Proceedings of the IASS 2024 Symposium
Redefining the Art of Structural Design
August 26-30, 2024, Zurich Switzerland
Philippe Block, Giulia Boller, Catherine DeWolf,
Jacqueline Pauli, Walter Kaufmann (eds.)

Self-Aware Spatial Pattern Model: beyond shape, towards adaptive form

Mohammad Hassan SALEH TABARI*, Günther H. FILZ

*Universität Innsbruck, Faculty of Architecture, LSU Lightweight Structures Unit
Technikerstrasse 21c, 6020 Innsbruck, Austria
mohammad.saleh-tabari@uibk.ac.at

Abstract

In our ongoing research, inspired by the nest-building strategies of weaverbirds, we have explored the potential of utilizing variations in pattern topology for the design of bending-active surface structures. While the results show success in addressing multiple objectives such as architectural and geometrical aspects during the design process, they also reveal challenges. One of these is the question of how to integrate the considered target evaluation, such as structural performance, directly into the design process without compromising its freedom. Studies also reveal that weaverbirds intuitively optimize structural patterns by learning from past experiences. In this context, inspired by weaverbirds' ability to learn and adapt, reinforcement learning (RL) is proposed to equip the geometrical design with evaluation and self-learning capabilities in finding bending-active strips on a surface, focusing on geometry and structural performance. The paper presents a geometrical system controllable with numerical models for pattern generation on surfaces, setting the stage for the RL model's heuristic interaction with the design environment in the design of the complex bending-active structures.

Keywords: Pattern topology, Bending-active Structures, Structural performance, Reinforcement learning, Decision-making.

1 Introduction to the design of patterns for spatial, bending-active structures:

Bending-active structures utilize the ability of material systems to take on three-dimensional curved shapes from initially straight elements [1]. The use of developable flat structural elements enhances the ease of fabrication and assembly of these types of structures. The concept of active bending as a structural principle is not new. Primitive nomadic cultures have used elastic bending to construct small dwellings from lightweight, flexible materials, like the Madan Mudhif and the Mongolian Yurt [2]. surface structures are made from flat, unrollable elements that are bent and joined together to form a curved surface structure [2]. The resulting surface curvature is depending on the pattern by which the structural elements are connected [3]. The geometry of the bending-active structures has direct relation with the physical properties of the elements. For example, the elements cross-section or elastisity module may not allow the designer to have a structural element with a desired curvature [1]. Integrating material information, the prediction of geometrical transformation, and control of the bending behavior are key aspects in design bending-active surface structures [4]. Consequently, the design of these structures relies on notational and geometric representations, emphasizing that only a design that is accurately drawn and measured is possible to be realized. There are multiple design approaches and interests in the design of the bending-active structures by developable elements [4–7]. By setting up a geometry-based approach and numerical model, the system's geometry is used as a controlled means to generate curves on the surface that approximate the actual bent elements. Recent research offers a comprehensive overview of specific curve types, such as asymptotic and geodesic curves, suitable for generating structurally efficient, architectural structures with developable strips, an approach particularly interesting for the design of bending-active structures [5].

However, an increased variety in designs may be achieved, if early-stage parameters, including the curve generation, can be considered and flexibly changed during the process of design [8]. The result will be

Copyright © 2024 by Mohammad Hassan SALEH TABARI & Günther H. FILZ. Published by the International Association for Shell and Spatial Structures (IASS) with permission.

non-repetitive structures similar to the weaverbird's nest structures [9, 10], where weaver birds takes advantage of the non-repetition for adaptation of their nest to the environmental conditions [11] , such as response to varying climates or specific locations on the tree. Inspired by the nest-building strategies of weaverbirds, our ongoing research explores the potential of utilizing variations in pattern topology specifically for the design of elastic shells. Our previously presented method extracts surface curvature features to identify and apply suitable pattern topologies [8]. While the results show success in addressing integration of objectives such as architectural, and geometrical aspects during the form-finding process, they also reveal challenges. One of these is the question of how to integrate the considered target evaluation, such as structural performance, directly into the form-finding process without compromising the design freedom. In addition to aforementioned nest building strategies studies show weaverbirds' ability to learn from earlier experiences to intuitively find the best performing structural pattern [12]. From the experiences in nest building, weaverbirds learn how to reach a smoother structural pattern while maintaining the stability of the nest [13]. The same approach could be achieved in the design of bending-active surface structures with a geometrical system capable of learning from experiences.

In this context, we propose employing reinforcement learning (RL), rooted in behavioral science, to equip the geometrical design with evaluation and self-learning capabilities. Integrating geometrical design, especially when dealing with the complex geometry of bent and twisted elements, with a control system model such as reinforcement learning, is a challenging process. The reinforcement learning (RL) model has the task to accomplish two objectives: first, to assess the accuracy of the system's geometry relative to the actual bending geometry; and second, to evaluate the efficiency of the geometry as a structural component. We aim to develop a generative model with learning capabilities that support designers in decision-making for the creation of non-repetitive, adaptable bending-active structures. Based on the aforementioned challenges, we consequently consider a stepwise but hierarchical investigation that includes setting up a geometrical system controlled by the RL agent for finding geometry, evaluating the agent's perception of the environment and its decision-making processes, integrating structural analysis engines like finite element modeling with the geometric system, and training a generative model that learns to control geometric modeling with feedback from structural analysis results.In this paper, we focus on a geometrical system, controllable with numerical models, for finding a developable strip on a surface to create a pattern, where the output geometry results from a decision-making sequence. Expanding the geometrical modeling into separate states of decision-making facilitates later evaluation of overall results. By monitoring the steps involved in generating and evaluating the geometry, we can inform the RL model, enabling it to learn how to generate a model with desired characteristics like the consideration of structural performance.

To do so, we set up a geometrical design environment based on our mentioned combined numerical-geometrical approach for the generation of curves forming a structural pattern on a surface that is later suitable to be realized as a bending-active structure. We monitor the heuristic interaction of the RL model with the commissioned environment to design complex bent and twisted structures. Here, RL attempts to complete a pattern on a surface while considering the structural characteristics. The findings from this research bring us closer to our long-term goal, inching us towards creating a design methodology for elastic spatial structures that draw inspiration from weaverbirds' nests and self-learning principles.

2 Curves in differential geometry

In this chapter the fundamental geometrical theories applied in generating curves that represent bending-active elements will be presented. First, the definition of a parametric curve in three-dimensional space is explained. Second, the correlation between the curve and the surface is explained. Finally, the analysis of a ruled surface along a curve on a surface is explained.

2.2 Parametric curves in a three-dimensional space

Intuitively, we think of a curve as a path traced by a moving particle in space. This approach is formalized by considering a curve as a function of a parameter, say t. Thus, the domain of a curve is an interval (a, b) (possibly $(-\infty,\infty)$) consisting of all possible values of a parameter t [14].

$$\gamma(t) = \big(x(t), y(t), z(t)\big) \tag{1}$$

Vetor tangent is a vector that touches the curve at a single point and points in the direction of the curve's immediate path. With this eqaution we can calculate the vector tangent at the point where curve parameter t = t0 :

$$\acute{\gamma}(t) = \big(\acute{x}(t_0), \acute{y}(t_0), \acute{z}(t_0)\big) \tag{2}$$

So, the equation of the line tangent (direction) T to the curve at the point where t = t0 is:

$$T = (\ x :\ x(t_0) + \acute{x}(t_0)t, \qquad y :\ \ y(t_0) + \acute{y}(t_0)t \quad , \qquad z :\ \ z(t_0) + \acute{z}(t_0)t\ ) \tag{3}$$

The unit vector in the direction of $\acute{T}$ is called the principal normal vector and is denoted by N

$$N(s) = \frac{\acute{T}_{(S)}}{|\acute{T}_{(S)}|} \tag{4}$$

The plane determined by vectors T and N is called the osculating plane. If curve is planar, this plane is the plane in which curve lies.

Figure 1: Considering the curve's parameter domain interval between (0,1), the osculating planes at three points on the curve at [t = 0.25, t = 0.5, t = 0.75] are illustrated. The green arrow represents the normal (N), the red arrow represents the tangent (T), and the black arrow represents the binormal (B = T × N). The osculating circle with radius R lying on the osculating plane at parameter t defines the curve's curvature K at that parameter, equal to 1/R. The figure on the left shows the profile twist aligned with the curve's binormal vectors at parameter t, based on the osculating plane's rotation.

Torsion τ is a measure of how much a curve is twisting out of the plane of curvature (osculating plane).

$$\tau\ =\ -\ \frac{dB}{ds}\ .\hat{N} \tag{5}$$

Where N is the normal vector, B is the binormal vector, and $(\frac{dB}{ds})$ is the derivative of unit binormal vector with respect to arc length s. If a curve lies completely in a plane, its torsion is zero because it is not twisting out of that plane. This information gives us insight about the necessary material properties and structural stability for a design. This implies that understanding the shape and behavior of a curve in space can help determine how to build around or along it. For example, high rate of twist in the profile necessitates a material with a high rate of flexibility, which indicates a direct relationship between the geometric characteristics of a design (in this case, how much it twists) and the physical properties of the used materials. Beside this information, the analysis of a curve with respect to a known surface in three-dimensional space will provide additional information about the developable surface along the curve. These surfaces are easier to construct, meaning that complex three-dimensional structures can be built with less advanced assembly techniques. The following chapters will detail the geometrical analysis of a curve with respect to a surface.

2.2 Curves on a surface

Considering a point that is moving on a specific surface. The point has two coordinates of U and V that represents its location on a surface. We can evaluate the surface normal at the corresponding coordinate of the point on a surface. Having the tangent vector T and the corresponding normal of the surface at a point on the surface defines a Darboux frame [5, 15] consisting of the tangent vector t (the mapped vector T on the surface), n surface normal, and the u cross product of the t and n.

The Darboux frame is similar to the local coordinate of the curve in three-dimensional space. However, the Darboux frame represents the coordinate system of a curve in relation to a specific surface. The rotational movement of the Darboux frame along a curve on a reference surface is governed by a vector of angular velocity, called d [5].

$$d = \tau_g t - k_n u + k_g n = \begin{cases} \acute{t} = d * t = & k_g u + k_n n \\ \acute{u} = d * u = & -k_g t + \tau_g n \\ \acute{n} = d * n = & -k_n t + \tau_g u \end{cases} \quad (6)$$

Where the kg is the geodesic curvature, the Tg is the geodesic torsion, and the Kn is the normal curvature.

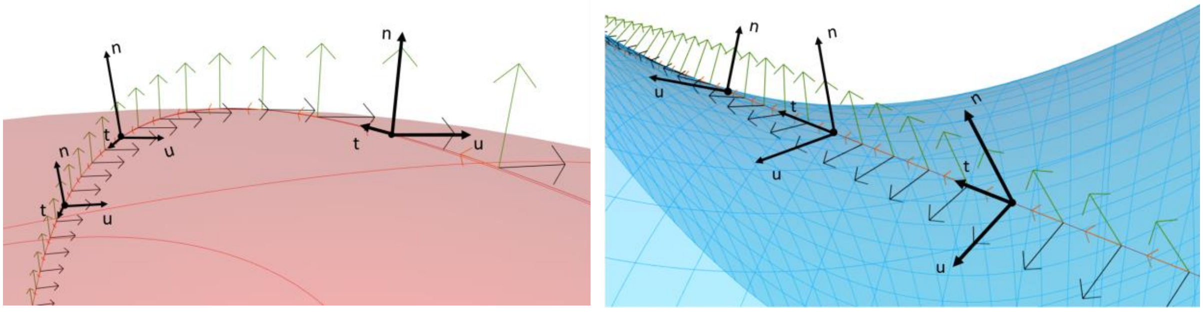


Figure 2: The rotation of vector t around the N axis is the geodesic curvature (Kg). The rotation of vector n around the u axis is the normal curvature (Kn). The rotation of vector n around the t axis is the geodesic torsion (Tg). The figure on the right shows a curve path where Kg and Kn are zero but has Tg. The figure on the left shows a curve path where Kg and Tg are zero but has Kn. The green arrow represents the normal (n), the red arrow represents the tangent (t), and the black arrow represents the binormal (u = t × n).

If the moving point's tangent vectors (directions) are defined in a way that the geodesic curvature remains constantly zero, the final curve is a geodesic curve. [16]. If the tangent vectors (directions) along which the point moves are defined in such a way that the normal curvature remains constantly zero, the final curve is an asymptotic curve [17, 18].

3 Rule direction

With the Darboux frame we can evaluate the direction of a ruled surface Ψ along the curve on a reference surface Φ. The angle α represents an angle that determines the orientation of a developable surface Ψ relative to a reference surface Φ [5]. α is the angle by which the normal vector n of the reference surface Φ is rotated to obtain the normal vector n^{α} of the developable surface Ψ. This rotation typically happens around an axis defined by the binormal vector u, which is perpendicular to both n and the tangent vector T of a curve on Φ. The formula to find the normal of the developable surface along a curve on a referenced surface is as follow [5]:

$$n^{\alpha} = \cos \alpha n + \sin \alpha u \quad (7)$$

When alpha equals 90 degrees, the cosine of alpha is zero, leading the left part of the equation to become zero. Since the sine of alpha equals 1, the normal to the ruled surface aligns with the U direction of the Darboux frame, indicating that the rule surface is orthogonal to the reference surface. When alpha equals 0 degrees, the sine of alpha is zero, leading the right part of the equation to become zero. Since the cosine of alpha equals 1, the normal to the ruled surface aligns with the N direction of the Darboux frame. This indicates that the rule surface is tangential to the reference surface.

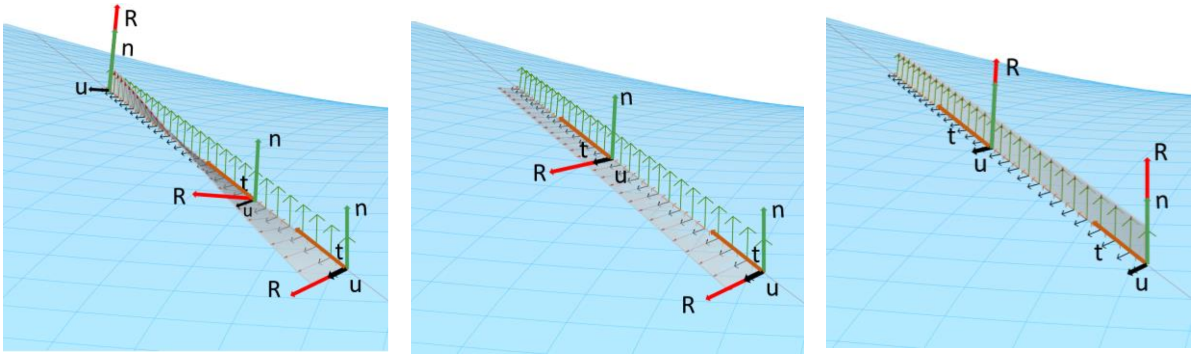


Figure 3: The left figure depicts the direction of the twisted rule direction along a curve on a surface Φ and its corresponding rule surface Ψ. The twist progresses from Alpha 0 to Alpha 90. The figure in the middle shows the rule directions tangential to the surface Φ and the corresponding Surface Ψ. The left figure shows the rule directions orthogonal to the surface Φ along the curve and the corresponding surface Ψ. The green arrow represents the normal (n), the red arrow represents the tangent (t), the black arrow represents the binormal (u = t× n), and the red arrow represents the rule direction R.

Considering the ruled surface Ψ along a curve on a surface Φ as the final structural element the alpha 90 means our structure element is perpendicular to the surface and the alpha 0 means our structure element is lying on a surface we design. The alpha between 0 to 90 indicates the twist of the profile. To create a sequence of directions that are rotating the ruling direction of surface Ψ by 90 degrees concerning the normal of a reference surface Φ, we can change the normal nα of Ψ. It means at the beginning the normal of Ψ is tangential to the normal of the Φ and gradually ends up perpendicular to it. When the surface Φ is not flat but curved, the procedure to find a chain of directions that results in a smooth rotation of the ruling direction by 90 degrees becomes more complex. From equation 7, the nα of the surface Ψ is defined from the u and n vectors of the Darboux frame of a point on surface Φ. For a single point on a surface, there is an infinite number of T directions in which to move. For a curved surface Φ, each of the T directions results in various u and n vectors, none of which may change the nα smoothly enough so that Ψ remains developable. In this case, the resulting structural strip cannot be constructed from a flat sheet.

3.1 Developable Surfaces and Ruled Surfaces

A ruled surface can be defined as a surface in which for every point on the surface, there is a straight line (ruling) that lies on the surface. Mathematically, a ruled surface Σ can be parameterized as:

$$\Sigma(u,v)=\mathbf{a}(u)+v\mathbf{b}(u) \tag{8}$$

where u and v are parameters, $\mathbf{a}(u)$ is a curve on the surface (directrix), and $\mathbf{b}(u)$ is a vector function that gives the direction of the rulings. For a ruled surface to be developable, it must also satisfy the condition of zero Gaussian curvature. This is equivalent to saying that the normal vector along a ruling does not change as one moves along the ruling. The mathematical condition for a ruled surface to be developable involves the twist vector τ being perpendicular to the tangent vector of the directrix $\mathbf{a}'(u)$, essentially indicating that the surface does not twist along the ruling. The fundamental equation related to Gaussian curvature K *[19]* for a surface in is given by:

$$K=k1\cdot k2 \tag{9}$$

where $k1$ and $k2$ are the principal curvatures at a point on the surface. For a surface to be developable, at least one of the principal curvatures ($k1$ or $k2$) must be zero everywhere on the surface, leading to K=0.

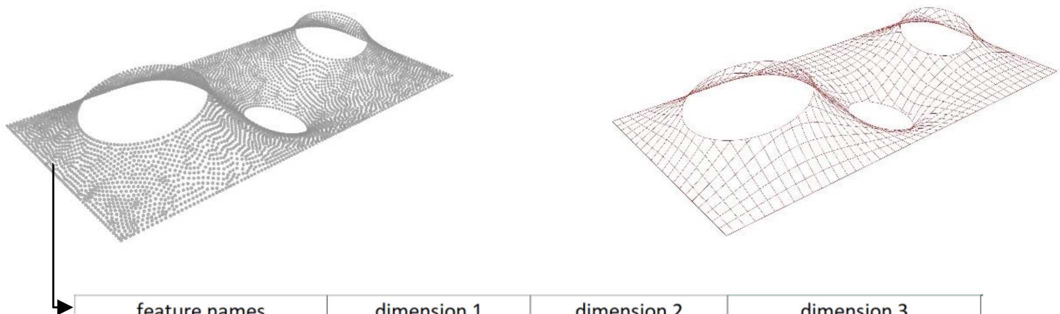
3 Reinforcement learning

Reinforcement learning, part of the machine learning field, is a system control that follows the Markov decision theory. To apply reinforcement learning the system must have the discrete time stochastic control process feature [20]. Discrete time means time moves forward in finite intervals. Stochastic means the future state depends only partially on the actions taken. The control process means the solution to the problem is defined based on decision-making to reach the target state..

3.1 The environment and the state

In reinforcement learning, we have an agent and an environment. Anything over which the agent does not have full control is considered part of the environment [20]. The agent's interaction with an environment involves actions that iterate over time intervals. The environment can be a continuing space or a discrete space. A continuous environment is like a car as an agent in a real street. In a discrete environment, such as in a game like Pacman, the agent acts as Pacman, while the environment encompasses the finite game scenes pixels, ghosts, dots, fruits, and number of the remained lives. The action space represents the freedom and the complexity of the decision-making. For instance, in the Pacman example, the action the agent can take is to decide to move in one of the four directions: up, right, down, or left. Such a limited discrete space is significantly simpler in comparison to the action space required for controlling a car to remain on the road. In our project, the agent is a point traveling on the surface to create a curve on a surface. The environment is a numerical matrix contains information of the curvature, principal curvature directions, surface normal, and spatial pattern occupancy for a set of equally distributed points on a reference surface.

The Rhino3d /Grasshopper [21] is used to generate the surface and the points. The data points and related features are exported in JSON format from Rhino3d. The JSON files are then loaded into a NumPy array format, making them operable with OpenAI Gym [22] for reinforcement learning applications. The source code will be published in a paper containing the developed numerical model.



feature names	dimension 1	dimension 2	dimension 3
coordiante	$x \in \mathbb{R}$	$y \in \mathbb{R}$	$z \in \mathbb{R}$
cruvatures	*kappa1* $\in \mathbb{R}$	*kappa2* $\in \mathbb{R}$	Gaussian curvature $\in \mathbb{R}$
principal curvature1 vector	$x \in \mathbb{R}$	$y \in \mathbb{R}$	$z \in \mathbb{R}$
principal curvature2 vector	$x \in \mathbb{R}$	$y \in \mathbb{R}$	$z \in \mathbb{R}$
surface normal	$x \in \mathbb{R}$	$y \in \mathbb{R}$	$z \in \mathbb{R}$
occupancy	boolean	boolean	boolean
rule_direction	$x \in \mathbb{R}$	$y \in \mathbb{R}$	$z \in \mathbb{R}$

Figure 4 displays a table of features for a single point on a referenced surface on which the agent moves. The environment comprises these features from all points on the surface.

During the training phase, with the initial point having no direction, the agent receives the principal curvature direction as a reference direction and has the possibility of rotation from 0 to 360 degrees to define the start direction. From the first iteration, the action space is a continuous space within the range of -30 to 30 degrees. The agent point can select a continuous decimal variable from this range to rotate its direction for movement on a referenced surface. In the prediction phase, after the training, the initial direction will be predefined for the agent, and the agent will start from the first iteration. The limitation of rotation to a range of 60 degrees (-30 to 30) ensures the smooth curvature of the final curve. If the agent has a wide range rotation possibility, it will result in sharp turns that cause high curvature, leading to a high torsion rate that requires a very flexible material for building. In addition, it significantly increases the probability of directions that are non-developable. Based on our experiments and taking into account the geometrical considerations explained in Chapter 2 regarding torsion and the developable surface, we established the range of -30 to 30 degrees in such a way that it does not restrict the agent's freedom for exploration, while simultaneously reducing the error domain thus accelerate the learning process.

The state contains the complete information about the environment that the agent observes in order to take an action. After each action, the environment will be updated based on the agent's interaction. In our research, the state is a local neighborhood of numerical data around the agent. The KNN (K-number of Nearest Neighbors) algorithm [23] is used to take a local matrix of size 7x32x3 around the point (agent location) from the environment to define the state. The dimension of the state matches that of the environment, with a limited number of data points being observable. The action that the agent selects from the action space depends on the policy the agent follows. For a given state of an environment, the same agent could take different actions based on different policies. Over the training process, the agent learns to find the optimal policy that leads to better actions. More detailed explanations will follow after the clarification of the reward in the next chapter.

3.3 The trajectory and rewards

The chain of the actions forms a trajectory which in our project is a path of a curve on a surface. After each action, the agent point moves along the action's direction and creates a small line segment. The chain of the segments connecting forms a curve. Once the trajectory reaches the final state it is completed. If the agent cannot complete a trajectory within limited time of intervals, the trajectory will be terminated. The limitation of the interval depends on the target of the training, size of the environemnt and etc. The finished trajectory, either by reaching the goal or termination, forms an episode. There are two types of rewards: immediate and final. The immediate reward is the gain that directly follows once the agent makes an interaction. The final reward, called the "Return," is the sum of the discounted rewards in an episode. The discount value is a squared gamma value that reduces the immediate reward of the actions as the time interval increases in an episode. If an agent doesn't receive any instant rewards, the agent faces the delayed feedback situation, making it harder for the agent to associate which actions lead to positive outcomes. Monte Carlo Methods in reinforcement learning are example of a learning with delayed feedback [24]. These methods wait until the end of an episode and then use the cumulative reward for learning. In general, in reinforcement learning the objective is to find the optimal policy to maximize the long-term sum of the discounted of all rewards.

We have three criteria to define the reward for the agent. The first criterion is related to the geometry of the curve, the second is related to the structure, and the third is related to the connection of the pattern. For the geometry, we aim for the curve to have smooth curvature characteristics, making it possible to be built with a bend and twist of a flat strip. For the structure, we target the deflection of the structure that could be reinforced with a proper profile direction. For the pattern connection, we aim for the maximum number of elements at a certain point that can intersect. Each of these rewards is explained in detail below.

3.3.1 Reward for geometrical features

Choosing between instant and delayed rewards to evaluate if the generated curve of the agent on a surface Φ traces a constant developable surface Ψ is challenging due to the computational cost of the process and the geometrical evaluation involved. Instant rewards provide immediate feedback and help in quick learning but risk focusing on short-term gains over the overall objective, potentially leading to suboptimal paths. Delayed rewards encourage optimization towards the global goal and simplify reward design, but they slow down the learning process due to infrequent feedback. Additionally, they complicate the identification of which actions were beneficial. The best approach would be to blend both quick learning adjustments and global solution evaluation. This combination allows for leveraging the benefits of immediate feedback for quick learning while ensuring optimization towards the overall objective through delayed rewards.

3.3.2 Reward for structural stiffness

To evaluate the structural characteristic of the generated curve of the agent on a surface, we consider the instant reward based on stiffness respectively the momentum of inertia. For simplicity, the segments of the agent's movements on a surface will be considered as beams that are connected in a row. The surface that constitutes the environment encounters both highly curved and low curved areas. In the low curvature area, the bending axis of the theoretical beam segment is aligned with the u vector of the Darboux at the agent point location. In this case, the profile direction of the ruled surface along the direction of the segment beam according to the surface indicates the momentum of the inertia. Because the structural element will be built from one continuous element, the modulus of elasticity is always the same. The moving step of the agent is also constantly the same, which means the beam segments have the same length. As a result, the maximum deflection *ymax* occurs where there is less moment of inertia *I*. In this context, based on the curvature change, the agent receives a higher reward as long as the profile stays orthogonal to the surface when the curvature is low.

3.3.3 Reward for connection

During the path trace, as the agent observes the state, it will check the occupancy feature of the closest data point to itself. If the status occupancy has all three dimensions True, it means the agent has reached a node that is already fully occupied. In addition, if the closest data point has a "True" occupancy value the agent checks the rule direction feature of the data point. With measuring the deviation between the rule direction of the data point and the agent, if the deviation equals zero, the episode finishes. If there is a deviation and the agent collapses, meaning the trajectory is terminated.

4 Interaction of the RL model with exemplary environments

This chapter presents two experiments involving the RL agent's exploration in creating curves on the surface to model developable strips. The first experiment focuses on if and how the agent will find a solution to find a developable, maximum length strip on a given surface. In the second experiment, the agent experiences the first step to generate a pattern. The bounding box encompassing the surfaces of the environments has the size of 120x120x60 cm for experiment I and 130x130x100 cm for experiment II. The size of the surfaces inside the aforementioned volumes provides enough area for the agent, with a step size of 1.5 cm, to generate smooth transitions of the strip profile from tangential to orthogonal orientation to the surface and vice versa, including all orientation angles in-between. From our experiments in previous research, we observed that the transition between geodesic and asymptotic curve types, which necessitates a twist in the strip's profile, occurs smoothly within a range of 70 cm. The same range is considered in the presented experiments I and II. The agent can traverse the surface in 90 steps, and a maximum time interval of 300 is considered to ensure that the agent has sufficient time for exploration.

4.1 Experiment I

The target of experiment I is to check if and how the agent manages to find the longest possible, developable strip on the surface while also ensuring maximum structural stiffness, which is assumed depending on magnitude of curvature of and profile orientation to the surface. More specifically, this experiment explores whether and how the agent perceives the environment and updates its policy to make decisions in controlling geometric modeling. The surface of the environment is the saddle-shaped surface of a hyperbolic paraboloid. The occupancy feature of all data points of the environment of the mentioned surface is “False”. This means that the surface is not populated with any strip profiles. As the agent moves on the surface, iteratively it changes the occupancy features of the data points it passes to “True”. In this experiment, the two reward functions of the geometry and the structural stiffness are considered by the agent.

Proximal Policy Optimization (PPO) [25] is used for the agent policy optimization during exploration. Monitoring the exploration process shows the spread of the action values is decreasing over the exploration, indicating that the policy's action distribution is becoming more confident. In addition, monitoring the proportion of the variance and errors in the returns that are predicted by the agent, alongside the spread of the action values, illustrates two aspects. First, over the iterations, the agent gains confidence in predictions as by avoiding populating highly curved areas with asymptotic strips, it can reduce the loss of reward from the moment of inertia. Second, the agent gains confidence that it can increase the length of the strip by traveling along the less curved areas of the surface that have a larger portion, which in turn leads to higher rewards. Figure 5 (right) clearly illustrates, that the agent tried to reach the corner of the surface to obtain more rewards from the diagonal path instead of the direct path on the surface.

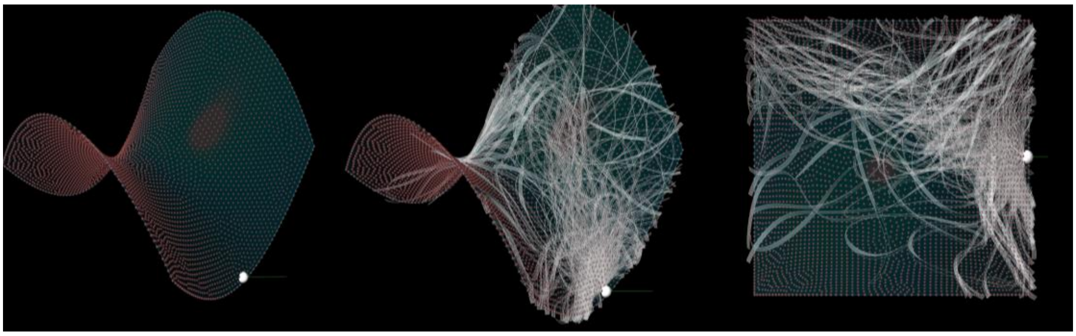


Figure 5: The environment of the training consists of a set of equally distributed points on a surface. (left) The agent´s exploration of the surface in 50000 iterations in perspective (center) and top (right) view.

In addition, more moment of inertia can be gained in areas of low curvature by adjusting the strip´s profile direction perpendicular to the surface on these paths. This leads to an increase in path length and stiffness gain. Consequently, the agent receives more rewards.

4.1 Experiment II

The target of the experiment is to check if and how the agent manages to find a solution for completing a pattern on a surface under consideration of the structural stiffness and of the developability of strips. The environment consists of an anticlastically curved cone-surface with a set of equally distributed points and an initial pattern of geodesic curves on it as shown in Figure 6 (left). The occupancy feature of the data points on the surface represented in red color, respectively where the patterns are, is “True”. The agent starts from the top strip and aims to find a suitable path to add a new strip and consequently form a pattern.

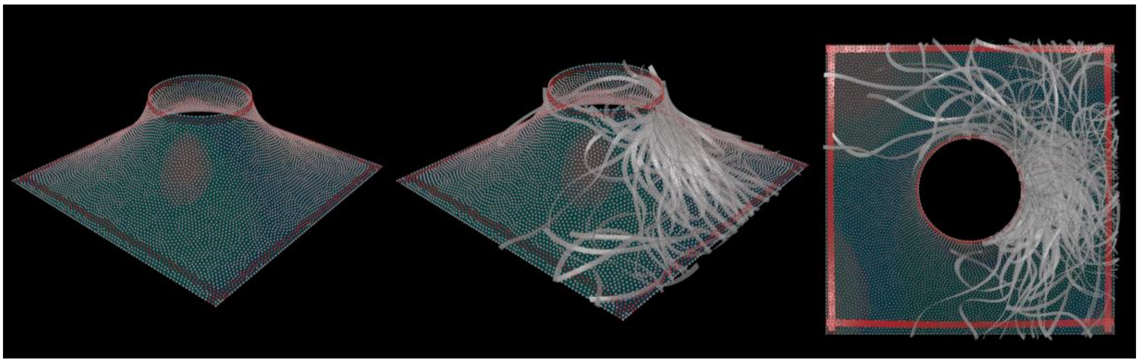


Figure 6: The environment with a set of equally distributed points on a surface (right). The points in red color have occupancy value “True”. The agent`s exploration of the surface over 50000 iterations (center and right).

The same Proximal Policy Optimization (PPO) as in experiment I is used for agent policy optimization in experiment II. In this experiment, the agent encounters a new environment that has a set of patterns in the initial state of the exploration. As Figure 6 illustrates that the agent could explore the surface by tracing smooth curves. This indicates that the action space provides adequate freedom in the exploration of the environment. The areas with denser lines indicate spots where the agent has spent more iterations, implying that these areas are more critical to predefined objectives. Monitoring the proportion of the variance in the returns that are predicted indicates that the agent has identified the areas with smooth transitions of the curvature as significant for optimizing the structural element, focusing its learning on fine-tuning the solution. Figure 7 (left) shows that the focused area of the agent in the environment is where the agent could provide both more orthogonal rule direction in low curvature areas and meet the existing pattern members with stable rule direction, means tangentially or orthogonally.

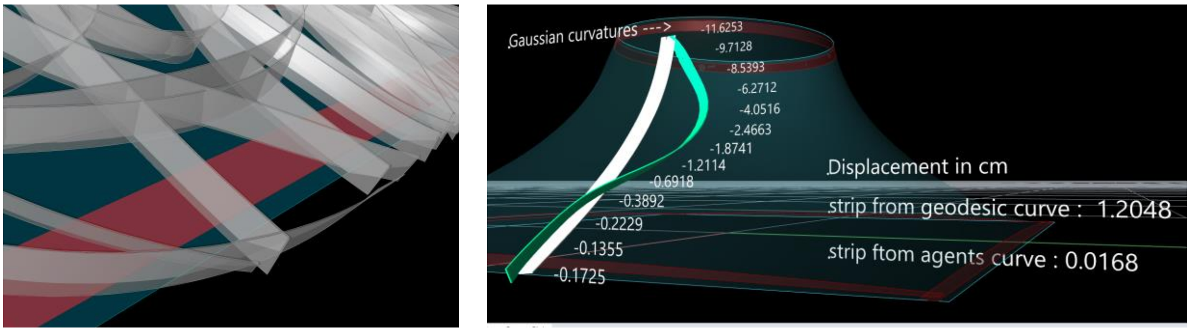


Figure 7: The comparison between the agent’s curve a and the geodesic curve manually added by the author to generate a strip. The numbers indicate the Gaussian curvature value at specific spots on the surface.

As shown in Figure 7 (left), the agent's policy leads it to follow the path that provides higher momentum of inertia based on Gaussian curvature change. In Figure 7 (right), a strip manually added by the author (Figure 7 (right) white strip) is compared with the agent's suggestion for finding a strip to complete a pattern. The author´s curve is geodesic, representing a fast and simple way to generate a developable strip and pattern. In the same condition, the agent finds a strip path (Figure 7 (right) green strip) with consideration of surface curvature and profile orientation to the surface. The agent´s solution results in a curve that follows a geodesic orientation and turns into an asymptotic orientation where the surface has lower curvature. The comparison shows that the reward function improves the agent's policy in increasing the stiffness, where the resulting strip mesh of the curve that the agent generated has a lower displacement value. The karamba3d mesh analysis is used for the comparison of the displacements. The start and end points of the strip meshes are considered as support, indicating connection to the existing pattern, the strips are regarded with a thickness of 6.5 mm. The length of the geodesic curve and the curve generated by the agent are respectively 68 cm and 82 cm.

5 Discussion and Conclusion

This paper presents geometrical solution finding, controllable by a Reinforcement learning model, for creating a pattern of bending-active strips on a surface. The developed methodology has been tested in two experiments with species tasks, detailed in Chapter 4. The experiments aimed to assess whether and how the RL agent observes its environment and adjusts its policy when selecting actions to control geometry generation. The results of the first experiment illustrate that the structure of the observable environment, detailed in section 3.1, provides meaningful geometrical information that helps the RL agent boost its policy in decision-making. The result of the second experiment illustrates the reward set strategies, detailed in section 3.3, are effective so that the agent optimizes its policy aligned with the goal of the task defined in the experiment.

More experiments in various environments are required to make the agent capable of decision-making in states encountering complex pattern generation. To enable the agent to improve its proficiency in the design of complex patterns, we consider a set of parallel training in different environments that will be presented in future studies. The structural analysis of the overall pattern beside the individual strip is beneficial for the agent to gain awareness of the consequences of its actions. However, fine-tuning the reward functions for this purpose presents a challenging aspect of research and could potentially lead to hierarchical and sequential action selection within the action space. Moreover, this adjustment will lead to an increase in the observable features of the environment, thereby enhancing the complexity of the agent's perception. In general, the successful implementation of the geometrical system capable of self-learning paves the way for a comprehensive research, wherein we aim to develop a generative model supporting designers in crafting non-repetitive, adaptable bending-active structures. In our future work, we aim to enhance the features of the geometrical system and the agent's perception capability, to extend the RL model proficiency in the design of complex bending-active surface structures.

References

[1] P. Nicholas, D. Stasiuk, and T. Schork, "The social weavers: considering top-down and bottom-up design processes as a continuum," in *ACADIA, Proceedings of the 34th Association for Computer Aided Design in Architecture*, 2014.

[2] S. Brancart, L. De Laet, A. Vergauwen & N. De Temmerman, "Transformable active bending: a kinematical concept,"

[3] M. H. Saleh Tabari, F. Khojastehmehr, and G. H. Filz, "Parametric thinking for decision-making in elastic gridshell design," in *Proceedings of the IASS Annual Symposium 2022*, pp. 19–22.

[4] J. Lienhard, H. Alpermann, C. Gengnagel, and J. Knippers, "Active Bending, a Review on Structures where Bending is Used as a Self-Formation Process," *International Journal of Space Structures*, vol. 28, 3-4, pp. 187–196, 2013, doi: 10.1260/0266-3511.28.3-4.187.

[5] C. Tang, M Kilian, P. Bo, J. Wallner, and H. Pottmann, "Analysis and design of curved support structures," in – *Advances in Architectural Geometry*, 2016.

[6] C. Tang, P. Bo, J. Wallner, and H. Pottmann, "Interactive Design of Developable Surfaces," *ACM Trans. Graph.*, vol. 35, no. 2, pp. 1–12, 2016.

[7] S. Flöry and H. Pottmann, "Ruled Surfaces for Rationalization and Design in Architecture," in *ACADIA 2010 ife in:formation*.

[8] M. H. Saleh Tabari and G. H. Filz, "Cognitive multi-pattern gridshell design: a method inspired by weaverbirds' nest building strategies," in *Proceedings of IASS 2023*.

[9] I. E. Bailey, A. Backes, P. T. Walsh, K. V. Morgan, S. L. Meddle, and S. D. Healy, "Image analysis of weaverbird nests reveals signature weave textures," *R. Soc. open sci.*, vol. 2, no. 6, p. 150074, 2015, doi: 10.1098/rsos.150074.

[10] P. T. Walsh, M. Hansell, W. D. Borello, and S. D. Healy, "Repeatability of nest morphology in African weaver birds," *Biology letters*, vol. 6, no. 2, pp. 149–151, 2010.

[11] P. T. Walsh, M. Hansell, W. D. Borello, and S. D. Healy, "Individuality in nest building: do southern masked weaver (Ploceus velatus) males vary in their nest-building behaviour?," *Behavioural processes*, vol. 88, no. 1, pp. 1–6, 2011.

[12] M. Hansell and R. Overhill, *Bird Nests and Construction Behaviour*: Cambridge University Press, 2000.

[13] N. E. Collias and E. C. Collias, "An Experimental Study of the Mechanisms of Nest Building in a Weaverbird," *The Auk*, vol. 79, no. 4, pp. 568–595, 1962.

[14] Lia Vas, *Differential Geometry*, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID: 48629872

[15] E. S. Y. TOPBAŞ, İ. GÖK, N. EKMEKCİ, and Y. YAYLI, "Darboux Frame of a Curve Lying on a Lightlike Surface," *Mathematical Sciences and Applications*, pp. 121–130, 2016.

[16] *Geodesics revisited*, 2012. [Online]. Available: http://www.cmsim.eu/papers_pdf/january_2012_papers/25_cmsim_2012_pokorny_1_281-298.pdf

[17] T. Oberbichler, E. Schling, and K.-U. Bletzinger, "Tracing curvature paths on trimmed multipatch surfaces," *Applied Mathematical Modelling*, vol. 118, pp. 253–271, 2023, doi: 10.1016/j.apm.2023.01.033.

[18] E. Schling and R. Barthel, "Repetitive Structures," in *Gengnagel, Baverel et al. (Ed.) 2020 – Impact: Design With All Senses. DMSB 2019*, pp. 360–375.

[19] J. M. Sullivan, "Curvatures of Smooth and Discrete Surfaces," *Discrete differential geometry*, 2007.

[20] R. S. Sutton, F. Bach, and A. G. Barto, *Reinforcement Learning: An Introduction,* 2nd ed. Massachusetts: MIT Press Ltd, 2018.

[21] R. McNeel and Associates, *Rhinoceros 3D.* [Online]. Available: https://www.rhino3d.com/

[22] G. Brockman *et al.,* "OpenAI Gym," Jun. 2016. [Online]. Available: http://arxiv.org/pdf/1606.01540

[23] D. Coomans and D. L. Massart, "Alternative k-nearest neighbour rules in supervised pattern recognition," pp. 15–27, 1982, doi: 10.1016/S0003-2670(01)95359-0.

[24] T. Hester and P. Stone, "TEXPLORE: real-time sample-efficient reinforcement learning for robots," *Mach Learn*, vol. 90, no. 3, pp. 385–429, 2013, doi: 10.1007/s10994-012-5322-7.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017.