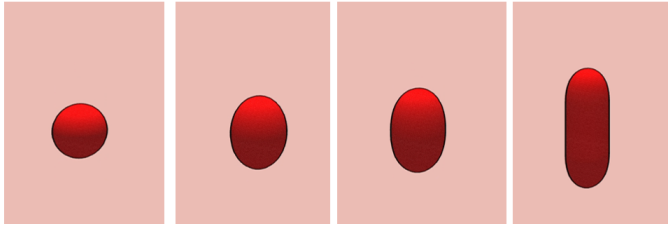


Fig. 10: Morphing between the user-specified initial (left) and final (right) states.

1 with traversing a hole H within a surface S . Let R_t and R_h be the
 2 radii of the target and the hole-bounding spheres, respectively.
 3 We seek to model the following behavior:

- 4 • If $R_t < R_h$, the target escapes without undergoing any de-
 5 formation.
- 6 • If $R_t > 2R_h$, excessive compression of the target will occur,
 7 rendering it unable to escape.
- 8 • If none of the above conditions apply, the target will de-
 9 form while retaining an almost constant volume.

10 In our system, the target cell T and the surface S (a vessel)
 11 are implicit surfaces. The hole is an implicit primitive generated
 12 by a negative field created from a skeleton point. To simulate
 13 the impact of passing through the hole on the shape of the tar-
 14 get, which must compress and elongate before returning to its
 15 original shape on the other side of the surface, we introduce a
 16 deformation field in addition to the one that defines T , as fol-
 17 lows.

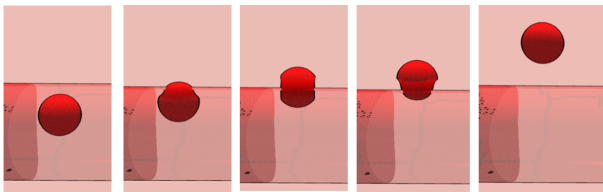


Fig. 11: Cell deforming to cross a membrane through a hole.

18 We position a deformation primitive on top of the hole, di-
 19 viding its region of influence into three distinct zones: an outer
 20 ring, a central ring, and the remaining spherical zone around
 21 the center. In the outer ring, we apply a negative field to lo-
 22 cally compress the target shape and ensure it fits correctly into
 23 the available space of the hole. In the central ring (correspond-
 24 ing to the size of the hole), a positive field is used to preserve
 25 the volume of the target during deformation. In this way, the
 26 target's shape stretches along its trajectory as it approaches the
 27 hole and the same applies when it exits from the other side. We
 28 set the deformation field to zero in the spherical inner region.

29 Fig. 11 illustrates the impact of applying this deformation
 30 primitive to a passing target. In our specific case study, this
 31 approach allows us to efficiently simulate the behavior of white
 32 blood cells crossing the vessel membrane.

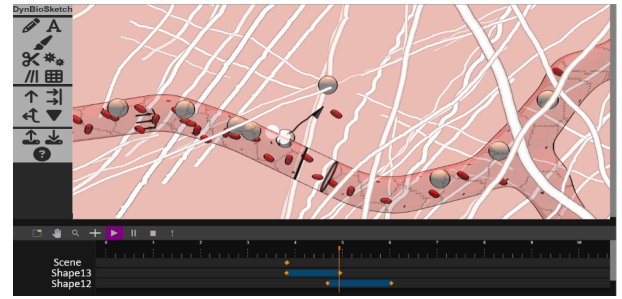


Fig. 12: Illustration of a narrative scenario in DynBioSketch: the user sketched a trajectory arrow that saved an event at time 4.8 and returned to this event to replay the displacement.

5.3. Narrative scenarios

34 As shown in Fig. 12, the interface of our creative system in-
 35 cludes a toolbox and a timeline. When the user launches an ani-
 36 mation, the timeline becomes active. We manage two categories
 37 of animation: 1. periodic animations that persist throughout the
 38 session, such as the flow in a container or the random move-
 39 ment of bacteria; 2. non-periodic animations triggered by spe-
 40 cific events (for example, the appearance of a chemical gradient
 41 due to a cut) are the only events displayed on the timeline, as
 42 are the animations that follow them.

43 Our system offers two modes: 1. creation mode, which lets
 44 users navigate and create or modify shapes and generate ani-
 45 mations and position-triggering events on the timeline; 2. playback
 46 mode that lets users view the current animated DynBioSketch
 47 from a specific point in time.

48 When instantiating a new event, we establish an event state.
 49 This state records both the time of the event and the current
 50 state of the elements in the environment. When the user returns
 51 to this event, we reuse these saved states. Fig. 12 illustrates an
 52 example at time 4.8, where the user uses the arrow tool to draw a
 53 white blood cell out of a vessel through a hole. Switching from
 54 one mode to the other enables users, especially biologists, to
 55 systematically illustrate the desired phenomenon and to design
 56 and record an associated narrative if needed.

6. User interface and Visualization choices

6.1. Interface of DynBioSketch

58 In addition to a timeline located below the main window, we
 59 provide the user with a main design interface at the left, com-
 60 posed of 14 widgets, and described below from top to bottom
 61 and left to right (see Figure 13):
 62

- 63 • Pencil: A pencil tool to draw annotations;
- 64 • Letter A: Annotation tool to create a text box for annota-
 65 tions;
- 66 • Brush: A 2D painting tool to create or edit organic shapes;
- 67 • Scissors: A cutting tool to virtually cut the membrane of
 68 a shape, enabling one to see inside (cutting gesture within
 69 a shape) or to cut through the shape, as we did in Fig. 1,
 70 right to model the cut through tissues (cutting gesture from
 71 the outside, orthogonal to the shape);

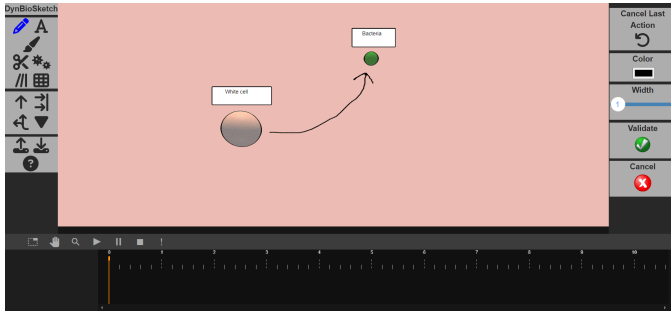


Fig. 13: The DynBioSketch's interface: the creation tools for 3D geometry and for sketching the volumetric environment are at the left of the main window (showing here with a white cell and an arrow to make it move to a bacteria and two annotations). The timeline for triggering events and playing the animation is at the bottom. The interface at the right contains standard tools such as setting the size and color of the brushes and undoing/redoing

- 1 • Duo of cells: Shape distribution tool to select a sample of
2 distribution (a few elements such as cells) and then trigger
3 the synthesis of the full distribution within the container
4 containing these elements;
- 5 • Vertical lines: 2D sketch distribution tool to open a win-
6 dower for the 2D sketching of the environment texture
7 (fibers in our main example) and then call for the synthesis
8 of a 3D environment in an extended domain;
- 9 • Grid: Tessellation tool to sketch 2D black dots, used to de-
10 fine Voronoi regions to texture the membrane of an organic
11 shape;
- 12 • Single arrow: Trajectory arrow tool to define a trajectory
13 arrow for an organic shape;
- 14 • Double arrow and vertical line: 3-arrow flow tool to sketch
15 the 3-arrows flow used to animate 2.5D illustrations;
- 16 • Fork arrow: Frame-based flow tool to sketch the frame-
17 based flow used for 2.5D and 3D illustrations;
- 18 • Triangle: Gradient triangle tool to create gradient trian-
19 gles;
- 20 • Upload: : Upload tool to retrieve an existing DynBioS-
21 ketch model;
- 22 • Download: Download tool to save a DynBioSketch model;
- 23 • Help: Help tool to display a window with a reminder of
24 the different tools;

In addition to the modeling and animation tools we described throughout this paper, the DynBioSketch interface allows users to annotate their sketches by selecting the pencil tool from the left menu and choosing the color and size of the stroke from the right menu. Additionally, users can add text boxes and insert descriptive text to explain what is happening in the scenario.

6.2. Illustration-inspired rendering

The user benefits from an expressive, non-photorealistic rendering throughout the creation process. This rendering aims at facilitating the overall perception of the 3D scene and animated events, presenting the entire scene as a shaded 3D sketch reminiscent of the artistic illustration in Fig. 1 (left).

We render each shape generated by our sketch-based modeling system with a default material incorporating Phong shading with procedural value noise, resulting in a paper-like effect. In addition, we apply a black outline effect using a pen-and-ink shader (see the vessel and fibers in Fig. 1). This rendering approach is versatile and can be adapted to the requirements of the animation scenario.

Organic tessellation of surfaces. To give surfaces, such as vessels, the appearance of an organic membrane, we offer an option for texturing a surface with Voronoi-like regions representing the membrane cells. We marked these regions with black dots representing their so-called *kernels*.

To generate such textures, the user clicks at the desired positions to place the kernels on the selected surface. With each click, we use ray scattering from the camera's point of view to identify $P1$, the first intersection with the implicit surface, chosen as the kernel point. Instead of using the second ray-surface intersection point $P2$ as the kernel on the other side of the surface, we introduce randomness in the ray direction before calculating the next intersection point, starting with $P1$. This randomization avoids unrealistic cell alignments. Automatic point pattern generation techniques on surfaces may be used as well.

Once we position all the kernels, we calculate an approximate Voronoi diagram on the free-form surface, directly encoded in the surface's geometric attributes to facilitate 3D navigation. As a first step, we classify the mesh's vertices according to their proximity to the nearest core. As a trade-off between efficiency and precision, we chose Euclidean distance as a rough approximation to geodesic distance. We identify and display in black the closest vertex to each kernel and then traverse the mesh faces to locate triangles whose vertices are associated with three different kernels. These faces mark the corners of the desired Voronoi regions.

We then introduce their barycenter, together with its interpolated normal, as a new vertex of the mesh. This vertex is used to tessellate the face into three sub-triangles, connecting it to the original vertices. Similarly, we add a new vertex to the barycenter of each edge between two vertices linked to distinct cores, inserting it into the mesh by dividing the triangle into two segments. This process produces distinct Voronoi-type regions delimited by dotted lines with a shape error proportional to the length of the triangles' edges in the input mesh (see Fig. 14).

Visualization of nested shapes. Thanks to the artistic inspiration of Fig. 1 (left), particularly in the representation of the vessel's local transparency to reveal the interior at the far left, we have incorporated a similar approach to simplify the design of nested shapes such as cells within a vessel. While all surfaces are partially transparent by default, users can effectively "erase" a section of the membrane with a brush or by intersecting it

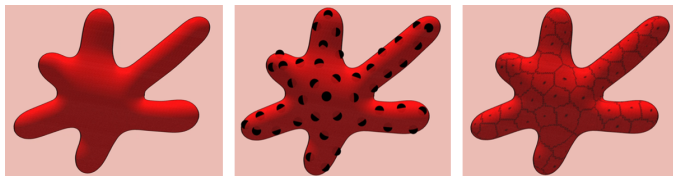


Fig. 14: Organic-looking tessellation of surfaces: (a) shape to texture; (b) user-selected kernels; (c) the resulting tessellation.

with a box. This action makes the interior fully visible, simplifying the sketching of sub-forms. In addition, as shown in the accompanying video, we have simplified the phagocytosis phenomenon by progressively reducing the transparency of the bacteria, then of the white blood cell, which has "absorbed" the bacteria, rather than precisely simulating the absorption itself.

Faking a cut through tissues. Modeling and animating a cross-section of biological tissue presents a significant challenge, particularly due to its anisotropic behavior influenced by fibers and vessels. To tackle this challenge, we used a visualization trick to symbolize a cut-through tissue (see Fig. 15) associated with a temporal event that triggers changes in animation modes.

We provide the user with a cutting tool inspired by a traditional cutter to sketch a cutting trajectory in the 2D visualization plane. Assuming that due to tissue tension, a "real" cut would result in a triangular opening, we simplify the visualization by deactivating the display of all elements inside a triangular prism with an isosceles triangle as a basis (with an angle of $\frac{\pi}{12}$ in our implementation) whose vertex lies at the end of the cut trajectory. To do this, we register all shapes (implicit surfaces and 3D textures such as fibers) intersecting the user's stroke in the projection. We then add other features in the triangular region to this list, using ray scattering at grid-based sample points on the triangular shape, letting us remove all parts projecting into the cut.

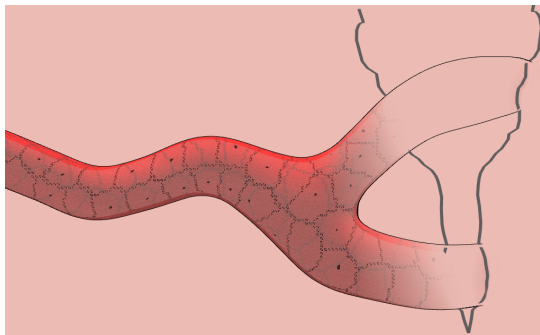


Fig. 15: Rendering effect faking a vertical cut through organs.

6.3. Visualization and navigation within stroke distributions

As mentioned in Sec. 4, most of the static 3D environment, which serves as a background, is constructed using 3D features. We use separate materials for their inner and outer areas for optimal visualization. The central inner area adopts the original color of the stroke, while the contour color changes smoothly from light gray to black, creating a seamless blend with a darker

background. To enhance the three-dimensional appearance of elongated features such as fibers, we create a harmonious blend of inner zone and contour colors, giving the feature a cylindrical appearance.

In our implementation, we use a traveling zoom to move seamlessly from the interactive sketching session, where the user draws the input 2D vector texture to the exploration of the output 3D scene. The volumetric feature distribution layers are presented in each frame with increasing transparency and decreasing feature thickness relative to camera distance. This process is carried out stroke by stroke, depending on the distance between the stroke center point and the camera. In addition, we modify the color of distant strokes by converting their RGB values to HSL format and adjusting the luminance parameter from its original value to 1, thus transforming them into white. When a stroke is located behind the camera, we move it in depth towards the deepest visible volume, creating the illusion of infinite space in depth. This approach delivers superior visual results during exploration compared to applying the same transformation to volumetric layers one by one while maintaining real-time performance (see Fig. 4).

7. Results & Validation

Performances. We implemented our prototype system in WebGL and generated the table below by measuring the runtime performance of Google Chrome on an Intel(R) Core(TM) i7-7920HQ CPU operating at a frequency of 3.10 GHz, with our case study serving as a reference at each stage. Although not optimized, our prototype is thus already convenient for interactive use.

Process	Execution time in sec.
3D field of fibers	0.098
Vessel creation	0.791
Surface texturing	1.13
Cut (on the shader)	Instant
Cell creation (per cell)	0.090
Cell distribution	2.44s
Flow	0.015
Vertical cut event	3
Cell deformation	0.058

User Study. We evaluated our prototype through a user study among experts. The objective was twofold: firstly, after a short user experience, to assess participants' feelings about the relevance of DynBioSketch to facilitate communication and exploration of dynamic biological phenomena; and secondly, to allow them to express themselves on its advantages and disadvantages compared to the illustration tools they are currently using. The second part was challenging as we could not identify any common baseline for informal, interactive illustration and communication among biology experts. Each biologist seemed to have become accustomed to using a different tool, sometimes diverted from its primary use to meet their objectives. Unable to set up a comparative test accessible to all participants, we formulated this part of the experiment as open questions, allowing everyone to compare DynBioSketch with their favorite tools.

Participants: They included 6 women and 5 men aged 30 to 67 years, all working in biological research. Their experience in the field ranged from 5 to 45 years. None of them had participated in our preliminary study or the system design.

Protocol: We first provided a brief introduction to the tool, followed by a 5-minute interactive demonstration in the example of the infection phenomenon. Participants then engaged in hands on experimentation with basic examples (see Fig. 16), allowing them to become familiar with our prototype before exploring their specific target phenomena. These testing sessions were conducted on a 2.5 GHz Windows 10 laptop with 8 Go RAM and lasted approximately 20 minutes. Due to time constraints, the participants only experimented with the 2.5D version of the tool, but we presented them a short demo video of the 3D version in the introduction. Participants concluded their session by completing a survey using the questions listed below.

Questions: Parts of them involved rating, while others were open questions, enabling us to gather more information about the participant's usages and opinions. The first questions (Q1–Q4) focused on evaluating our model independently, while the subsequent ones (Q5–Q8) involved comparing our prototype with the participant's usual practice. To allow that, we asked each participant to specify first which tool(s) they currently used for scientific illustration and communication among peers and to use them in the comparison. The exact list of questions was:

- Q1: Please rate the time it takes to be familiar with DynBioSketch on a scale of 1 (several weeks) to 5 (a few hours).
- Q2: Please rate the ease of creating a full biological environment in DynBioSketch, including modeling individual cells, creating a distribution of cells within a container, sketching a distribution of fibers, and extending it to 3D, on a scale of 1 (very difficult) to 5 (very easy).
- Q3: Please rate the ability of DynBioSketch to convey motion and deformation for the infection phenomenon case, on a scale from 1 (very far from expectations) to 5 (very close to expectations).
- Q4: : Please rate the ability provided by DynBioSketch to specify a triggering event and replay the animation from the event creation time, on a scale of 1 (very difficult) to 5 (very easy).

After having listed the illustration tools you use daily for your thinking or communication with peers, please compare DynBioSketch with your favorite tool(s) by rating the following questions from 1 (strongly disagree) to 5 (strongly agree). Is DynBioSketch better than your favorite tool for:

- Q5: Representing a complex biological environment?
- Q6: Depicting a biological phenomenon (including the narrative part: triggering event and consequences)?

- Q7: Communicating about a biological phenomenon? 52
- Q8: Exploring ideas about a biological phenomenon (such as what might occur between two states, etc.)? 53 54

We concluded the survey with an open question allowing participants to share their thoughts on DynBioSketch 2.5D versus the full 3D modeling and animation for conveying biological phenomena, as well as any additional comments they wished to provide. 55 56 57 58 59

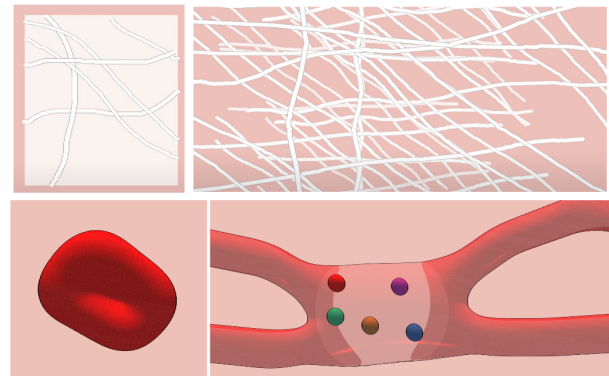


Fig. 16: Some designs from our participants during the user study.

Results of the user-study. We computed the mean and standard deviations of the responses gathered for each question. 60 61

Evaluation of the tool itself (Q1–Q4): Fig.17 illustrates the results of questions Q1–Q4. All users expressed satisfaction with DynBioSketch's user-friendly modeling interface (see the results of Q1 in Fig. 18), describing it as "an easy-to-use tool". The participants also stressed the importance of integrating 3D shape representations to enhance understanding of biological environments (Q2). Moreover, two users specifically advocated for introducing more randomness in synthesizing shape distributions. 62 63 64 65 66 67 68 69 70

The users generally expressed approval of the symbols used to represent motions and were pleased with the resulting movements, yielding a mean score of 4.30 with a standard deviation of 0.6 for Q3 (see Fig.17). They especially valued the kinetic aspect offered by our prototype and highlighted "the ability to specify gradients" as particularly useful. Additionally, one user suggested incorporating collision processing with background fibers could enhance the fidelity between symbols and generated motion. Furthermore, enabling animation within such schematic representations was noted to "help sustain attention" by participants. 71 72 73 74 75 76 77 78 79 80 81

The participants successfully managed the design of a biological phenomenon and replayed the sequence from the triggering event (Q4). One user praised DynBioSketch for enabling them to "express temporal processes to facilitate discussion around a problem" and "communicate their vision of a phenomenon". Additionally, three users identified a notable weakness in the perceived simplicity of our deformation tool. 82 83 84 85 86 87 88

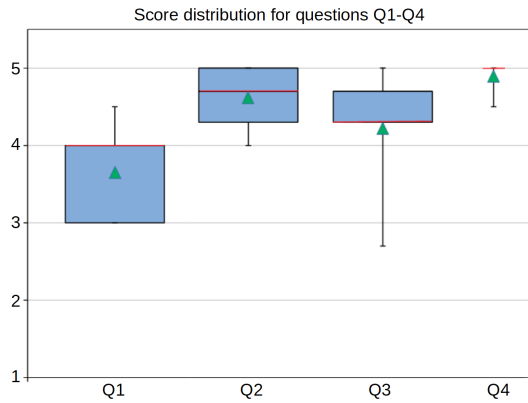


Fig. 17: Score distribution for questions Q1–Q4 evaluating DynBioSketch: the mean is indicated by the green triangle and the median by the red line.

1 Comparisons with tools commonly used by biologists (Q5–Q7):
 2 Fig. 18 presents the comparisons of DynBioSketch with the
 3 tools commonly used by biologists. Ranked by frequency of
 4 use, their current favorite tools are as follows: PowerPoint
 5 (8 votes), Paper and pencil (5 votes), 3D Movies (3 votes),
 6 BioRender (2 votes), Adobe Photoshop (2 votes), Adobe Illus-
 7 trator (1 vote), Microsoft Paint (1 vote), and Artist Illustration
 8 (1 vote).

9 The ability of DynBioSketch to integrate 3D shape repre-
 10 sentations to represent a complex biological environment was
 11 highly praised (Q5) by the participants, particularly noting the
 12 limitations of 2D representations or fixed perspectives inherent
 13 in traditional methods like paper and pencil, as well as soft-
 14 ware such as PowerPoint, Photoshop, and Illustrator. Despite
 15 the availability of pre-defined shapes or built-in functionalities
 16 in such software, users highlighted features missing in our pro-
 17 totype, such as calibration and spatial scale.

18 Participants found that offering a 3D environment, anima-
 19 tion, and narration in DynBioSketch (Q6) surpassed the capa-
 20 bilities of traditional tools like paper and pencil, PowerPoint, or
 21 BioRender, restricting the results to 2D and static representa-
 22 tions.

23 The most noteworthy findings from participants' free com-
 24 ments revealed that biologists found the tool to be "extremely
 25 useful when communicating with each other" and "convenient
 26 to communicate how they see a phenomenon". Despite having
 27 fewer features than professional software, our prototype was
 28 considered more effective in communicating biological phe-
 29 nomena (see Q7 with yellow bars in Fig.18) than commonly
 30 used tools by 82% of users as it enabled the creation of more
 31 complex events than PowerPoint or traditional paper-and-pencil
 32 methods. However, two users suggested that a professionally
 33 produced video might be a better alternative for communi-
 34 cation.

35 Biologists praised DynBioSketch for "helping to express
 36 what they imagined simply", "helping detect inconsistencies in
 37 initial mental models" and "prompting questions that would not
 38 have arisen with flat representations". As illustrated with green
 39 bars in Fig.18, DynBioSketch was considered more convenient
 40 to explore ideas about a biological phenomenon (Q8) than tra-
 41 ditional tools and software.

42 Finally, biologists generally expressed that our tool ad-
 43 dressed a current gap in their toolkit: an interactive platform for
 44 interactively creating 3D animated environments accompanied
 45 by narration. They viewed this concept as a valuable addition
 46 to their resources and particularly highlighted DynBioSketch's
 47 unique ability to represent easily a 3D biological phenomenon.
 48 In the following table (Table 1), we evaluated the following
 49 features: Interactive interface (I), Creation of 3D environment
 50 (3D), Animation (A), and Narration (N).

	I	3D	A	N
DynBioSketch	✓	✓	✓	✓
PowerPoint	✓	P	X	P
Paper and pencil	✓	P	X	P
Movie	X	✓	✓	✓
BioRender	✓	P	X	P
Adobe Photoshop	✓	P	X	P
Adobe Illustrator	✓	P	X	P
Microsoft Paint	✓	P	X	P
Artist's illustration	X	P	X	P

Table 1: Evaluation of DynBioSketch and the tools used by biologists with respect to features. "X" means not handled, "P" means partly handled.

51 The industrial software tools highlighted above (PowerPoint,
 52 BioRender, Adobe Photoshop, Adobe Illustrator, and Microsoft
 53 Paint) are confined to a 2D environment and restrict users to
 54 creating, at most, 3D objects viewed from a fixed perspective
 55 (as denoted by 'P' in Table 1). Furthermore, as these tools lack
 56 animation capabilities, narration is typically limited by labels,
 57 text, or arrows. While movies provide immersion into an envi-
 58 ronment and animation, they lack interactivity.

59 *Final open questions and identified shortcomings:* Through its
 60 interactivity and combination of modeling and animation, Dyn-
 61 BioSketch has unanimously been considered a very promising
 62 tool for exploring ideas on biological phenomena.

63 However, several participants emphasized the importance of
 64 incorporating calibration options similar to those found in pro-
 65 fessional software like BioRender to broaden the utility of Dyn-
 66 BioSketch for a wider range of scenarios. Two of them also
 67 noted that enabling progressive refinement, possibly at other
 68 space and time scales, would have been a great addition to the
 69 current tool.

70 In addition, we were surprised and deceived, to be honest,
 71 by the participants' answers to the last question about their
 72 preference between DynBioSketch in 2.5D and a 3D version.
 73 While we had put a lot of effort into developing our 2.5D ver-
 74 sion, based on non-photorealistic yet expressive modeling, ani-
 75 mation, and rendering, thinking we were offering a good com-
 76 promise between expressivity and simplicity of control, 8 users
 77 among 11 expressed that they preferred the 3D version. They
 78 said that such 3D version was needed to communicate their
 79 phenomenon of study. All, however, noted that the 2.5D ver-
 80 sion was an excellent way to generate simplified representa-
 81 tions used when teaching or to produce the visual summary im-
 82 ages required for each publication in biology. While we respect
 83 their appeal to full 3D to reach more realism, we did not have

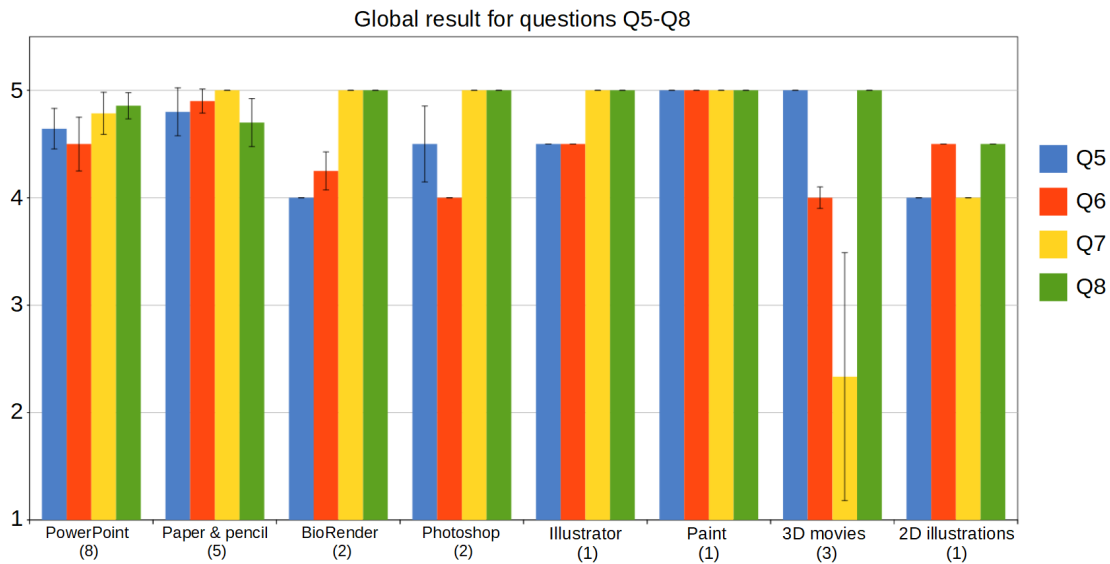


Fig. 18: User-study results for *DynBioSketch* comparison with former tools or usages biologists were familiar with: Each participant was allowed several comparisons, so the number of participants mentioning a given tool is indicated in the bottom line: mean (color) and standard deviation (black) for the answers to Q5-Q8. Note that in addition to the tools they were themselves using, biologists also compared the illustration they get when they ask an artist to design a 3D movie or a 2D illustration for them (two last columns).

enough time during the user study to make them manipulate the full 3D version. Although we think that participants could have changed their minds if they were aware of the difficulty of manually controlling the camera trajectory and zoom factor, in addition to designing a full animated 3D scene, this unexpected result encouraged us to pursue, in future work, the extension our illustration tool to full 3D animated illustrations—probably to be enhanced with some automated camera control.

8. Discussion & Limitations

We presented a complete pipeline for creating, animating, and rendering interactive depictions of biological scenes with a particular focus on representing the phenomenon of infection. Drawing inspiration from illustrations in cell biology, our design choices emulate biological sketching techniques. However, we have also paid attention that our user interface is user friendly for experts in biology and novices alike. Moreover, our system offers users considerable flexibility in modeling various shapes at different scales, thanks to our sketch based system for the 3D implicit modeling system and cutting tool. In terms of animation, we recognize that different phenomena may require incorporating additional motion or deformation behaviors.

While our aim was not to produce exact replicas of the inspiration figure, our tool aimed at complementing existing illustration tools by providing an intermediate between abstract, conceptual sketches, and full 3D animation reproducing or simply visualizing the real phenomena with precision. The images and animations created by our prototype are intended to bring a clear understanding of a phenomenon, which users may enrich with additional annotations, text, or labels.

While not covering every aspect, our designed goals (Section 3.3) were sufficient for the cell biology scenario we investigated, enabling users to recognize *DynBioSketch*'s potential

to create dynamic visual illustrations. However, we identified the following three limitations of our current solutions:

Firstly, as mentioned above, our solution for animating sub-shapes within a 3D container automatically induces spiraling trajectories. While this was considered quite natural by most users, we are not offering any control of this feature, which only depends on the timestep used and the local amount of bending of the container's shape. This twisting motion could be a problem for some applications.

Secondly, while implicit surfaces are a suitable model to create interactively and deform organic-looking shapes, their lack of parametrization made texture mapping problematic. While we resorted to explicit mesh tessellation to create our Voronoi membrane cells, our method is not general and leaves room for improvement. With a goal of better Voronoi-like texture of free-form surfaces, more precision would be needed in terms of sampling kernel positions, and the tessellation computation should use the geodesic distance along the surface, as was done by Meuschke et al. [32].

Lastly, to facilitate real-time tissue cutting, we rendered objects transparent instead of simulating any complex change in their geometry. In future work, representing some local deformations occurring during a "real" cutting event would enable us to convey the phenomenon more convincingly. This would necessitate physics-based volumetric simulation in anisotropic materials, accounting for the tension exerted by fibers and vessels in biological tissues and the subsequent relaxation of these tissues upon cutting. Although realizing real-time animation at this level of detail is challenging, this would be an interesting extension of our work.

At a more conceptual level, biological phenomena are multi-scale, from our scale to molecules through organs and cells. Inspiring from recent work in visualization, enabling multi-scale representation at the design stage, such as letting biologist stop

the animation, zooming on a part of their design, and adding details at other temporal or spatial scales, would be an ambitious yet exciting avenue for future work.

9. Conclusion

We presented DynBioSketch, an interactive platform intended to help biologists express their conceptual understanding of infectious phenomena through animated 3D sketches. It also serves as a tool for non-specialists to explore immune responses to infection. After a prestudy involving biology experts to identify relevant animation forms and tools, we developed a complete pipeline to achieve this goal and implemented a prototype system. We tested the system on a complex scenario: modeling the immune response of tissues to a skin cut. Users managed to create a DynBioSketch accompanied by a narrative within a few minutes. Our user study underscores the need for a system like DynBioSketch as a valuable addition to presentations. Although not comprehensive at present, we anticipate that our innovative system and its future improved extensions will create new opportunities for visual communication, education, and research, extending beyond biology.

In particular, we would like to extend DynBioSketch's capabilities to the representation of other phenomena, with a particular focus on multi-resolution sketching. This capability would empower users to dynamically create and zoom into animated scientific illustrations with virtually unlimited levels of detail. Similar to the approach proposed by Wither and al. [51], integrating machine learning techniques would help facilitate the reproduction of sketch-based detail in different sections of a media form. This functionality is essential for seamless navigation and zoom-out functions.

CRedit authorship contribution statement

Pauline Olivier: Conceptualization, Methodology, Software, Validation, Formal Analysis, Writing - Original draft, Writing - Review and editing. **Tara Butler:** Methodology, Writing - Original draft. **Pascal Guehl:** Methodology, Software, Writing - Review and editing. **Jean-Luc Coll:** Validation, Writing - Review and editing. **Renaud Chabrier:** Conceptualization, Validation, **Pooran Memari:** Conceptualization, Methodology, Formal Analysis, Writing - Review and editing, **Marie-Paule Cani:** Conceptualization, Methodology, Supervision, Project administration, Writing - Review and editing

Declaration of competing interest

The authors assert that they do not have any identifiable financial conflicts of interest or personal relationships that might have been perceived to influence the work reported in this paper.

Acknowledgments

We are grateful to the reviewers for their useful comments and remarks that helped us improve the paper. We are thankful to the researchers from IAB Grenoble for their time, interest in

this project, and participation in our user study. This paper was partly funded by Marie-Paule Cani's fellowship on Creative AI from Hi!Paris.

References

- Chabrier, R. *Globule, the magazine of all cells*. 2018. ISBN 978-2-9563725-0-9.
- Bernhardt, A, Pihuit, A, Cani, MP, Barthe, L. Matisse: Painting 2d regions for modeling free-form shapes. In: *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling*. ISBN 9783905674071; 2008, p. 57–64. doi:10.2312/SBM/SBM08/057-064.
- Zanni, C, Bernhardt, A, Quiblier, M, Cani, MP. SCALE-invariant Integral Surfaces. *Computer Graphics Forum* 2013;32(8):219–232. doi:10.1111/cgf.12199.
- Olivier, P, Chabrier, R, Memari, P, Coll, JL, Cani, MP. Bio-Sketch: A New Medium for Interactive Storytelling Illustrated by the Phenomenon of Infection. In: *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association. ISBN 978-3-03868-216-5; 2023, p. 63–7412 pages. doi:10.2312/vcbm.20231214.
- Viola, I, Gröller, ME, Hadwiger, M, Bühler, K, Preim, B, Ebert, D. Illustrative Visualization. In: *Eurographics 2005 - Tutorials*. 2005, doi:10.2312/egt.20051052.
- Isenberg, T. A Survey of Illustrative Visualization Techniques for Diffusion-Weighted MRI Tractography. In: *Visualization and Processing of Higher Order Descriptors for Multi-Valued Data*. ISBN 978-3-319-15090-1; 2015, p. 235–256. doi:10.1007/978-3-319-15090-1_12.
- Lawonn, K, Viola, I, Preim, B, Isenberg, T. A Survey of Surface-Based Illustrative Rendering for Visualization. *Computer Graphics Forum* 2018;37(6):205–234. doi:10.1111/cgf.13322.
- Preim, B, Saalfeld, P. A survey of virtual human anatomy education systems. *Computers & Graphics* 2018;71:132–153. doi:10.1016/j.cag.2018.01.005.
- Abdrashitov, R, Bang, S, Levin, D, Singh, K, Jacobson, A. Interactive Modelling of Volumetric Musculoskeletal Anatomy. *ACM Trans Graph* 2021;40(4):1–13. doi:10.1145/3450626.3459769.
- Pihuit, A, Cani, MP, Palombi, O. Sketch-based modeling of vascular systems: A first step towards interactive teaching of anatomy. In: *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*. ISBN 9783905674255; 2010, p. 151–158. doi:10.2312/SBM/SBM10/151-158.
- Saalfeld, P, Baer, A, Preim, U, Preim, B, Lawonn, K. A sketch-based interface for 2d illustration of vascular structures, diseases, and treatment options with real-time blood flow. In: *Computer Vision, Imaging and Computer Graphics Theory and Applications*. ISBN 978-3-319-29971-6; 2016, p. 19–40. doi:10.1007/978-3-319-29971-6_2.
- Saalfeld, P, Stojnic, A, Preim, B, Oeltze-Jafra, S. Semi-Immersive 3D Sketching of Vascular Structures for Medical Education. In: *Proceedings of the Eurographics Workshop on Visual Computing for Biology and Medicine*. ISBN 9783038680109; 2016, p. 123–132. doi:10.5555/3061507.3061528.
- Wei, LY, Lefebvre, S, Kwatra, V, Turk, G. State of the Art in Example-based Texture Synthesis. In: *Eurographics 2009 - State of the Art Reports*. 2009, doi:10.2312/egst.20091063.
- Sendik, O, Cohen-Or, D. Deep Correlations for Texture Synthesis. *ACM Trans Graph* 2017;36(5):1–15. doi:10.1145/3015461.
- Tu, P, Lischinski, D, Huang, H. Point Pattern Synthesis via Irregular Convolution. *Computer Graphics Forum* 2019;38(5):109–122. doi:10.1111/cgf.13793.
- Reddy, P, Guerrero, P, Fisher, M, Li, W, Mitra, NJ. Discovering pattern structure using differentiable compositing. *ACM Trans Graph* 2020;39(6):1–15. doi:10.1145/3414685.3417830.
- Landes, PE, Galerne, B, Hurtut, T. A shape-aware model for discrete texture synthesis. *Computer Graphics Forum* 2013;32(4):67–76. doi:10.1111/cgf.12152.
- Barla, P, Breslav, S, Thollot, J, Sillion, F, Markosian, L. Stroke pattern analysis and synthesis. *Computer Graphics Forum* 2006;25(3):663–671. doi:10.1111/j.1467-8659.2006.00986.x.
- Ijiri, T, Mèch, R, Igarashi, T, Miller, G. An Example-based Procedural System for Element Arrangement. *Computer Graphics Forum* 2008;27(2):429–436. doi:10.1111/j.1467-8659.2008.01140.x.

- [20] Hurtut, T, Landes, PE, Thollot, J, Gousseau, Y, Drouillhet, R, Coeurjolly, JF. Appearance-Guided Synthesis of Element Arrangements by Example. In: Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering. ISBN 9781605586045; 2009, p. 51–60. doi:10.1145/1572614.1572623.
- [21] Ma, C, Wei, LY, Tong, X. Discrete Element Textures. *ACM Trans Graph* 2011;30(4):1–10. doi:10.1145/2010324.1964957.
- [22] Roveri, R, Öztireli, AC, Martin, S, Solenthaler, B, Gross, M. Example Based Repetitive Structure Synthesis. *Computer Graphics Forum* 2015;34(5):39–52. doi:10.1111/cgf.12695.
- [23] Olivier, P, Memari, P, Cani, MP. Structured shape-patterns from a sketch: A multi-scale approach. In: Proceedings of Graphics Interface 2022. 2022, p. 161 – 170. doi:10.20380/GI2022.17.
- [24] Zhu, B, Iwata, M, Haraguchi, R, Ashihara, T, Umetani, N, Igarashi, T, et al. Sketch-based dynamic illustration of fluid systems. *ACM Trans Graph* 2011;30(6):1–8. doi:10.1145/2070781.2024168.
- [25] Kazi, RH, Chevalier, F, Grossman, T, Zhao, S, Fitzmaurice, G. Draco: Bringing Life to Illustrations with Kinetic Textures. In: Proceedings of the SIGCHI. CHI '14. ISBN 9781450324731; 2014, p. 351–360. doi:10.1145/2556288.2556987.
- [26] Kazi, RH, Chevalier, F, Grossman, T, Fitzmaurice, G. Kitty: Sketching Dynamic and Interactive Illustrations. In: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology. ISBN 9781450330695; 2014, p. 395–405. doi:10.1145/2642918.2647375.
- [27] Milliez, A, Noris, G, Baran, I, Coros, S, Cani, MP, Nitti, M, et al. Hierarchical motion brushes for animation instancing. In: Proceedings of the Workshop on Non-Photorealistic Animation and Rendering. ISBN 9781450330206; 2014, p. 71–79. doi:10.1145/2630397.2630402.
- [28] Milliez, A, Guay, M, Cani, MP, Gross, M, Sumner, RW. Programmable Animation Texturing using Motion Stamps. *Computer Graphics Forum* 2016;35(7):67–75. doi:10.1111/cgf.13004.
- [29] Xing, J, Kazi, RH, Grossman, T, Wei, LY, Stam, J, Fitzmaurice, G. Energy-Brushes: Interactive Tools for Illustrating Stylized Elemental Dynamics. In: Proceedings of the 29th Annual Symposium on User Interface Software and Technology. ISBN 9781450341899; 2016, p. 755–766. doi:10.1145/2984511.2984585.
- [30] Goodsell, DS. The machinery of life 2009;doi:10.1007/978-0-387-84925-6.
- [31] Cakmak, E, Jäckle, D, Schreck, T, Keim, DA, Fuchs, J. Multiscale visualization: A structured literature analysis. *IEEE Transactions on Visualization and Computer Graphics* 2022;28(12):4918–4929. doi:10.1109/TVCG.2021.3109387.
- [32] Meuschke, M, Voß, S, Beuing, O, Preim, B, Lawonn, K. Glyph-based comparative stress tensor visualization in cerebral aneurysms. *Computer Graphics Forum* 2017;36(3):99–108. doi:10.1111/cgf.13171.
- [33] Garrison, LA, Goodsell, DS, Bruckner, S. Changing aesthetics in biomolecular graphics. *IEEE Computer Graphics and Applications* 2023;43(3):94–101. doi:10.1109/MCG.2023.3250680.
- [34] Garrison, L, Meuschke, M, Fairman, J, Smit, NN, Preim, B, Bruckner, S. An Exploration of Practice and Preferences for the Visual Communication of Biomedical Processes. In: Eurographics Workshop on Visual Computing for Biology and Medicine. ISBN 978-3-03868-140-3; 2021, p. 1–12. doi:10.2312/vcbm.20211339.
- [35] Gershon, N, Page, W. What storytelling can do for information visualization. *Commun ACM* 2001;44(8):31–37. URL: <https://doi.org/10.1145/381641.381653>. doi:10.1145/381641.381653.
- [36] Tong, C, Roberts, R, Borgo, R, Walton, S, Laramée, RS, Wegba, K, et al. Storytelling and visualization: An extended survey. *Information* 2018;9(3). doi:10.3390/info9030065.
- [37] Mörth, E, Bruckner, S, Smit, NN. Scrollyvis: Interactive visual authoring of guided dynamic narratives for scientific scrollytelling. *IEEE Transactions on Visualization and Computer Graphics* 2022;1–12doi:10.1109/TVCG.2022.3205769.
- [38] Chen, Q, Cao, S, Wang, J, Cao, N. How does automation shape the process of narrative visualization: A survey of tools. *IEEE Transactions on Visualization and Computer Graphics* 2023;1–20doi:10.1109/TVCG.2023.3261320.
- [39] Meuschke, M, Garrison, L, Smit, N, Bruckner, S, Lawonn, K, Preim, B. Towards narrative medical visualization. 2021. [arXiv:2108.05462](https://arxiv.org/abs/2108.05462).
- [40] Kleinau, A, Stupak, E, Mörth, E, Garrison, LA, Mittenentzwei, S, Smit, NN, et al. Is there a Tornado in Alex’s Blood Flow? A Case Study for Narrative Medical Visualization. In: Eurographics Workshop on Visual Computing for Biology and Medicine. ISBN 978-3-03868-177-9; 2022,doi:10.2312/vcbm.20221183.
- [41] Meuschke, M, Garrison, LA, Smit, NN, Bach, B, Mittenentzwei, S, Weiß, V, et al. Narrative medical visualization to communicate disease data. *Comput Graph* 2022;107(C):144–157. doi:10.1016/j.cag.2022.07.017.
- [42] Garrison, LA, Meuschke, M, Preim, B, Bruckner, S. Current Approaches in Narrative Medical Visualization. Springer Nature Switzerland. ISBN 978-3-031-41652-1; 2023, p. 95–116. doi:10.1007/978-3-031-41652-1_4.
- [43] Whitby, M, Berry, D. Molecular visualizations of dna. In: ACM SIGGRAPH 2003 Video Review on Electronic Theater Program on Electronic Theater Program - Volume 144. ISBN 9781450377829; 2003, p. 4. doi:10.1145/1006032.1006036.
- [44] McGill, G. Molecular movies... coming to a lecture near you. *Cell* 2008;133(7):1127–1132. doi:10.1016/j.cell.2008.06.013.
- [45] Gardner, A, Autin, L, Barbaro, B, Olson, AJ, Goodsell, DS. Cellpaint: Interactive illustration of dynamic mesoscale cellular environments. *IEEE Computer Graphics and Applications* 2018;38(6):51–66. doi:10.1109/MCG.2018.2877076.
- [46] Gardner, A, Autin, L, Fuentes, D, Maritan, M, Barad, BA, Medina, M, et al. Cellpaint: Turnkey illustration of molecular cell biology. *Frontiers in bioinformatics* 2021;1:660936. doi:10.3389/fbinf.2021.660936.
- [47] Kouřil, D, Strnad, O, Mindek, P, Halladjian, S, Isenberg, T, Gröller, ME, et al. Molecumentary: Adaptable narrated documentaries using molecular visualization. *IEEE Transactions on Visualization and Computer Graphics* 2023;29(3):1733–1747. doi:10.1109/TVCG.2021.3130670.
- [48] Witko-Sarsat, V, Rieu, P, Descamps-Latscha, B, Lesavre, P, Halbwachs-Mecarelli, L. Neutrophils: Molecules, functions and pathophysiological aspects. *Laboratory Investigation* 2000;80:617–653. doi:10.1038/labinvest.3780067.
- [49] Stylianopoulos, T, Munn, LL, Jain, RK. Reengineering the Physical Microenvironment of Tumors to Improve Drug Delivery and Efficacy: From Mathematical Modeling to Bench to Bedside. *Trends in Cancer* 2018;4(4):292–319. doi:10.1016/j.trecan.2018.02.005; special Issue: Physical Sciences in Oncology.
- [50] Gamara, J, Chouinard, F, Davis, L, Aoudjit, F, Bourgoin, S. Regulators and effectors of arf gtpases in neutrophils. *Journal of Immunology Research* 2015;2015:1–15. doi:10.1155/2015/235170.
- [51] Wither, J, Boudon, F, Cani, MP, Godin, C. Structure from silhouettes: a new paradigm for fast sketch-based design of trees. *Computer Graphics Forum* 2009;28(2):541–550. doi:10.1111/j.1467-8659.2009.01394.x.
- [52] Galin, E, Akkouche, S. Blob Metamorphosis based on Minkowski Sums. *Computer Graphics Forum* 1996;15(3):143–153. doi:10.1111/1467-8659.1530143.