



Agent-Based Principal Strips Modeling for Freeform Surfaces in Architecture

Hua Chai¹ · Luis Orozco^{2,3} · Fabian Kannenberg^{2,3} · Lasath Siriwardena^{2,3} · Tobias Schwinn^{2,3} · Hanning Liu¹ · Achim Menges^{2,3} · Philip F. Yuan¹

Accepted: 28 January 2024 / Published online: 26 February 2024
© The Author(s) 2024

Abstract

The principal curvature (PC) of a freeform surface, as an important indicator of its fundamental features, is frequently used to guide their rationalization in the field of architectural geometry. The division of a surface using its PC lines into principal strips (PSs) is an innovative way to break down a freeform surface for construction. However, the application of PC networks in architectural design is hindered by the difficulty to generate them and flexibly control their density. This paper introduces a method for PS-based reconstruction of freeform surfaces with different umbilical conditions in the early stages of design. An agent-based modeling approach is developed to find the umbilics and increase the degree of control over the spacing of PC lines. This research can effectively expand the application range of PS-based surface reconstruction methods for freeform architectures.

Keywords Principal strip · Geometry · Agent-based modeling · Freeform surface · Umbilics

Introduction

With the support of digital design and construction technology, architects have increasingly favored freeform surfaces since the end of the twentieth century (Carpo 2012). Supporting structures, building skins, and decorations with freeform surfaces have held a common place in architectural practice. However, the realization of freeform surfaces in architecture has consistently posed a significant challenge (Eigensatz et al. 2010). In practice, designers have developed a series of feasible methods for constructing freeform surfaces by weighing different factors such as material properties, geometry, structure, and costs. These methods involve discretizing or semi-discretizing the surface, enabling construction with repeatable molds or planar sheet materials (Schling and Wan 2022), thereby reducing production costs and installation difficulty. Common discretization methods include

fitting surfaces with triangular meshes or planar tessellation (Pottmann et al. 2007). Semi-discretization methods include contour slicing of surfaces (Kirschman and Jara-Almonte 1992) and fitting surfaces with developable strips (Pottmann 2013; Gavriil et al. 2019; Kahlert et al. 2011).

Encapsulating fundamental geometric features of surfaces, principal curvature (PC) lines offer distinct advantages in guiding the rationalization of freeform surfaces. PC lines constitute a network of conjugate and orthogonal curves that align with maximum and minimum normal curvature directions, which has been employed in architectural applications to partition freeform surfaces into quad meshes for paneling (Pottmann 2013; Schiffner et al. 2013). PC-based surface approximation has the potential to enhance various aspects of architectural projects. The geodesic torsion of PC lines is always 0 (Oberbichler et al. 2023), which implies the existence of torsion-free support structures and nodes. When designing support beams for freeform surfaces, maintaining the alignment of the beams' axes with the PC lines and ensuring that the rectangular cross section remains tangential or orthogonal to the surface can result in minimize torsion in the beams (Tang et al. 2016). As a rule of thumb in engineering mechanics, 'higher stresses occur in higher curvature regions' (Fischer-Cripps 2007), which means that the PC directions can also be used to guide structural reinforcement.

In the field of computer graphics, numerous studies already offered fully automatic methods for segmenting a surface into principal patches (Kälberer et al. 2007; Bommès et al. 2009; Evolute GmbH 2021). However, these studies often partition the surface into flat quads directly, with a low degree of control for designers over the outcome. This level of automation is excessive for certain scenarios in architectural applications, particularly in cases that rely on strip-based materials such as bending-active construction systems (Lienhard 2014). In this context, principal strip (PS), which is derived from semi-discretizing a surface with edge curves following the maximal curvature, provides an effective way to construct freeform surfaces. The concept of PS was put forward by Pottmann et al. (2008) to semi-discretize a surface into developable strips, where the 'edge curves follow the principal curvature lines of maximal curvature and to place rulings along the directions of the smaller principal curvature.' PS-based approximation managed to optimize a freeform surface into singly curved conical and circular panels using its PC line network. While Pottmann et al. (2008) 'restricted the models of the principal strip to circular and conical models, which are convenient for panelization', Takezawa et al. (2016) extended the concept of PS by directly unfolding the double-curved strips onto a plane without pre-optimizing them into flat panels. This method was used to fabricate freeform geometry by weaving two layers of PSs together, such as double curved CFRP shells whose fiber directions align along their PC (Takezawa et al. 2021).

Research Question and Aims

When exploring the practical application of PS in architectural design, it becomes essential to seek a balance between automated generation and intervention

based on diverse design considerations from architects. A uniform distribution of PC lines is frequently preferred, which can be generated with the assistance of differential geometry (Joo et al. 2014), or iterative algorithms supplemented by manual intervention (Takezawa et al. 2016). In architectural design, an increased level of control and customization over the generation process of PSs is required to incorporate diverse design goals such as surface smoothness and raw material dimensions. The current design tools available to architects, taking parametric design platform Rhinoceros 3D (McNeel et al. 2023a) and Grasshopper (McNeel et al. 2023b) as an example, only provide curvature line generation based on user-defined points (Vestartas 2018; Greco 2019; Oberbichler 2021), which lack the capability to perform PC line network generation and density control.

In this context, this research aims to provide architects with a method for generating and regulating PSs in the early stages of design. The proposed method does not only facilitate architects' understanding of freeform surfaces through topological skeleton analysis, but also offers a higher degree of control over the generated PSs, allowing for the incorporation of diverse design goals by introducing agent-based modeling (ABM) approach.

Methodology

An ABM approach is increasingly used to manage the complexity inherent in architectural design and construction processes (Stieler et al. 2022). An agent can simulate the behavior of a specific geometric element or building component under various constraints (Snooks 2022). ABM is well suited to deal with the complexity brought on by the comprehensive consideration of various factors, including but not limited to material properties, structural performance, and fabrication constraints by translating those diverse constraints into agent behaviors (Groenewolt et al. 2018). The environment of the agent system can be a freeform surface such as shell structures (Schwinn et al. 2014; Menges et al. 2022) and building facades (Gerber et al. 2017). ABM is also used to optimize surface performance by associating the structural or environmental features of a surface with agent behaviors (Chai et al. 2022).

In this study, ABM is employed to provide architects with an intuitive way of finding the singularities of freeform surfaces and regulating the layout of the PC lines by associating agent behaviors to the curvature of the surface. The customizable behavior definitions inherent in ABM systems, which can directly relate to diverse architectural considerations in terms of geometry, performance, or construction, enable further flexibility in PSs density control. The iterative nature inherent in ABM methods limits complete automation in the design process and tends to increase computation times. Nevertheless, this approach could enhance architects' comprehension and control over the generation process of PSs by translating a variety of constraints into purely geometric behaviors of agents.

Main Contributions and Limitations

The contributions of this paper can be summarized as follow:

- We develop an ABM-based approach to find the singularities of freeform surfaces and reconstruct surfaces with principal strips, which can be used for constructing freeform roofs, facades, and shell structures from planar sheet materials.
- Our approach introduces the topological skeleton of the surface by identifying singular points on the surface, making the generation process more intuitive.
- This approach not only enables the generation of an even distribution of PC Lines or PSs, but also allows for the integration of diverse customized design objectives, thereby enhancing design flexibility.

The method proposed in this study is specifically designed for generating PSs on freeform surfaces without planar regions. Although PC line network can also serve as guidance for paneling, it is important to note that this research does not aim to provide a quad mesh method.

Paper Structure

This paper is organized as follows: Sect. 1 gives an overview of the topic, questions, aims and methodology. Section 2 provides an explanation about the geometrical features of umbilics on freeform surfaces, which serve as the primary obstacles in designing uniformly distributed PSs. Section 3 introduces the research method and workflow. The software framework and the generation method of the PC lines are also described as the basis of the research. Section 4 details the proposed ABM algorithm to control the distribution of PC lines on free-form surfaces. Section 5 introduces the possible application scenarios for this method in architectural design. The method is also extended to other principal trajectories, such as principal stress lines. The last section summarizes the value of the research achievements and proposes directions for future research.

Surface Umbilics

While PC lines typically form a network of conjugate and orthogonal curves, the presence of umbilical points introduces variations in network topologies (Fig. 1). An umbilical point, also known as an umbilic, is a special kind of point on the surface at which the curvatures of the surface are equal in all directions. In other words, the surface is spherical at that point. At an umbilical point both principal curvatures are equal, and every tangent vector is a principal direction (Berry and Hannay 1977). As a result, the PC line network also appears differently around the umbilical point.

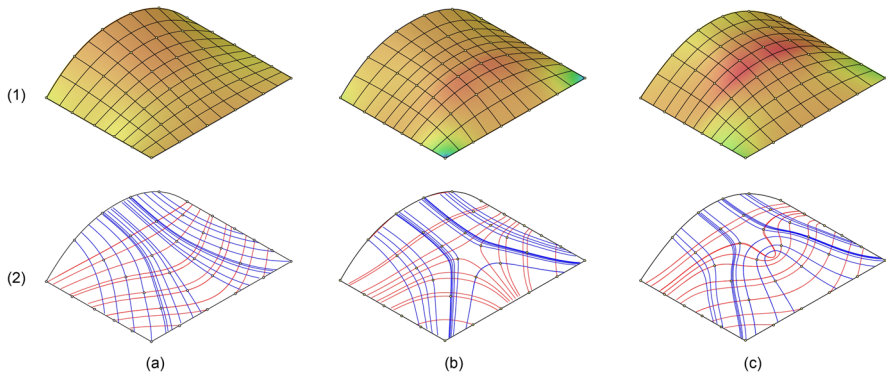


Fig. 1 (1) three surfaces with different mean curvatures are used to visualize (2) PC line networks based on uniform seeding points. Simple changes in the surface shapes resulted in different umbilical features

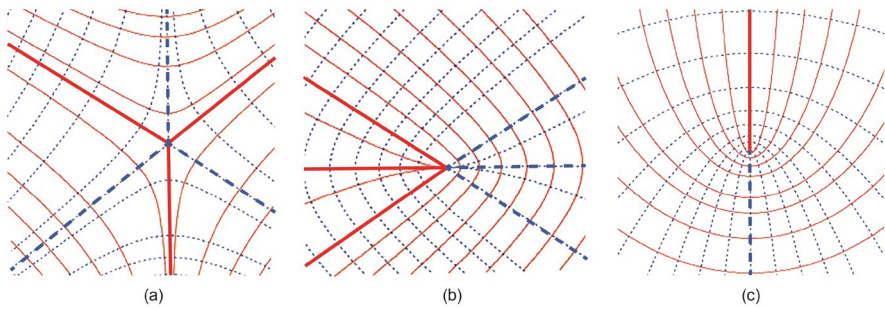


Fig. 2 Three basic types of surface umbilics: star (a); lemon (b); and monstar (c). Image: based on Patrikalakis and Maekawa 2002

There are three basic types of umbilics on surfaces: star, lemon, and monstar (Fig. 2) (Maekawa et al. 1996). The configurations of PC lines near umbilics can also be categorized into two basic types, trisector and wedge points, according to their underlying tensor fields. A trisector point corresponds to a star umbilic, while a wedge point can be either lemon or monstar. The straight PC lines passing through a trisector point would appear as three branches starting from the trisector, as the bold lines shown in Fig. 2a. The straight PC lines passing through a wedge point could be either three (Fig. 2b) or one (Fig. 2c). Since the topology only changes at umbilics, the straight PC lines passing through umbilics can be used to describe the topological structure of a surface (Delmarcelle and Hesselink 1994). The detailed explanation of the straight lines and the tensor fields behind them can refer to Delmarcelle (1995).

The presence of umbilics poses a primary obstacle to the generation and control of PSs. PC lines passing through the umbilics no longer form a single continuous curve. At the same time, a notable directional variation is apparent in the PC lines in the vicinity of umbilical points. The topological differences

arising from umbilics make it challenging to find a universal algorithm for generating PC line network.

Design Methods

Density control for orthogonal conjugate networks is easy to implement. PC lines passing through umbilics can be viewed as the topological skeleton, which could divide the surface into a series of regions in which the PC lines always form an orthogonal network. Therefore, an important step before PS-based surface reconstruction is to find the umbilics of the surface. The ABM-based regulation of the density of PC lines will be carried out inside the subdivided regions.

Workflow

The proposed PS-based surface reconstruction is carried out through two core steps, each with its own agent simulation, and described above in the introduction to umbilics. The first agent system finds the position of the umbilical points on the surface. The topological skeleton is derived from the umbilics to divide the surface into a series of regions. The second agent system draws and optimizes the layout of PC lines in the regions with another agent system according to the user-defined design goals. A curve in each region is selected as a seed curve, which will be used to generate a series of seed points and act as a moving rail for the seed points. The seed points, used as agents, move on the seed curve to generate and adjust the spacing of PC lines to achieve certain design goals.

Software and Framework

Geometric Modeling Platform

To better assist the architectural design process, Rhinoceros 3D and Grasshopper are used for calculations, geometric modeling, and visualization. The development of agent-based simulation is also programmed based on the RhinoCommon library of Rhino3D.

ABM Framework

The program design of ABM in this research is based on the ABxM framework developed by ICD, Stuttgart, which is an open-source platform for exploration with agent-based systems (Nguyen et al. 2022). ABxM includes an agent library called ABxM.Core.dll, and a Grasshopper plugin. The core content of the agent system in this research, including customized agents, agent systems and behaviors, is developed in Microsoft Visual Studio based on ABxM.Core.dll. The solver in the ABxM plug-in in Grasshopper is called to iteratively execute the behaviors of all agents in the agent systems.

Drawing PC Lines

Accurately generated PC lines forms the basis of this study. Before calculating the PC lines, the 'Smooth' command of Rhino3D is used to smooth the surface while maintaining its boundaries. This avoids the excessive change in PC line density caused by sharp folds in the surface.

A PC line is drawn iteratively finding approximate directions starting from a seed point. First, the two principal curvatures and principal directions at the current seed point are obtained. Then the Fourth Order-Runge Kutta method (also known as RK4 method) is used to deduce the next seed point at the given step size to the current one. A PC line is formed by sequentially connecting all sample points. As can be seen in Fig. 3-1 and -2, for every sample point not on the edge of the surface, the PC line is the result of joining two lines starting from this point towards the principal direction and the opposite direction.

There are two important aspects of this algorithm (Fig. 3): first is the situation at the edge where the distance from the last seed point to the edge is smaller than the given step size. In this case, the intersection point of the PC line with the edge is adopted (point P in Fig. 3-3). Second is the case when the PC line appears as a closed loop. Since the calculation algorithm is an iterative approximation, it is almost impossible to close the curve. The strategy adopted here is to calculate the distance between the new seed point and the starting point at the end of each iteration and close the curve directly if the distance is less than a given tolerance (in this case, the given step size). A smooth closed loop can be achieved with a relatively small step size.

ABM-Based PC Lines Layout

Two agent systems have been developed, each corresponding to one of the two steps of the ABM-based PC lines layout optimization method. They are (1) the umbilic agent system and (2) the strip agent system. The umbilic agent system is used to find umbilical points on the surface, while the strip agent system is used to regulate the width of the resulting PSs. In order to clearly show the logic of the agent-based algorithm, a simple free-form surface with one trisector point is taken as an example in this section (Fig. 1-b).

Umbilic Agent System

The key to finding umbilics with ABM is to convert the unique features of umbilics into goals that can be gradually approached by agents. Assuming that the curvatures at a seed point on the surface are $C_{\max}$ and $C_{\min}$ respectively, and then the absolute value of the difference between $C_{\max}$ and $C_{\min}$ is ΔC . Since the two principal curvatures are equal at the umbilical point, ΔC at an umbilic is 0. Therefore, the problem of finding umbilics can be transformed into a minimization problem: find

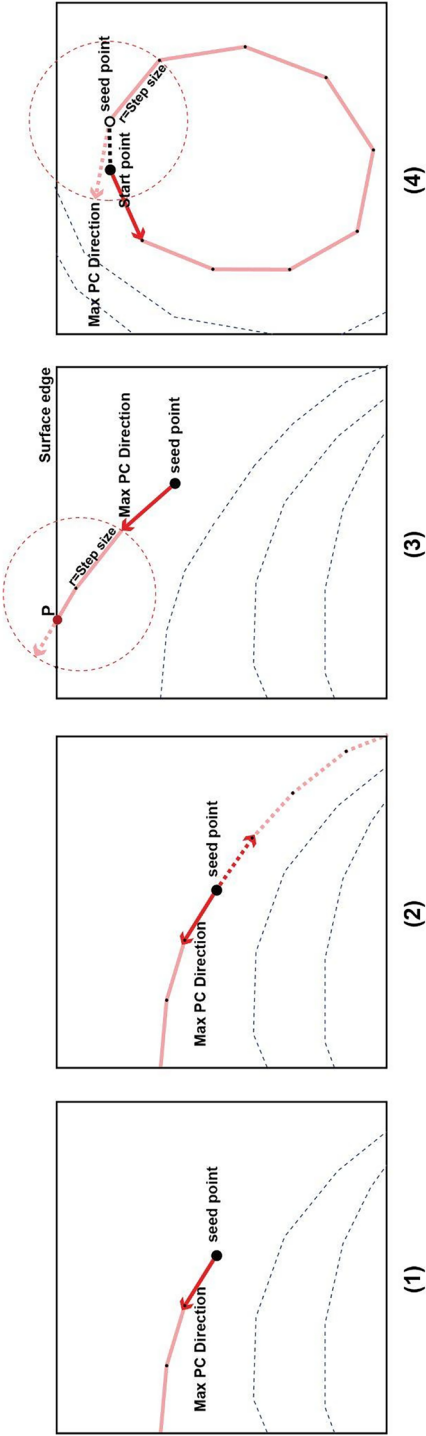


Fig. 3 Drawing a PC line: a PC line is the join of two lines in opposite directions (1–2); edge situation (3); loop situation (4)

the minimum ΔC , $\Delta C_{\min}$, as a function of the PC values at point (U, V) in the surface's parameter domain. Assume $C_{\max}$ and $C_{\min}$ are both functions of U and V value.

$$C_{\max} = f_1(U, V) \quad (1)$$

$$C_{\min} = f_2(U, V) \quad (2)$$

$$\Delta C = |C_{\max} - C_{\min}| \quad (3)$$

$$\Delta C_{\min} = \min_{0.0 \leq U, V \leq 1.0} |f_1(U, V) - f_2(U, V)| \quad (4)$$

Agent Definition

Umbilic agents are instantiated as random points on the environment surface and take their coordinates in the surface's parameter domain (U, V) as their position attribute. The agents will move in the parameter space of the surface towards umbilics. For each iteration, the agent will get a moving vector V_m that is a fixed ratio k of the product of the unit vector v in one of the principal directions and the step size s , weighted by ΔC at the agent's UV.

$$V_m = k \Delta C s v \quad (5)$$

The distance an agent moves at each iteration of the agent simulation therefore decreases proportionally as it approaches an umbilic.

Each agent also has a Boolean attribute 'isFinished' that is used to determine whether an umbilic has been found. The 'isFinished' attribute is false by default. Once the length of V_m is less than a preset tolerance, the agent is considered to have reached an umbilic. 'isFinished' is then set to true, and the agent will not be calculated in the next iteration.

Agent Behavior

To intuitively observe the curvature of the surface and the movement of the agents, a 3D contour map is established with x and y values matching the environment surface, but with ΔC as the Z-axis parameter. The environment surface and the contour map can be overlapped in plan to obtain an intuitive view of the relationship between the point position and ΔC (Fig. 4). It should be noted that the points on the contour map with the lowest z value, especially those approaching $z=0$, are the approximate locations of the umbilics.

The umbilic agent's behavior finds v , the agent's movement vector. Assuming we are at a point where PCs are different, the objective is to find which direction the agent should move in order to reduce the difference between PCs. As the surface environment's curvature and PC directions are known, the agent can

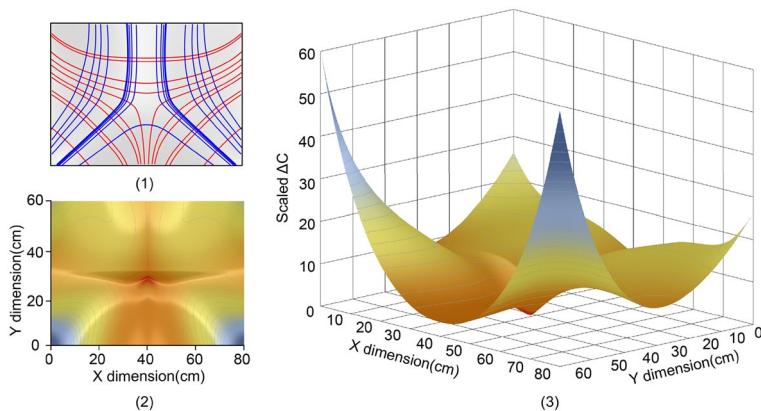


Fig. 4 3D contour map of ΔC (scaled from [0.000214, 0.08763] to [0, 60])

test and compare the resultant ΔC of a small step in each of the four principal directions. The direction v is the one in which the behavior measures the greatest reduction in ΔC (Fig. 5).

Each agent draws a polyline trail on the environment surface with all the positions it has passed (Fig. 6). This agent trajectory shows that the agent finally stops at the bottom of the valley, with a gradually decreasing amplitude of motion. After that, the 'isFinished' attribute of the agent will switch to true. Although higher accuracy can be achieved, the points that are eventually found by the agent simulation are still only approximations of the real umbilics.

Once an agent 'isFinished', two things need to be identified. First, it needs to be judged whether the agent has reached a real umbilic. It is possible for the agent to find a local optimum, rather than an umbilic point, depending on the location of the starting point. Since the ΔC value of an umbilical point is zero, it can be judged whether an 'isFinished' agent is close to the global optimal solution by comparing the ΔC value to zero. Second, if the agent finds a true umbilic, then whether it's a trisector point and wedge point needs to be judged. The number of intersection points of the two PC lines passing through the agent can be used to distinguish the type of umbilic point: if the agent is close to a wedge point, there will be two intersections of the PC lines, and if it is close to a trisector point, there will only be one intersection (Fig. 7). The umbilical points filtered out will be used to prepare the environment required by the strip agent system.

When working in Grasshopper, designers have the flexibility to either generate seeds randomly for the umbilic agents, or manually specify the seed's location on the surface through an interactive process by looking at the contour map of ΔC . The presence of the contour map serves to visually present the approximate locations of all umbilics to the designer, ensuring that the agents do not miss any of the umbilics.

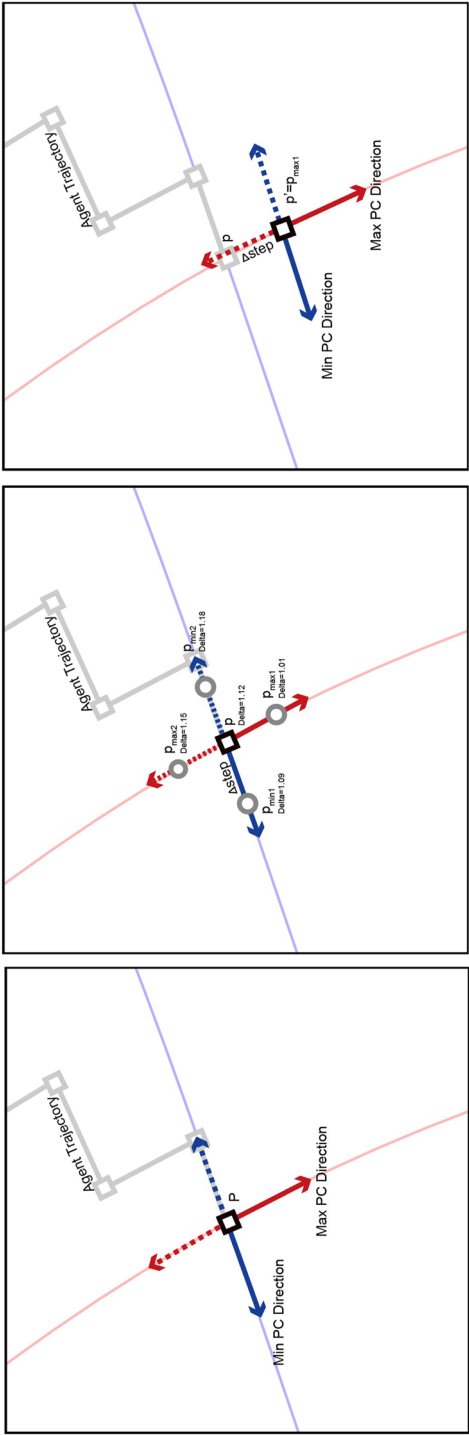


Fig. 5 The umbilic agent's behavior is to find the direction of its movement

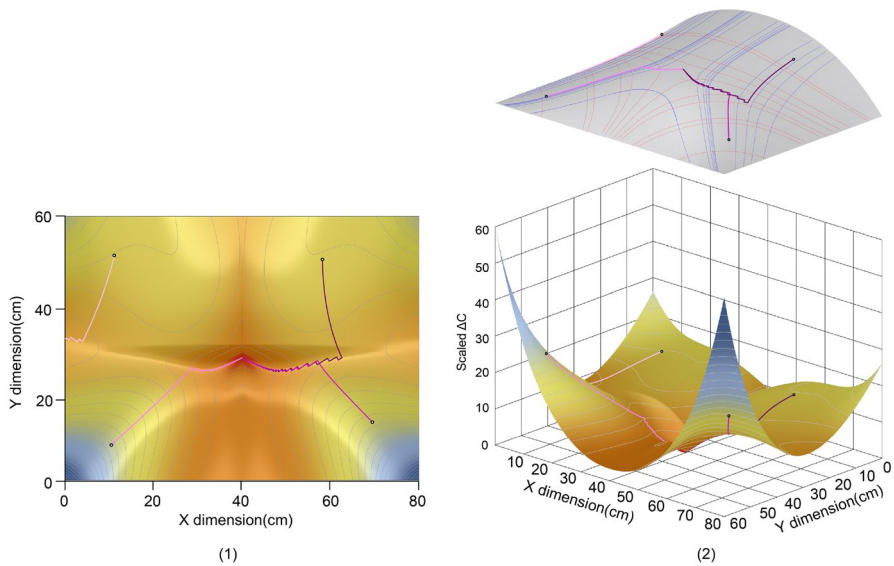


Fig. 6 Movement trajectory of umbilic agents

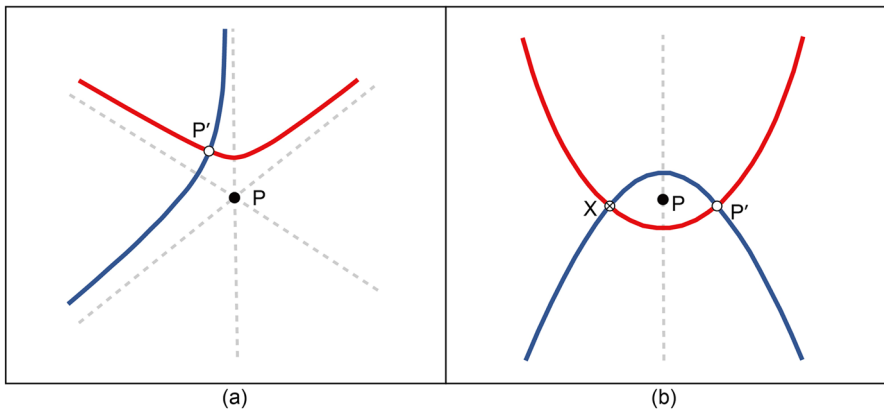


Fig. 7 Assuming p is an absolute umbilic point, and p' is a found umbilic point: (a) If p is of the trisector type, the curvature lines passing through p' will have only one intersection point; (b) If p is of the wedge type, the curvature lines passing through p' will have two intersection points

Strip Agent System

Preparation of the Environment

As mentioned above, the PC lines that pass through the umbilical points divide the surface into a series of regions in which there are only orthogonal PC line networks. The strip agent system adjusts the layout of the PC lines within these regions.

Here, it is only necessary to divide the surface using the skeleton derived from the trisector points. Even if there are wedge points within the divided regions, they can be addressed through the design of strip agent behavior, which will be discussed later.

Ideally, the straight PC lines passing through a trisector point would appear as three branches starting from the trisector (Fig. 8-1). However, because the umbilic agent system only produces approximate points, the PC line drawing algorithm can only draw one continuous line, that is to say, two of the multiple branches at a time (Fig. 8-2). Thus, it is necessary to draw the third branch separately. The strategy here is to move the umbilic point within a very small distance until a third PC line can be drawn starting from it towards the third direction. Then the third branch can be obtained by pulling the starting point of the third line back to the umbilic point (Fig. 8).

For the case surface in this section, three curves in each of the maximum and minimum principal directions are drawn with the above method, forming the topological skeleton of the surface. The layout of PC lines in two directions can be considered and adjusted separately (Fig. 9). Taking the maximum PC direction as an example, three curves in the maximum direction divide the surface into three regions (Fig. 9-1a), while the three curves in the minimum direction (Fig. 9-2a) serve as the seed curve for strip agent (Fig. 9b). Figure 9-1c shows the results of using the maximum curve as the seed curve for the strip agent, whereas Fig. 9-2c shows the results from the minimum curve. Only one seed curve is needed in each region. In fact, the seed curve doesn't have to be a PC curve; Any curve on the surface that can sequentially string together PC lines in opposite direction can serve as the seed curve. Another thing to note is that the endpoints of the seed curve need to be slightly offset from the region's boundary, as it can be challenging to generate a continuous PC line right at the edges.

Agent Definition

The strip agents are initialized with a series of seed points on the seed curve. Each agent only moves in the parameter domain of its seed curve and has a position attribute ' t_a ', which represents its parameter value within the curve's parameter domain. Strip agents have a Boolean attribute 'isFixed'. 'isFixed' is set to true for agents located at the endpoints of the seed curve to keep their positions fixed throughout the simulation. Agents that share the same seed curve are 'neighbors'. 'DirectNeighbors' are points that are directly adjacent to each other.

A PC line runs through the seed point of each strip agent. These PC lines divide the regions into a series of PSs. An agent and its two direct neighbors together define two PSs. The differences between the target features of its two PSs determine the agent's behaviors in the parameter domain of its seed curve. As with umbilic agents, the Boolean attribute 'isFinished' can be set to true to remove them from the agent simulation. Once the sum of all agents' movement is less than a preset tolerance, the simulation has converged and ends. In Grasshopper, designers can freely set the initial positions for the strip agents. This can be achieved either through the uniform subdivision of a seed

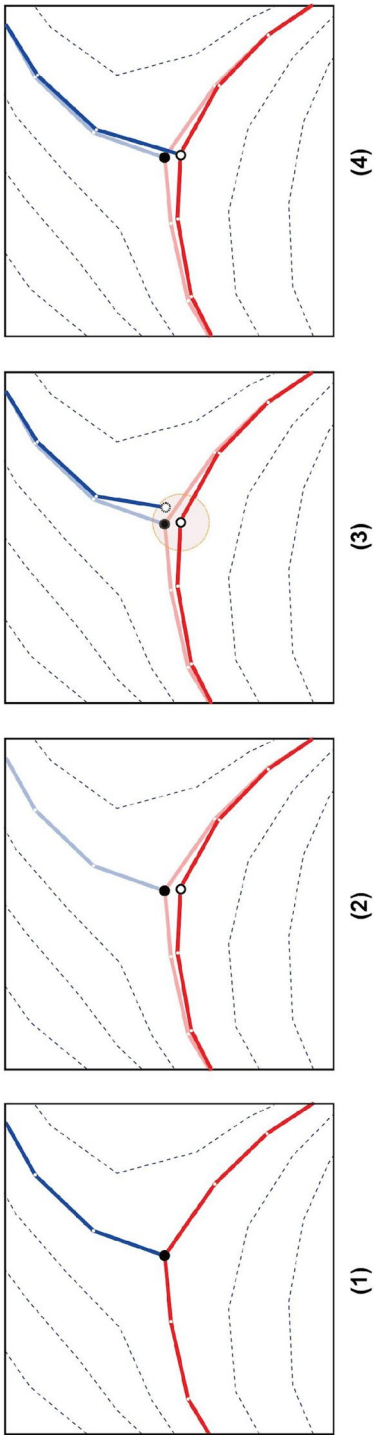


Fig. 8 Strategies for drawing PC lines passing through trisector points (1) the PC lines that pass through the actual umbilic; (2) the PC lines that pass through the found umbilic; (3) move the umbilic point within a small distance until a third PC line can be drawn; (4) merge the auxiliary points to the found points

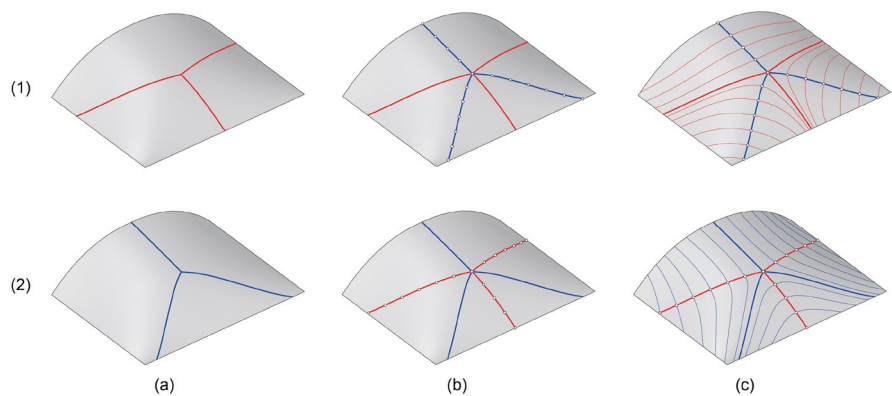


Fig. 9 Preparation of the environment for strip agents in both (1) maximum and (2) minimum directions

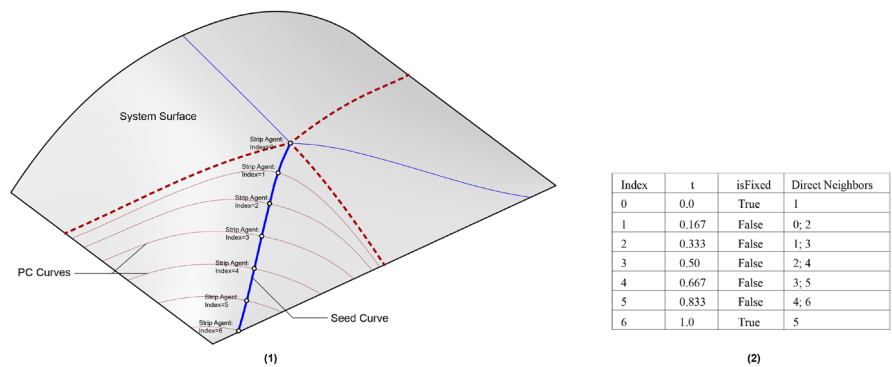


Fig. 10 The setup of the strip agent

curve (as shown in Fig. 10) or by manually/randomly selecting points along the seed curve. Indeed, at this stage, PC line networks with varied densities can be generated by evenly dividing the seed curve with different spacings (Fig. 11). It worths noting that evenly spaced agents do not produce evenly spaced PC lines. Further optimization of agent distribution is required to generate evenly spaced PC lines.

For complex free-form surfaces on which wedge points exist among the regions, there will be a situation where the PC line that passes through an agent has additional intersection points with the seed curve other than the one at t_a . In this case, the movement of this agent is jointly determined by the two sets of PSs adjacent to the two intersection points (Fig. 12). To simplify the simulation process, a compromise strategy is to artificially select a curvature line near the wedge point and treat it as a boundary. The seed curve ceased at the assumed boundary to avoid interference caused by additional intersections.

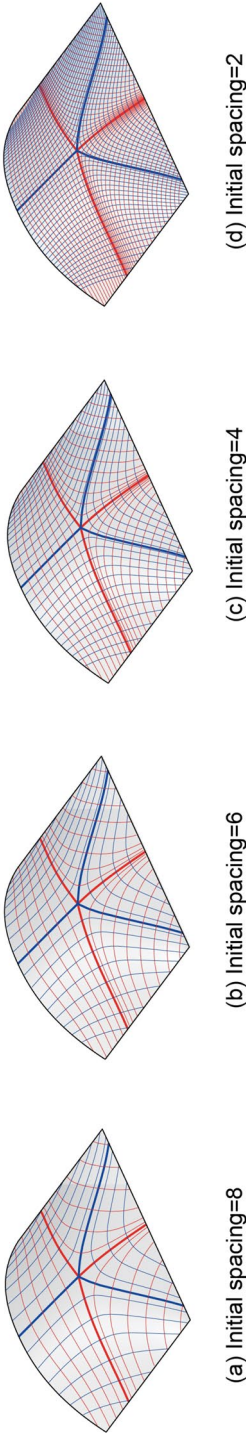


Fig. 11 PC line networks with varied densities can be generated by evenly dividing the seed curve with different spacings

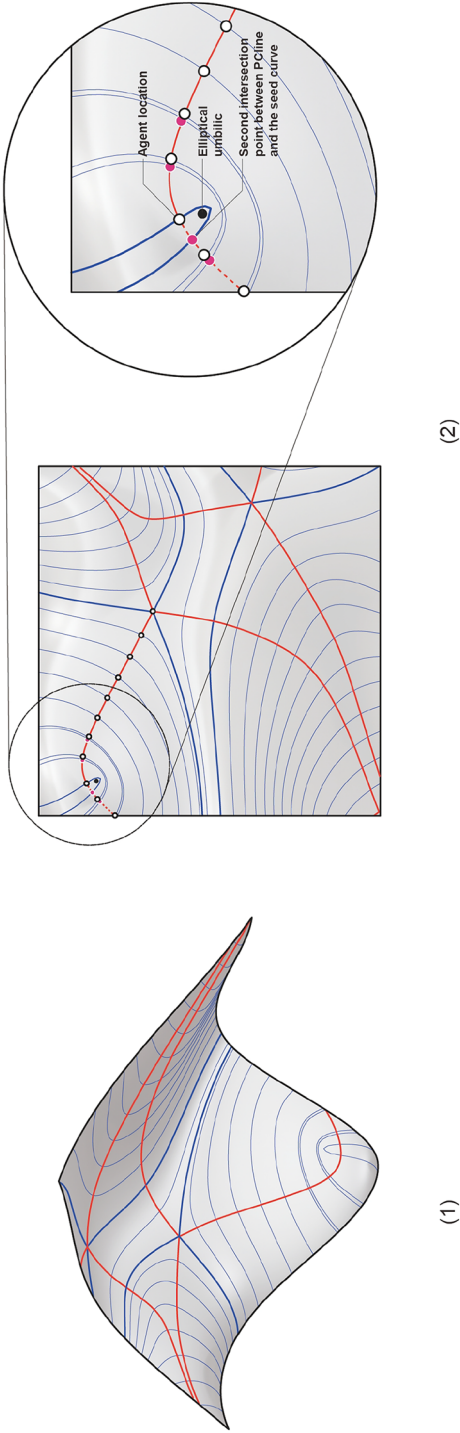


Fig. 12 The situation where wedge points occur

Agent Behavior

The behaviors of the strip agent are closely related to a preset goal. The agent system allows the designer to define the goal of the PC line layout for different considerations. In this section, we will take the goal of keeping the average spacing between the PC lines consistent as an example. That would keep the average width of each PS the same. Each agent achieves this goal through the combined action of three behaviors:

- Spacing behavior: average the spacing of PC lines directly by moving the agents. The average widths of its two adjacent PSs are first calculated through sampling on strips (Fig. 13a). The difference between the average width is ΔD . To average the spacing, the agent will move in a direction that decreases ΔD with a step size Δt (Fig. 13b). Supposing that the agent's motion in the parameter domain of its seed curve is T_m , T_m is constant proportional (k) to the product of Δt and ΔD :

$$T_m = k\Delta t\Delta D \quad (6)$$

- Add-agent behavior: When the width of a PS is greater than a set threshold (for example, twice the target width), a new agent is added in the middle (in the parameter domain of the seed curve) of the two adjacent agents of the PS. This behavior avoids excessive average widths (that, for example, exceed the limit of the available sheet material.)
- Remove-agent behavior: When executing the behavior of agent A, if the distance between A and one of its direct neighbor agents B is less than the threshold (such as half of the target average spacing), agent B will be removed in the next iteration. This remove-agent behavior could avoid an overly dense layout of PC lines. Together with the add-agent behavior, it could keep the average width of the PS within a desirable range.

Implementation: PC Lines Layout with Different Goals

ABM allows different aspects that influence the layout of the curvature lines to be integrated, such as geometric features, structural performance, material properties, and fabrication constraints.

In this study, two goals are selected to test the agent-based simulation. The first goal is to keep the width of PSs consistent, as described above. The second goal is to make the width of the PSs smaller where the curvature of the surface is larger, while larger where the curvature is smaller. This is achieved by adding the local curvature factor of the surface into the width adjustment process. In the case of uniform curvature, the curvature difference of two PC lines is positively related to their spacing. Thus, taking the spacing weighted by the curvature difference as the goal parameter could relate the width of the strip to local curvature. The agent moves iteratively to equalize the parameter value. The add and remove behaviors are also used to keep the parameter value within a certain range.

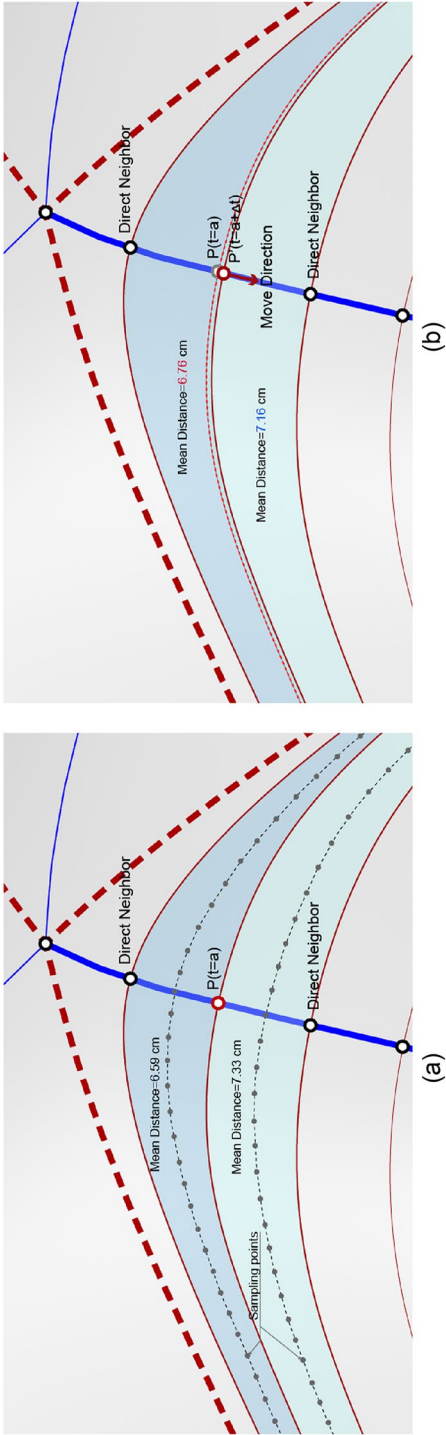


Fig. 13 Spacing behavior of strip agents

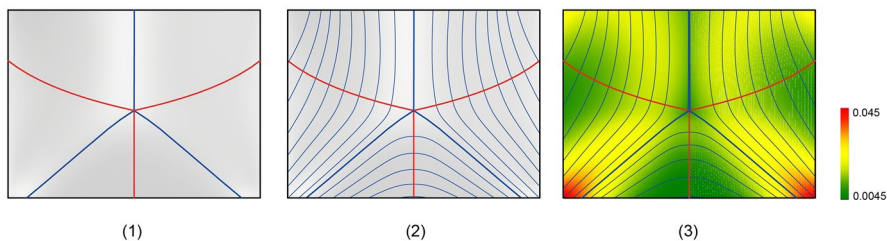


Fig. 14 Agent-based method implemented on the case surface with one: (1) surface skeleton; (2) layout with mean strip width; (3) curvature-weighted layout

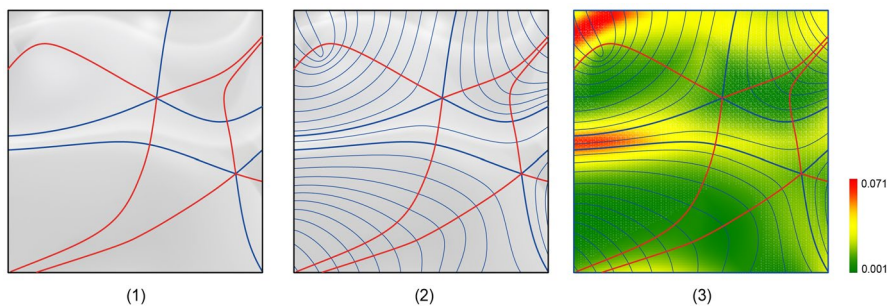


Fig. 15 Agent-based method implemented on two freeform case surfaces: (1) surface skeleton layout; (2) layout with mean strip width; (3) curvature-weighted layout

The two case surfaces—one is the case surface we use to showcase the agent systems, another is the freeform surface with two trisector points and one wedge, as shown in Fig. 12—are first used to demonstrate the effect of the implementation. As can be seen from the figures, the results of the average spacing appear more balanced, whereas curvature-weighted layout results align better with the curvature distribution. (Figs. 14, 15).

More generally, the proposed method is tested on a larger variety of surfaces. Figure 16a shows a freeform surface with one umbilic point. Most areas of this surface present very low curvatures while the top middle part show high ones. There is no need for topological skeleton drawing for this kind of surfaces without umbilics. The seed curve can be chosen from the PC lines in an opposite direction. In Fig. 16b, a surface with compound umbilics is tested. A trisector and four wedges have been found on the surface, and their interactions have led to the formation of a complex topological skeleton. Closed PC lines have appeared on the surface, including two branches originating from the trisector. The results of both evenly spaced and curvature-weighted layouts indicate that the method proposed in this study can achieve favorable outcomes in both cases. It should be noted that surfaces with planar regions are not considered in this research because planar regions don't have curvature, which would pose certain challenges to the streamline integration algorithms.

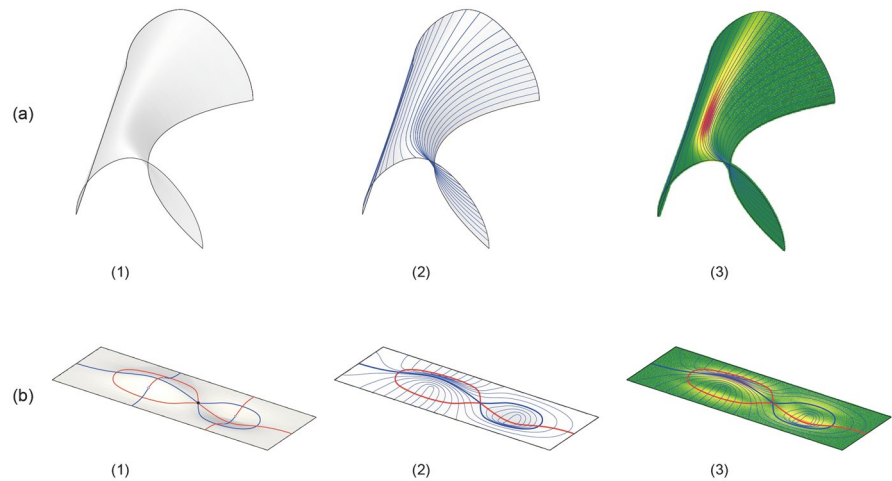


Fig. 16 Agent-based method implemented on two freeform case surfaces: (1) surface skeleton layout; (2) PS layout with mean strip width; (3) curvature-weighted layout

Discussion

To facilitate the construction of the surface, it is a common idea to optimize each PS into a developable surface and then unfold it (Figs. 17, 18). Flattened PSs can be cut directly from sheet materials such as plywood panels, aluminum boards, etc. Assembling the PSs together along the shared edges will automatically form the designed freeform surface. The assembly process is informed by the strip geometry with no molds required. Since PSs-surface is a ‘semi-discrete surface,’ which can be viewed as a limit condition of quad meshes (Pottmann et al. 2008), the method

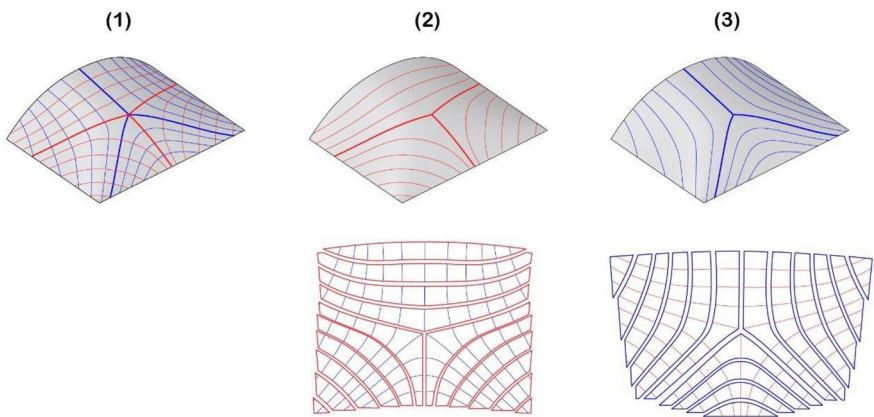


Fig. 17 Unfolding of the case surface: (1) surface with both sets of PC lines; (2) maximum PSs; (3) minimum PSs

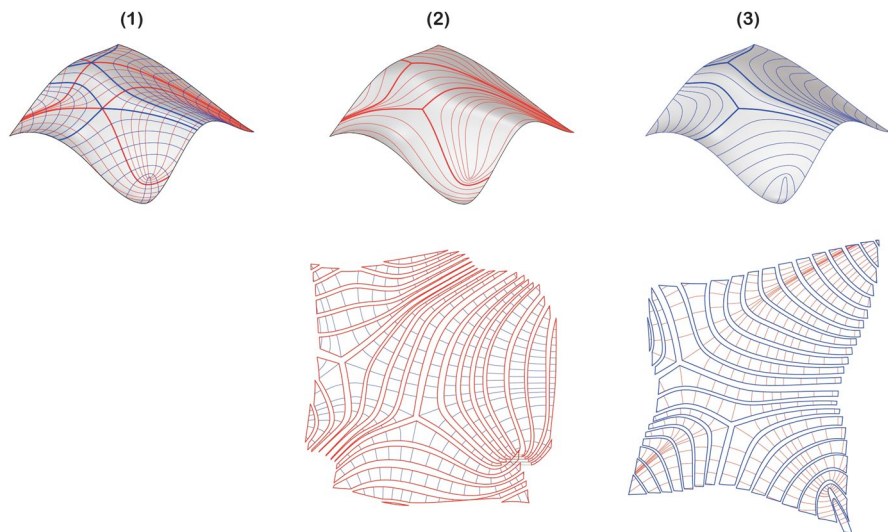


Fig. 18 Unfolding of the case surface with two trisector points and one wedge: (1) surface with both sets of PC lines; (2) maximum PSs; (3) minimum PSs

proposed can also facilitate the partitioning of freeform surfaces into panels using PC lines in both directions. The advantages of penalization using principal curvature lines have been validated in projects such as the Eiffel Tower Pavilions (Schiftner et al. 2013). It is worth noting that the mesh generated by the intersection of PC lines is not a strictly quad mesh, especially when considering the presence of triangular faces along the edges. While PC line network can serve as a guide for quad meshing, creating an effective quad meshing algorithm requires thoughtful incorporation of factors like quad size, boundary conditions, and adaptive subdivision (Chen et al. 2019).

Compared to fabricating surfaces based on other types of curves, such as orthogonal curve networks that do not consider surface features, and geodesic curves that describe the shortest path between two points on a surface, PSs can better adapt to the shape of surfaces as they inherently describe local curvature properties of surfaces. The fabricated results can exhibit better surface continuity and also hold potential advantages for constructing developable curved support structures. Despite this, geodesic curves offer advantages in the construction of gridshell structures using straight planks (Oberbichler et al. 2023).

Compared with previous studies (Pottmann et al. 2008; Joo et al. 2014; Takezawa et al. 2016), the approach proposed in this study is more suitable for architects for three main reasons: first, by considering umbilics and their corresponding topological skeletons, architects can obtain clear and intuitive understanding of the surface; second, with ABM, architects are allowed to incorporate diverse design considerations in the generation process; third, the method follows a geometric approach that aligns better with the way architects think. It is carried out in Rhino3D and Grasshopper, which is widely utilized by architects for parametric design

exploration. In this way PS-based surface reconstruction can be more conveniently integrated into the exploration of various freeform architectural geometries including shells, skins, and roofs, endowing buildings with unique aesthetic characteristics (Fig. 19).

The main limitations of the proposed method in this study can be summarized as follows: first, this approach is not an automated algorithm, rather, it is a process that requires manual intervention and behavior definition by architects. Consequently, architects need to have a fundamental understanding of both surface characteristics and ABM method. Second, ABM itself, being based on iterative algorithms, often demands a longer computation time. The computation time could be influenced by factors such as the number of agents, step size and termination conditions of the model. Taking the simple case surface as an example (Fig. 1b), let the dimensions of the surface boundary be 60 cm × 80 cm. The algorithm is executed on a PC with a 13th Gen Intel(R) Core(TM) i9-13900H 2.60 GHz CPU and a 64-bit Windows 11 operating system. For the Umbilic Agent system with an initial step size of 0.5 cm, the model can converge in approximately 29 s, reaching the umbilic point (where the absolute difference between the two directions of PC is less than 0.000005 cm). For Strip Agents, let all the PC lines drawn with a step size of 0.5 cm, agents are initialized with a spacing of 6 cm, initial step size of the agents is set as 0.05 cm, the spacing calculation involves 15 sampling points. If convergence is considered achieved when the movement amplitude of all agents is less than 0.005 cm, it would take approximately 48 s to converge. Adjusting the stop threshold to 0.001 cm would increase the required time to around 4 min and 35 s.

It is worth noting that this study can also be adapted to other types of curves, like principal stress lines. Principal stress lines, which are widely used in the field of structural design, have similar properties to PC lines. Therefore, the layout of principal stress lines can also be optimized with a similar method as has been demonstrated on PC lines, albeit with different goals. A simple slab, with uniformly distributed loads and supported by four corners, is tested for the even distribution of stress lines. Grasshopper plugin Karamba (Preisinger 2013) is employed for structural analysis. The algorithm for drawing PC lines is rewritten to draw principal stress lines. The algorithm is mainly necessary to adapt to mesh geometry, and the associated principal stress values and directions obtained from structural simulation. Two trisector points could be identified on the slab. After the slab is divided by

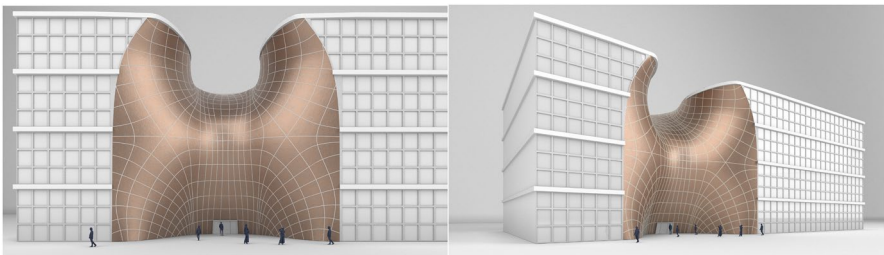


Fig. 19 The application of the PS-based surface subdivision method in a freeform façade

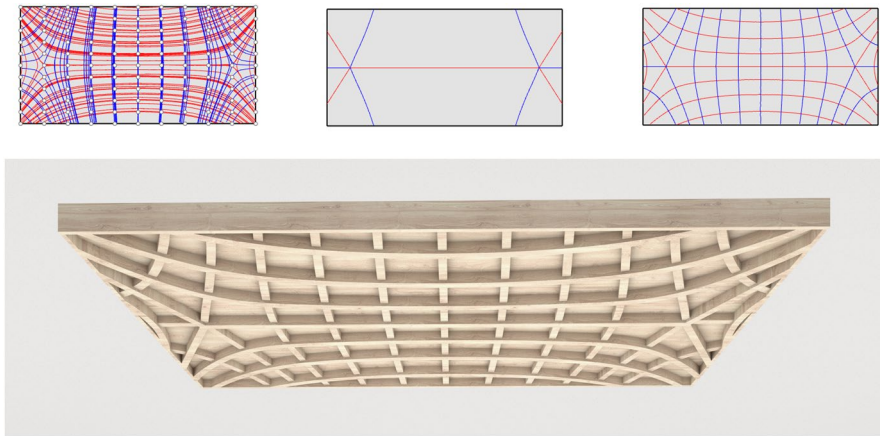


Fig. 20 Extended application in the design of principal stress lines

stress lines passing through the trisectors, the adjusted ABM algorithm is used to optimize the layout of stress lines. The results show that the method also works well for this case (Fig. 20). The application of this method to more complex surface structures and boundary conditions requires special research.

Conclusion

This study presents an ABM method for principal stress (PS) modeling tailored for freeform surfaces in architectural design. By incorporating umbilical points within the agent system, the method can accommodate surfaces with varying umbilical conditions. The ABM algorithm further enables the customization of curvature lines to align with user-defined objectives.

This study embodies a construction-oriented methodology aimed at aiding architects in the formal exploration of freeform buildings and structural systems during the early design phase. The application of Principal Stress (PS) in architectural contexts such as freeform facades and roofs offers improved surface approximations compared to other common approaches. Furthermore, the research method exhibits the potential for extension to other curve families, such as principal stress lines. Future plans involve implementing this method in the practical construction of double-curved shell structures.

Acknowledgements This research was funded by National Key R&D Program of China (2022YFC3802203), National Postdoctoral Program for Innovative Talents of China (BX20220228), China Postdoctoral Science Foundation (2022M722399), the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100) and the Fundamental Research Funds for the Central Universities.

Author Contributions Hua Chai: Conceptualization, Methodology, Software, Formal Analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization. Luis Orozco: Conceptualization, Methodology, Software, Writing - Review & Editing. Fabian Kannenberg:

Conceptualization, Methodology, Software, Writing - Review & Editing. Lasath Siriwardena: Software, Formal Analysis, Investigation, Data Curation. Tobias Schwinn: Conceptualization, Methodology, Software, Validation, Resources, Data Curation, Writing - Review & Editing, Supervision, Project administration. Hanning Liu: Visualization. Achim Menges: Conceptualization, Supervision, Project administration, Funding acquisition. Philip F. Yuan: Conceptualization, Supervision, Project administration, Funding acquisition.

Funding This research was funded by the German Research Foundation DFG under Germany's Excellence Strategy—EXC 2120/1—390831618.

Data availability Data available on request from the authors.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Berry M. V., and J.H. Hannay. (1977). Umbilic points on Gaussian random surfaces. *Journal of Physics A: Mathematical and General* 10(11): 1809 <https://doi.org/10.1088/0305-4470/10/11/009>
- Bommes D., H. Zimmer, and L. Kobbelt. (2009). Mixed-integer quadrangulation. *ACM Transactions On Graphics* 28(3): 1-10. <https://doi.org/10.1145/1531326.1531383>
- Carpo M. (ed.). (2012). *The digital turn in architecture 1992–2012*. John Wiley & Sons. ISBN:9781119951759
- Chai H., H.J. Wagner, Z. Guo, Y. Qi, A. Menges, and P.F. Yuan. (2022). Computational design and on-site mobile robotic construction of an adaptive reinforcement beam network for cross-laminated timber slab panels. *Automation in Construction* 142: 104536. <https://doi.org/10.1016/j.autcon.2022.104536>
- Chen, W., X. Zheng, J. Ke, N. Lei, Z. Luo, and X. Gu. (2019). Quadrilateral mesh generation I: Metric based method. *Computer Methods in Applied Mechanics and Engineering* 356: 652-668. <https://doi.org/10.1016/j.cma.2019.07.023>
- Delmarcelle, T., and L. Hesselink. (1994). The topology of symmetric, second-order tensor fields. *Proceedings Visualization '94*: 140–147. <https://doi.org/10.1109/VISUAL.1994.346326>.
- Delmarcelle, T. (1995). *The visualization of second-order tensor fields*. Stanford University.
- Eigensatz, M., M. Kilian, A. Schiftner, N.J. Mitra, H. Pottmann, and M. Pauly. (2010). Paneling architectural freeform surfaces. *ACM SIGGRAPH 2010 papers*. <https://doi.org/10.1145/1833349.1778782>
- Evolute GmbH (2021). EvoluteTools T.MAP. *Food4Rhino*. URL: <https://www.food4rhino.com/en/app/evolutetools-tmap>
- Fischer-Cripps, A. C. (2007). *Introduction to contact mechanics*. Springer New York, NY.
- Gavriil, K., A. Schiftner, and H. Pottmann. (2019). Optimizing B-spline surfaces for developability and paneling architectural freeform surfaces. *Computer-Aided Design* 111: 29-43. <https://doi.org/10.1016/j.cad.2019.01.006>
- Gerber, D. J., E. Pantazis, and A. Wang. (2017). A multi-agent approach for performance based architecture: design exploring geometry, user, and environmental agencies in façades. *Automation in construction* 76: 45-58. <https://doi.org/10.1016/j.autcon.2017.01.001>
- Greco L. (2019). Dodo, version v.03. *Food4Rhino*. URL: <https://www.food4rhino.com/en/app/dodo>

- Groenewolt, A., T. Schwinn, L. Nguyen, and A. Menges. (2018). An interactive agent-based framework for materialization-informed architectural design. *Swarm Intelligence* 12:155-186. <https://doi.org/10.1007/s11721-017-0151-8>
- Joo, H. K., T. Yazaki, M. Takezawa, and T. Maekawa. (2014). Differential geometry properties of lines of curvature of parametric surfaces and their visualization. *Graphical models* 76(4): 224-238. <https://doi.org/10.1016/j.gmod.2014.05.001>
- Kahlert, J., M. Olson, and H. Zhang. (2011). Width-bounded geodesic strips for surface tiling. *The Visual Computer* 27: 45-56. <https://doi.org/10.1007/s00371-010-0513-3>
- Kälberer, F., M. Nieser, and K. Polthier. (2007). Quadcover-surface parameterization using branched coverings. *Computer graphics forum* 26(3): 375-384. Oxford, UK: Blackwell Publishing Ltd. <https://doi.org/10.1111/j.1467-8659.2007.01060.x>
- Kirschman, C. F., and C. C. Jara-Almonte. (1992). A parallel slicing algorithm for solid freeform fabrication processes. *1992 International Solid Freeform Fabrication Symposium*: 26-33. <https://doi.org/10.15781/T2NS0MF5K>
- Lienhard, J. (2014). *Bending-active structures: form-finding strategies using elastic deformation in static and kinetic systems and the structural potentials therein*. Universität Stuttgart. <https://doi.org/10.18419/opus-107>
- Maekawa, T., F.E. Wolter, and N.M. Patrikalakis. (1996). Umbilics and lines of curvature for shape interrogation. *Computer Aided Geometric Design* 13(2): 133-161. [https://doi.org/10.1016/0167-8396\(95\)00018-6](https://doi.org/10.1016/0167-8396(95)00018-6)
- Menges, A., F. Kannenberg, and C. Zechmeister. (2022). Computational co-design of fibrous architecture. *Architectural Intelligence* 1. <https://doi.org/10.1007/s44223-022-00004-x>
- McNeel, R. et al. (2023a). Rhinoceros 3D, Version 7. *Robert McNeel & Associates*. URL: www.rhino3d.com.
- McNeel, R. et al. (2023b). Grasshopper in Rhino 7, Version 1.0. *Robert McNeel & Associates*. URL: www.grasshopper3d.com.
- Nguyen, L., T. Schwinn, A. Groenewolt, M. Maierhofer, M. Zorn, D. Stieler, L. Siriwardena, F. Kannenberg, and A. Menges, (2022). ABxM.Core: The Core Libraries of the ABxM Framework, *DaRUS*, V1. <https://doi.org/10.18419/darus-2994>
- Oberbichler, T., E. Schling, and K.U. Bletzinger. (2023). Tracing curvature paths on trimmed multipatch surfaces. *Applied Mathematical Modelling* 118: 253-271. <https://doi.org/10.1016/j.apm.2023.01.033>
- Oberbichler T. (2021). Bowerbird, Version 2.8.0. *Food4Rhino*. URL: <https://www.food4rhino.com/en/app/bowerbird>
- Patrikalakis, N. M., and T. Maekawa. (2002). Shape interrogation for computer aided design and manufacturing. Heidelberg: Springer. <https://doi.org/10.1007/978-3-642-04074-0>
- Pottmann, H., A. Aspert, M. Hofer, and A. Kilian. (2007). *Architectural geometry*. Bentley Institute Press. ISBN: 978-1-934493-04-5
- Pottmann, H., A. Schiftner, P. Bo, H. Schmiedhofer, W. Wang, N. Baldassini, and J. Wallner. (2008). Freeform surfaces from single curved panels. *ACM Transactions on Graphics* 27(3): 1-10. <https://doi.org/10.1145/1360612.1360675>
- Pottmann, H. (2013). Architectural geometry and fabrication-aware design. *Nexus Network Journal* 15:195-208. <https://doi.org/10.1007/s00004-013-0149-5>
- Preisinger, C. (2013). Linking Structure and Parametric Geometry. *Architectural Design* 83: 110-113. <https://doi.org/10.1002/ad.1564>
- Schiftner, A., N. Leduc, P. Bompas, N. Baldassini, and M. Eigensatz. (2013). Architectural geometry from research to practice: the Eiffel Tower Pavilions. *Advances in Architectural Geometry 2012*: 213-228. Springer, Vienna. https://doi.org/10.1007/978-3-7091-1251-9_17
- Schling, E., and Z. Wan. (2022). A geometry-based design approach and structural behaviour for an asymptotic curtain wall system. *Journal of Building Engineering* 52: 104432. <https://doi.org/10.1016/j.jobbe.2022.104432>
- Schwinn, T., O.D. Krieg, and A. Menges. (2014). Behavioral strategies: synthesizing design computation and robotic fabrication of lightweight timber plate structures. *ACADIA 2014*:177-188. <https://doi.org/10.52842/conf.acadia.2014.177>
- Snooks, R. (2022). Behavioral tectonics: agentBody prototypes and the compression of tectonics. *Architectural Intelligence* 1. <https://doi.org/10.1007/s44223-022-00007-8>

- Stieler, D., T. Schwinn, S. Leder, M. Maierhofer, F. Kannenberg, and A. Menges. (2022). Agent-based modeling and simulation in architecture. *Automation in Construction* 141: 104426. <https://doi.org/10.1016/j.autcon.2022.104426>
- Takezawa, M., T. Imai, K. Shida, and T. Maekawa. (2016). Fabrication of freeform objects by principal strips. *ACM Transactions on Graphics* 35(6): 1-12. <https://doi.org/10.1145/2980179.2982406>
- Takezawa, M., Y. Otoguro, K. Matsuo, T. Shibutani, A. Sakurai, and T. Maekawa. (2021). Fabrication of doubly-curved CFRP shell structures with control over fiber directions. *Computer-Aided Design* 136: 103028. <https://doi.org/10.1016/j.cad.2021.103028>
- Tang, C., M. Kilian, P. Bo, J. Wallner, and H. Pottmann. (2016). Analysis and design of curved support structures. *Advances in architectural geometry 2016*: 8-23. Hochschulverlag Zürich. ISBN: 978-3-7281-3778-4
- Vestartas P. (2018). Mesh Curvature, Version 1.2. *Food4Rhino*. URL: <https://www.food4rhino.com/en/app/mesh-curvature>

Hua Chai is a Postdoctoral researcher at the College of Architecture and Urban Planning (CAUP) of Tongji University. His research focuses on computational design and robotic production technologies for innovative timber building systems. He received his Ph.D. from Tongji University in 2022. In 2020, he spent a year as a guest researcher at the Institute for Computational Design and Construction (ICD), the University of Stuttgart. Hua Chai was honored with the fellowship of the China National Postdoctoral Program for Innovative Talents in 2022.

Luis Orozco is a Research Associate and PhD Candidate at the Institute for Computational Design and Construction (ICD) at the University of Stuttgart, Germany. He is also a registered architect in the state of New York, USA. He is a member of the EXC 2120: Cluster of Excellence IntCDC—Integrative Computational Design and Construction for Architecture where his research focuses on a novel, digital, modular wood building system to enable the adoption of prefabricated timber construction in additional building typologies.

Fabian Kannenberg is a Research Associate and Doctoral Candidate at the Institute for Computational Design and Construction (ICD) at the University of Stuttgart, where he also received his Master of Science in Architecture from the Integrative Technologies and Architectural Design Research (ITECH) program at the University of Stuttgart. His current research at ICD is affiliated with the Cluster of Excellence Integrative Computational Design and Construction for Architecture (IntCDC). Within this framework, he develops computational design methods for high-performance fiber composite building systems.

Lasath Siriwardena is an Australian Computational Designer working as a Research Associate at the Institute for Computational Design and Construction (ICD) at the University of Stuttgart. He is a member of the 'EXC 2120: Cluster of Excellence IntCDC—Integrative Computational Design and Construction for Architecture', where his research focuses on Agent-Based Modelling of Segmented Shell Timber Structures.

Tobias Schwinn is a Lecturer (Akademischer Rat) at the Institute for Computational Design and Construction (ICD) at the University of Stuttgart, Germany. In research and teaching, he is focusing on behavior-based approaches for integrating robotic fabrication and computational design processes in the context of segmented shell structures. He is one of the founding members and currently leader of ICD's "Agent-based Modeling and Simulation" research group, where he supports colleagues in their diverse research areas, such as design of adaptive buildings, fabrication-oriented design of modular concrete structures, and distributed robotics. As Head of Research and Research Infrastructure, he is in charge of the robotic fabrication equipment at the institute's Robolab and the Computational Construction Laboratory in Wangen.

Hanning Liu is a master student at the College of Architecture and Urban Planning (CAUP) of Tongji

University. His research focuses on AR-aided architecture design and robotic production.

Achim Menges is a registered architect in Frankfurt and Full Professor at the University of Stuttgart, where he is the founding director of the Institute for Computational Design and Construction (ICD) and the director of the Cluster of Excellence Integrative Computational Design and Construction for Architecture (IntCDC). The focus of Achim Menges' practice and research is on the development of integrative design at the intersection of computational design methods, robotic manufacturing and construction, as well as advanced material and building systems. His work is based on an interdisciplinary approach in collaboration with structural engineering, system engineering, production engineering, computer science, material science, natural science as well as social science and the humanities.

Philip F. Yuan is a Professor and Associate Dean of the College of Architecture and Urban Planning (CAUP) at Tongji University, Honorary Fellow of American Institute of Architecture (Hon. FAIA). He is the co-founder of DigitalFUTURES Association, a global educational initiative with a particular emphasis on the latest computational design and fabrication technologies in architecture. He serves as the Editor-in-Chief of the Architectural Intelligence journal. Yuan is a pioneer in architectural intelligence and robotic fabrication, making achievements in both academic research and architectural practice. He emphasizes the integration of technological innovation and architectural culture, building environment and ethics, constantly seeking human-machine collaboration in the era of post-humanism.

Authors and Affiliations

Hua Chai¹ · Luis Orozco^{2,3} · Fabian Kannenberg^{2,3} · Lasath Siriwardena^{2,3} · Tobias Schwinn^{2,3} · Hanning Liu¹ · Achim Menges^{2,3} · Philip F. Yuan¹

✉ Achim Menges
achim.menges@icd.uni-stuttgart.de

✉ Philip F. Yuan
philipyuan007@tongji.edu.cn

Hua Chai
chaihua@tongji.edu.cn

Luis Orozco
luis.orozco@icd.uni-stuttgart.de

Fabian Kannenberg
fabian.kannenberg@icd.uni-stuttgart.de

Lasath Siriwardena
lasath.siriwardena@icd.uni-stuttgart.de

Tobias Schwinn
tobias.schwinn@icd.uni-stuttgart.de

Hanning Liu
liuhanning@tongji.edu.cn

¹ College of Architecture and Urban Planning, Tongji University, Shanghai, China

² Institute for Computational Design and Construction, University of Stuttgart, Stuttgart, Germany

³ DFG Cluster of Excellence 2120-Integrative Computational Design and Construction for Architecture, University of Stuttgart, Stuttgart, Germany