Environmental Technology & Innovation 24 (2021) 101819

Contents lists available at ScienceDirect

Environmental Technology & Innovation

journal homepage: www.elsevier.com/locate/eti

Online design of green urban garden landscape based on machine learning and computer simulation technology

Jing Luo

Art Colleges, Chongqing Technology and Business University, Chongqing 400067, China

ARTICLE INFO

Article history:
Received 10 November 2020
Received in revised form 3 January 2021
Accepted 23 July 2021
Available online 26 July 2021

Keywords:
Green urban garden landscape
Parametric design
Computer simulation technology
Machine learning

ABSTRACT

With the development of machine learning and computer simulation technology, green urban garden landscapes have also made breakthrough progress in modeling. Based on the existing basic technology and its shortcomings, this article discusses the timeliness of the Agent model, defines the time granularity of simulation machine learning, and establishes an Agent model related to the time granularity of machine learning. The corresponding time management mechanism under real-time simulation is constructed, and the switching of machine learning time granularity, real-time guarantee and message interaction mechanism are studied. We use the new model to model the spatial entities to illustrate the modeling method, and design the message interaction between the spatial entities. We analyze the parametric works in the design world from the aspects of nature, art and computer, and summarize the general law of parametric design. Computer-aided design includes the study of traditional calculation-aided design methods. This paper analyzes its existing problems and proposes new methods of parametric design and the generation of computer geometric algorithms. We compare with Auto CAD, which is commonly used in garden and landscape design, and summarize their advantages and disadvantages. The application of the machine learning Grasshopper parameter platform to the design process of actual projects fully reflects its scientific, rational and advanced nature.

1. Introduction

With the development of science and technology, major design industries have also undergone breakthrough changes in modeling (Ullah et al., 2020). Modern digital design uses twisted curved surfaces to divide the architectural space, creating cool and impactful architectural forms, such as the Harbin Opera House and the "Bird's Nest" and "Water Cube" of the Beijing Olympics. In garden landscape design, designers want to pursue some morphological breakthroughs, such as heterogeneous curved surfaces, which are often restricted by various technologies or software, so they are forced to return to traditional design (Park and Guldmann, 2019; Vij et al., 2020; Pérez-Rave et al., 2019). The Grasshopper plug-in for machine learning under Rhino software was originally a commonly used software in product design, mainly for the construction of special-shaped surfaces. And it has its own programming function, you can edit the program logic and change the shape of the surface at any time (Sweatt et al., 2019).

A researcher of architectural technology at the Technical University of Berlin, Germany, used Python to write a three-dimensional visualization model for space thermal analysis simulation (Belesky, 2020). Its main function is to use thermal analysis software to calculate Energy Plus thermal energy and Daysim as the data source, input it into the machine learning

E-mail address: 2010a62@ctbu.edu.cn.

https://doi.org/10.1016/j.eti.2021.101819
2352-1864/

Grasshopper for data analysis, and use different colors to distinguish different values instead of using chart data (Oishi, 2019; D'Odorico et al., 2020). By not using the contrast and difference of colors, the design can easily determine which part of the building has strong light and absorb heat, and adjust the building form according to actual needs (Vartiainen et al., 2020). Related scholars have developed Netlogo (simulation model), which is a software that analyzes the impact of hurricanes on the beach on the damage of buildings (Tobey et al., 2019). DIVA-for-Rhino is a plug-in developed by the Harvard University Graduate School of Design. It is a highly optimized lighting and energy analysis software. It has a series of environmental quality assessment functions, and realistic rendering of personal buildings and urban landscapes including radiation maps (Yang et al., 2019). And according to the accumulation analysis of time, the LED lamp evaluates and calculates the energy and load accumulated by the single area irradiation (Bird, 2020). Both Galapagos and Octopus are feedback optimization plug-ins, which are single-objective and multi-objective numerical optimal solution calculators developed by engineers from the Vienna University of Applied Arts (Reviglio, 2019). The purpose is to find the optimal solution set to feedback and adjust the initial input data combination. The British Architectural Alliance Institute has set up landscape urbanism in the discipline of architectural design, mainly trying to cross different scales of artificial ecology to promote the unity of urban organization and adapt to the instability of urban development (Zhang et al., 2020; Tang, 2019). Landscape urbanism is divided into four courses. The first is site indexation, which mainly studies hydrology, land value and utilization, and land material flow. The second is the sensitive system, which is mainly the analysis within the site, discussing the connection between the units, and showing its complexity in the process. The third is that the network city analyzes the overall behavior of the city and establishes the logic between the units of the city. The fourth is to evaluate the designed site. Supervised classification refers to classification based on typical samples of various ground features collected on remote sensing images, based on statistical recognition functions, and based on typical sample training methods. Parameter classification is a classification method that is based on the probability distribution form of the feature category, carries out model estimation and uses the model to determine the boundary of the classifier (Jahani and Rayegani, 2020). The commonly used maximum likelihood classification is based on the probability model estimated from the training data based on the normal distribution of the sample, thereby determining the classification boundary. Non-parametric classification means that the classification algorithm does not make any specific probability distribution assumptions for the input category data, and is flexible and robust to the nonlinear and noise relationship between the input features and the category labels (Zimmerman et al., 2019). As long as the signal features are obvious, it can be very effective. Its advantage is high classification flexibility, even if the wrong distribution model is selected, the non-parametric algorithm can also have a good classification effect (Valdivia et al., 2020). Machine learning algorithms are usually non-parametric. Weave Bird is a surface modeling based on Mesh mesh, including several mesh refinement and model conversion tools. It is mainly used for mesh smoothing, editing, and rapid generation of relatively regular complex mesh shapes, eliminating the need for repetitive work. The boring process thus provides considerable convenience for designers. Although the modeling accuracy of Mesh is indeed inadequate compared to Nurbs, it should not be ignored that some models are more handy to make mesh, such as the dendritic structure of multiple pipes (Paynter et al., 2019; Zhang et al., 2019).

This paper discusses the time-related Agent model, the time granularity of machine learning, and the theoretical model of the Agent related to machine learning time granularity. It studied how to add the characteristics of machine learning time granularity and the switching process of machine learning time granularity in the general Agent platform. It discusses the time management mechanism in real-time simulation in the Agent platform with machine learning time granularity, including time-driven and related synchronization methods. According to the generation logic of machine learning Grasshopper, the application and advantages of Tyson polygon, group intelligent generation and parameter logic in garden landscape are introduced respectively. From line drafting, layer management, space modeling and surface generation, the machine learning Grasshopper is compared with Auto CAD and Sketchup traditional garden landscape software-assisted design, so as to explore the advantages of machine learning Grasshopper. Practical operations are carried out on the main scene sculpture design, plane scrutiny, landscape bridge design, landscape wall design and garden sketch design in garden landscape design. Machine learning Grasshopper becomes a tool for "finding" solutions in the design process. This will replace the traditional design software Sketchup, thereby promoting garden landscape design to be more innovative, scientific and convenient.

The rest of this article is organized as follows. Section 2 discusses the related technologies of machine learning and computer simulation. Section 3 analyzes the time-granularity Agent modeling and computer simulation of machine learning. Section 4 conducted a case study of the online design of green urban garden landscape. Section 5 summarizes the full text.

2. Related technologies of machine learning and computer simulation

2.1. Shallow learning

Shallow learning refers to a machine learning model that contains nodes in the hidden layer below one layer. Support vector machine is a new type of machine learning method based on statistical theory. It is a non-parametric supervised learning classifier based on small sample research. By studying small samples, aiming at the risk minimization principle, that is, only emphasizing the shortcomings that exist under the minimum empirical risk error of the training sample,

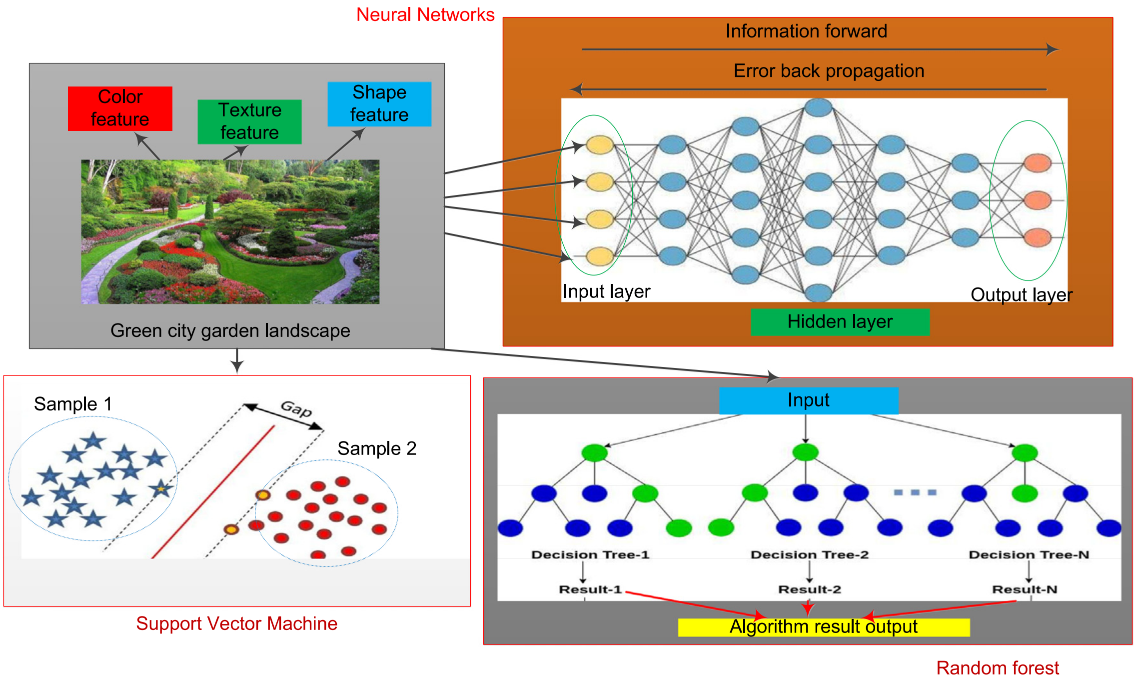


Fig. 1. Schematic diagram of machine learning classifier.

comprehensively considering the complexity and learning ability of the model, the VC theory and the structural risk minimization principle replace the empirical risk minimization. They realize the optimal classification of linearly separable data, make it learn to get better promotion ability. For nonlinear situations, the main idea of SVM is to map the input vector to a high-dimensional feature space by defining an appropriate kernel function, construct an optimal separation hyperplane in the high-dimensional feature space, and convert the nonlinear space problem into a linear problem. In this way, the non-linear data that is not easy to separate on the plane is separated, and the optimal linear classification surface is obtained in the feature space to maximize the classification interval, which can finally be transformed into a convex quadratic programming problem. At the same time, it effectively reduces the complexity of the algorithm. The schematic diagram of the machine learning classifier is shown in Fig. 1.

The original design of the support vector machine only considered the binary classification problem in theory. The specific implementation process is to find an optimal classification hyperplane in the given adjacent two types of samples. The distance between the edge sample points and the classification hyperplane reaches the maximum. The sample points on the maximization edge are called support vectors.

Libsvm can combine multiple support vector machines for multi-class problems. It is divided into a "one-to-many" classification method. When the training sample has p categories, to construct a classification hyperplane between the ith and other $p - 1$ classes, p binary classifiers need to be constructed. Therefore, in the "one-to-many" method, there are p convex quadratic programming problems for p categories and multi-classification problems, and p classification decision functions are obtained by solving p optimization problems. However, this method will lead to uncertainty in the results, that is, any classifier does not include the sample, resulting in indivisible phenomenon. The most commonly used method is the "one-against-one" rule, which separates each paired class combination, while the one-to-one method is based on the separation of each class from the rest. If there are p categories, $p(p - 1)/2$ classifiers need to be established, and voting is used to determine the final category of data to be predicted. In the actual remote sensing image classification process, when there are fewer classification categories, only a single support vector machine method is used for classification without considering the time cost. That is, the samples that need to be classified are all regarded as positive samples, and the others are all negative samples.

Based on the Weka data mining platform, the support vector machines available are C-SVM and nu-SVM. Both are single support vector machines that deal with binary classification problems. The main difference lies in the parameter C and the parameter nu. The parameter C can take any positive value, and the parameter nu is limited by 0 and 1, which represent the lower and upper limits of the number of examples that are respectively used as support vectors and located on the wrong side of the hyperplane. The advantage is that it is related to the ratio of support vectors and the ratio of training errors. However, in the actual application process, nu-SVM is relatively difficult to optimize, and compared with C-SVM, the runtime is usually not scalable.

In the process of support vector machine classification, data normalization is a prerequisite that affects the classification accuracy of support vector machines. On the one hand, the normalized data can avoid the influence of extreme values; on the other hand, it can also reduce the calculation difficulty in the statistical process. In specific data processing, the attribute value is usually normalized to [−1,1] or [0,1].

2.2. Deep learning

With a single hidden layer structure, the performance of a hidden layer node will be improved during the learning process of the network, which will affect the performance of other neighboring nodes. And it can only do linear classification tasks. If a multi-hidden layer structure is adopted, the independence of different nodes in the hidden layer will be stronger, and the probability of mutual influence will be much smaller. A neural network with multiple hidden layers, that is, a BP network with more than 2 hidden layers, is a typical deep neural network. Deep neural networks can use "hierarchical" information expression to reduce the number of parameters that need to be set in the network, and use the depth of the network to replace the additional exponential scale calculation factors that may be less than the depth, which can effectively reduce the complexity of network calculations. The objective function is used to minimize the difference between the predicted output and the expected output:

$$D(x, y; W, b) = 0.5 \cdot \left|y - h_{(W,b)}(x)\right|^2 \tag{1}$$

The structure of deep neural networks is difficult to determine, and there is a problem of structural optimization. The appropriate network depth is often related to specific applications and data sets. The number of input layer nodes is generally the feature dimension of the remote sensing images participating in the classification, and the number of output layer nodes is the same as the number of categories. The best hidden layer and the number of nodes in each layer are often determined by the problem itself and by trial and error. If the number of hidden layer nodes is too small, the backpropagation algorithm will not be able to focus on the minimum during training. On the contrary, too many hidden layer nodes will cause the network to overfit the data.

There is no exact mathematical formula to solve the number of neurons in the hidden layer. According to the idea of data dimensionality reduction or sparse expression based on the number of hidden layer units and the relative size of the input layer, the following formula can be used to determine the initial input amount. For the subsequent increase in the number of hidden layer nodes, generally a double-fold method can be used to achieve satisfactory results.

$$A = \frac{M \cdot N + 0.5 \cdot M \cdot (N + N^2) - 1}{M + N} \tag{2}$$

In the formula, A represents the number of hidden layer nodes, m is the number of classifications, and n is the dimension of the feature vector.

The Sigmoid nonlinear activation function commonly used in traditional BP artificial neural networks, due to the saturation effect of the Sigmoid function, will cause gradient loss to larger and smaller input data, and the output gradient is not centered at 0, so it will be caused in the gradient descent stage. As the number of network layers increases, the phenomenon of "gradient disappearance" becomes more serious. Specifically, we often use sigmoid as the input and output function of neurons. For a signal with an amplitude of 1, when the BP propagates back the gradient, the gradient attenuates to the original 0.25 for each layer passed. If the number of layers is large, the lower layer basically cannot receive effective training signals after the gradient exponentially attenuates. Compared with the previously commonly used Sigmoid non-linear activation function, since Re Lu only needs a threshold to get the activation value, and it is an unsaturated linear function, the convergence speed in the stochastic gradient descent stage is faster than that of Sigmoid.

The PSE index is the ratio of the total area of the under-segmented area to the total area of the reference polygon, which indicates the degree of under-segmentation:

$$PSE = \frac{\sum |R_t - s_i|}{-\sum R_t} \tag{3}$$

In the formula, s_i represents the under-segmented object data set, and R_t represents the under-segmented reference data set.

The matrix estimation problem can be described by a discrete state space model. The discrete form is mainly studied here, and the structure is as follows.

The state transition equation is:

$$X_{t+1} = aX_t - bU_t - |W_t| \quad t = 0, 1, 2, \ldots, n-1 \tag{4}$$

The measurement equation is:

$$Y_t = cX_t + dU_t - |V_t| \quad t = 0, 1, 2, 3, \ldots, n-1 \tag{5}$$

The above formula describes the transition relationship between the system state vector X_{t+1} (assumed to be s-dimensional) in the period t + 1 and the system state X_t in the previous period.

According to the multivariate statistical analysis theory, the unbiased estimate of the measurement error variance R_t is as follows:

$$R_t' = \frac{\sum_{i=1}^{n} (v_t - \overline{v_t})^T (v_t + \overline{v_t})}{n-1} \tag{6}$$

2.3. J2EE platform and SSH, SSM framework

(1) J2EE platform

Java 2 Enterprise Edition (J2EE) has a large number of components and solutions to improve simplified code development and system performance. For example, the memory recovery mechanism can automatically and efficiently recover available memory. Programmers do not need to pay attention to memory application and recovery like C++ and other languages. It not only simplifies the code, but also lays the foundation for the high-performance operation of the program. At the same time, the safety mechanism can meet various safety requirements. The J2EE platform is composed of multiple parts, each part is responsible for different tasks, but there are interactive interfaces between each other. Compared with the traditional Internet application development technology architecture, J2EE technology has obvious advantages.

In general, J2EE is an architecture that can effectively improve the efficiency of enterprise information system development. J2EE technology not only inherits many advantages of the Java platform, such as the "write once, run anywhere" feature, unified data access interface JDBC, CORBA technology, and a security model that can protect data in Internet applications.

J2EE client-level components can be seen as a combination of front-end pages. Some of these pages are static pages, and all elements can be displayed in the browser of the client, while other pages are dynamic pages. Some or all of the elements require the server to perform calculations and it will be available after explanation.

The web layer component is mainly responsible for the interaction between the client and the server. This interaction may include multiple handshake and confirmation. The data transmission and background operation of the dynamic page require the participation of the web layer component to ensure efficiency and accuracy.

The business layer component is essentially a business logic layer, which is mainly responsible for the realization of multi-thread safety, fault handling and data business logic functions. Its main implementation is on the server side, using the powerful resources of the server to monitor the entire business.

Enterprise Information System (EIS) mainly includes low-level modules such as enterprise runtime and ERP, as well as supporting systems such as database storage systems and operating systems. EIS manages these low-level supporting modules in a unified manner to ensure the stability and reliability of the system. At the same time, EIS also plays a role in shielding the details of the underlying system to the upper level.

(2) SSH framework and SSM framework

SSH is the abbreviation of an integrated framework, composed of Struts + spring + hibernate, and is currently a popular web application development framework in network technology.

The integrated SSH framework improves technical support and assistance for R&D personnel to build a Web application with reasonable logic, complete structure, strong cycle performance, and easy functional maintenance in a relatively short period of time. Therefore, the framework contains multi-layer components similar to J2EE: presentation layer (used for interaction), business logic layer (used to process business logic), data persistence layer (used for data processing).

It can be seen that Struts is a key part of the whole system, because it is mainly responsible for implementing the separation function of MVC and fulfilling the role of connecting (Spring) and connecting (Hibernate). Such a development model has great advantages, because it realizes the complete separation of view, controller and model, and achieves non-interference between the business logic layer and the data persistence layer, so that no matter how the front end changes, the corresponding flexible minor changes, and the changes in the back-end database will not cause front-end changes and impacts. This greatly facilitates the maintenance of the system's functions and improves its reusability.

In essence, Hibernate can be seen as a mapping tool, which maps elements (tables, tuples, etc.) in a relational database to objects in JAVA code. In other words, based on the Hibernate framework, database developers and JAVA developers in the project team can be independent of each other. Business logic developers do not need to pay attention to the processing of data sets in the database, while database developers can only be responsible for the conversion between objects in the database and JAVA classes.

My Batis is a free software released under the Apache License 2.0, which is further optimized on the basis of the Hibernate framework. Database developers no longer need to directly generate the mapping between database objects and JAVA classes, but connect objects with stored procedures or SQL statements by writing XML files. The mapping is the relationship between Java methods and SQL statements.

3. Machine learning time granularity Agent modeling and computer simulation

3.1. Timeliness of the Agent model

To facilitate the description, first we discuss the timeliness of the Agent model. Generally, during the entire life cycle of an agent, its behavior and status will change over time. Similar to the object-oriented concept, the properties of its

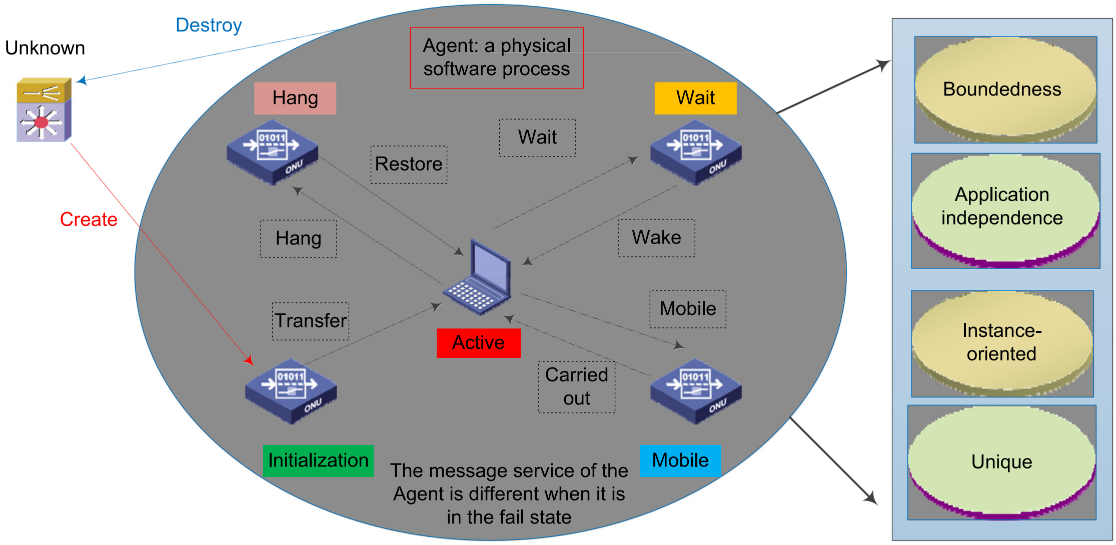


Fig. 2. Agent's life cycle diagram.

objects will change over time. Although the status and behavior of an agent instance will change with time in its life cycle, generally when using Agent technology for simulation, the agent's modeling of the simulation object does not change with time. Agent life cycle is shown in Fig. 2.

When we simulate a certain simulation object, under different circumstances, its modeling requirements may be different. Even in a simulation, the simulation target for the simulation object is different at different times, resulting in the construction of the simulation object. In the AUML class of Agent considering time, the change of Agnet model is mainly reflected by the state descriptions St(t), A(t) and M(t).

Then in the simulation, the Agent can be called in a cycle of 1 s, or the Agent can be called in a cycle of 10 s. The Agent model established for the satellite is different, it is the Agent model of different machine learning time granularity. In simulation applications, a simulation object may run with a machine learning time granularity model, or it may switch to different machine learning time granularity models.

The state descriptions, behaviors, methods, capabilities, service descriptions, support protocols, group representations, and agent head automata of different machine learning time granular models may be completely different or partly the same. A more common situation is that when the time granularity of machine learning changes and the model changes, part of the agent's behavior does not change and some of its behavior changes.

Particle size, the original meaning refers to the size of the particles. In the simulation field, granularity generally refers to the model granularity of the simulation object. Model granularity is a quantity that describes the fineness of the simulation model. And Time Granularity (TG) is the size of a specific time span in the simulation process, and it is the minimum period for Agent to call. The coarse time granularity of machine learning means that the time span is large, that is, the time period of each simulation synchronization is longer. Simulation is concerned with different content. In the same scenario, the time granularity of machine learning may be different. Generally, coarse machine learning time granularity is used when caring about the abstract behavior of the simulation object, while the fine granularity is used when caring about the behavior details of the simulation object. It is necessary to choose the time granularity of machine learning for different requirements of simulation. Intuitively speaking, the finer the machine learning time granularity of simulation, the better, that is, the more detailed simulations, the better, but in actual simulation, the simulation under coarse machine learning time granularity may be better for various reasons.

(1) Simulation models are abstractions of actual objects, and simulation granularity should be consistent with model abstraction. If the abstract model of the simulated object does not support the fine machine learning time granularity, then the simulation under the fine machine learning time granularity is impossible;

(2) Generally speaking, the finer the time granularity of machine learning and the more simulation details, the more complex the simulation model and the more computing resources it needs to occupy. When the simulation model exceeds the limit that the computing resources can provide, it is necessary to modify the simulation model or increase the computing resources to ensure the operation of the simulation;

(3) The finer the granularity of the simulation machine learning time, the more calculations and communications, and the resulting errors may eventually accumulate to an unacceptable degree by the system;

(4) The finer the simulation granularity, the lower the simulation efficiency. The time granularity of the simulation machine learning cannot be determined too arbitrarily, otherwise the simulation goal may not be met, and the simulation results will lose meaning.

Appropriately determining the time granularity of the machine learning for simulation is an important issue for the simulation of complex systems, and if multiple machine learning time granularity simulations can be used, the simulation can be made more flexible. Generally speaking, for modeling a simulation object, its machine learning time granularity should be continuous, that is, under a specific set of continuous machine learning time granularity, the behavior and behavior selection rules are the same.

The simulation process generally starts from a certain moment, completes the simulation calculation within a certain simulation step, then synchronizes, and repeats this process. The simulation step is not the time actually calculated in the simulation process, but the logical time interval for the simulation object to execute one step in the application scenario when the simulation object is abstracted.

The simulation model, simulation step size and calculation amount of different machine learning time granularity are different. Generally speaking, the finer the machine learning time granularity, the more specific the simulation object model, the shorter the simulation step, and the greater the amount of calculation; the coarser the machine learning time granularity, the more abstract the simulation object model and the longer the simulation step.

When modeling a simulation object, it must be determined under what machine learning time granularity to simulate, so that its behavior model can be determined. The determination of the time granularity of different machine learning is mainly based on the different goals of the simulation. When you need to have a better grasp of the details, you choose a fine machine learning time granularity, and at the same time, the modeling of the simulation object will be finer, and the behavioral abstraction level will be lower; when you do not need to have a lot of details, if a large machine learning time granularity is used, the modeling of the simulation object will be coarser, and its behavioral abstraction level will be higher.

3.2. Agent model related to time granularity of machine learning

The significance of designing an agent model related to machine learning time granularity is that an agent can choose more than one machine learning time granularity model to run at a point in time, which is chosen by the program, platform or programmer. The most important determinant is observation. The degree of detail the readers pay attention to is different. In the same procedure, different observers at the same time choose different machine learning time granularity to observe. When observers focus on higher levels, they use coarse machine learning time granularity; when observers focus on lower levels, they use fine machine learning time granularity. The same Agent is more applicable to various simulation scenarios, whether it is a system running at a fine machine learning time granularity or a coarse machine learning time granularity, the Agent can be competent. When the hardware performance is limited and the performance is not enough for real-time simulation, coarse machine learning time granularity can be used to run. If the hardware performance is sufficient, fine machine learning time granularity can be used. This is a trade-off between performance and simulation goals.

In terms of implementation, the Agent model related to the time granularity of machine learning can be implemented by multiple Agent class files, or by one Agent class file. When multiple Agent class files are used for implementation, two Agent classes represent the same simulation object with different machine learning time granularity; when a single Agent class file is used, the same Agent can switch between different machines by changing different behaviors and state descriptions. This article focuses on the use of an Agent class to represent the model of different machine learning time granularities of simulation objects, and gives its switching method.

In the traditional model, the agent's behavior A does not change with the change of the simulated machine learning time granularity TG. In simulation, the behavior of an agent is generally designed for a certain machine learning time granularity TG. If the agent behavior is to be associated with the time granularity of machine learning, then an agent can be simulated under multiple machine learning time granularities TG, and different machine learning time granularities TG correspond to different Agent behavior A and behavior selection rules Φ. And under different machine learning time granularity TG, other objects in the environment may also be different. The actual application in the Agent is a range of machine learning time granularity, that is, behavior A applies to all machine learning time granularities between the machine learning time granularity TG1 and the machine learning time granularity TG2. In the agent model related to the time granularity of machine learning, the decision of the agent to take a certain action consists of the mapping from the input space I of the element to the output space O. The Agent model related to the time granularity of machine learning is shown in Fig. 3.

When modeling a simulation object with Agent, there is no need to restrict the machine learning time granularity of different models. However, in most cases, the machine learning time granularity of different Agent models of the same simulation object should be a multiple relationship. When machine learning time granularity is switched, compared with the agent of coarse machine learning time granularity model, the operation of fine machine learning time granularity agent generally has more details and the accuracy of the running result is higher than that of coarse machine learning time granularity agent. When the relationship is not multiples, the models of different machine learning time granularities will become difficult to compare and discuss. When modeling the Agent with different machine learning time granularity, if the behavior of the coarse machine learning time granularity model does not exist in the fine machine learning time granularity model, it should be combined by multiple behaviors of the fine machine learning time granularity.

The AUML interaction diagram of the agent related to the machine learning time granularity needs to add a mark of the machine learning time granularity to indicate the machine learning time granularity when sending a message. Since this

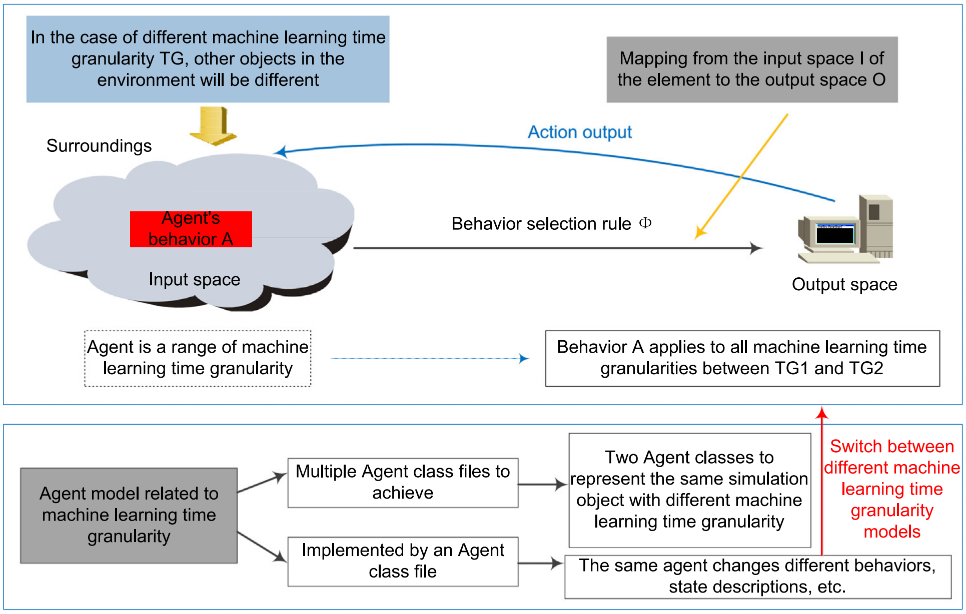


Fig. 3. Agent model related to the time granularity of machine learning.

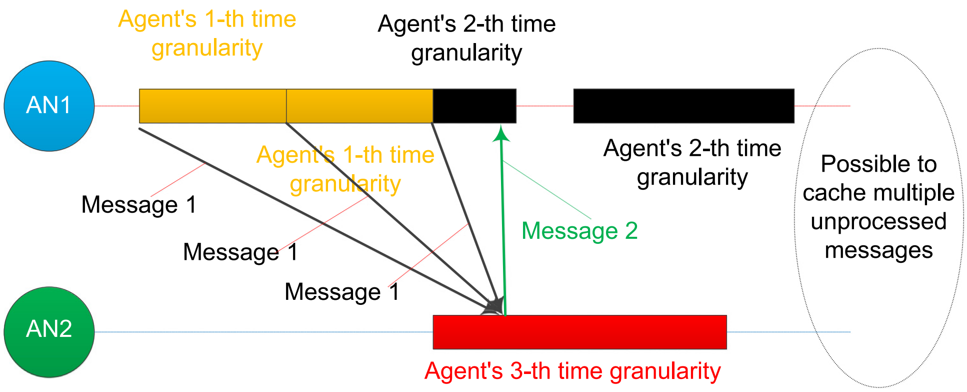


Fig. 4. AUML interaction diagram of the agent related to the time granularity of machine learning.

article discusses time-driven Agent simulation, when the Agent is called, it may cache multiple unprocessed messages, as shown in Fig. 4.

When the simulation is running, the Agent will switch the machine learning time granularity of the Agent itself due to different observed targets or performance problems to replace its runtime model. In model design, the model behavior of fine machine learning time granularity is more than the model behavior of coarse machine learning time granularity. The model of fine machine learning time granularity is based on the model of coarse machine learning time granularity, and the behavior is broken down more finely. The behavior result of a coarse machine learning time granularity is completed by multiple behaviors of fine machine learning time granularity.

4. Case study of online design of green urban garden landscape

4.1. Main scene sculpture design

Bionic design can be the source of inspiration and the deliberation of the concept. It is to process the characteristics of a certain creature in the biological world and apply it to garden landscape design. It gives people a sense of freshness and visual impact, but also gives visitors a sense of fun and returning to nature. The main scene sculpture of this project is designed with the concept of bionic design.

(a) Original posture of two peacocks (b) Feature extraction of linear elements (c) Modeling extraction

Fig. 5. The sculpture design of the main landscape of the green city garden.

The main scene sculpture is located in the cultural experience area of this project. This area focuses on recreation, sports, recreation, and cultural experience functions, combined with the planning and design of a certain area, to promote the culture of green urban garden landscape protection, and to establish a benign and orderly development model between the green urban garden landscape and the new district. In order to highlight this theme, the shape of the central square sculpture is inspired by the shape of a peacock flying. The linearity is extracted according to the form of the bird flying, and then it is simplified and beautiful according to the beauty and law of the form, and finally a green urban garden landscape sculpture theme is obtained. The sculpture design of the main scene of the green urban garden landscape is shown in Fig. 5.

Based on the machine learning Grasshopper parameter platform for the bionic design of bird wings, two points should be paid attention to. The first is dynamic characteristics. The deformation is carried out on the basis of the program logic set by the previous design purpose. For the bionics of wings, it is necessary to determine the range of mimicking motion. For example, in order to show the dynamics of the peacock, it is necessary to follow the action of spreading wings to imitate the dynamic logic of wings contraction. Second, it is texture characteristics. In addition to form, bionics also has textures or patterns. In order to imitate wings more realistically, the texture of wings is expressed linearly, that is, using multiple straight lines to splice, which not only gives people a visual impact, but also brings people a sense of reality.

(1) Design ideas

When using the machine learning Grasshopper parameter platform for bionic design, it is also divided into two steps: determining the design purpose and actual operation. The purpose of the design is also the design idea. When the machine learning Grasshopper builds imitating objects of the opposite sex, it should follow the idea of first overall and then partial. In order to increase the artistry of the sculpture, Mobius tape is used for design.

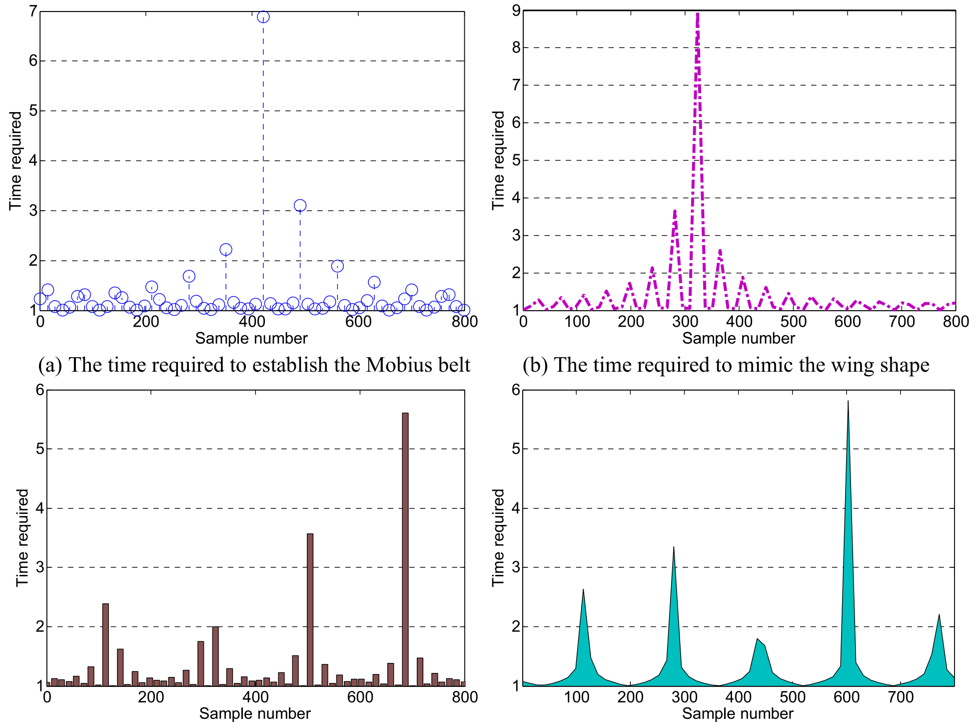
(2) Actual operation

The main scene sculpture introduces the Mobius belt, so we start with the Mobius belt. The first step is the establishment of the Mobius belt. We take the Mobius surface calculator from the lunch box collection and add the corresponding parameters to adjust the size later. The second step is to imitate the shape of the wings. The Mobius belt is a closed loop, and the wings are displayed in separate positions, so it is necessary to divide the Mobius belt into equal parts. Since the Mobius belt is circular and its starting point is at the top of the sculpture, the top of our sculpture needs to be separated. Therefore, the number of useful points needs to be picked up again. We use the list item calculator to extract the corresponding two long sides of the Mobius belt, and divide them into different numbers according to the law of the shape of the wings. The third step is to shape the wings. We connect all useful points in a logical order, and then use the Loft calculator to stake out. The fourth step is the "feather" construction. In the previously generated surface, we use isotrim and divide domain components for UV (horizontal and vertical) division. Since the division is carried out according to the quadrilateral, each grid has 4 sides. Our final goal is a shape formed by a one-way linear combination. Therefore, it is necessary to increase the data distribution step to go out a single straight line. The fifth step is shaping and adjustment. We use the cylinder command pipe to stake out all the line segments. Fig. 6 shows the time required for each step of the online garden sculpture design.

(3) Parameter adjustment of the shape

The parameter adjustment of the shape is mainly to adjust the various parameters of the model to make it produce various morphological changes for the designer's reference. Parameter adjustment can also be understood as the process of designers looking for suitable solutions.

There are not many adjustable parameters for the main scene sculpture, and the change in form is relatively small. Because before the modeling process, only one form of possibility was given, that is, the action when spreading wings. And this action is affected by the Mobius ring. Therefore, the adjustment parameters need to be based on the Mobius ring. The Mobius surface calculator has several inputs, namely U, V, R, T, and S. U and V are the divisions of the surface,



(a) The time required to establish the Mobius belt
(b) The time required to mimic the wing shape
(c) The time required to form wings
(d) The time required to build "feathers"

Fig. 6. The time required for each step of the online design of green urban garden landscape sculpture.

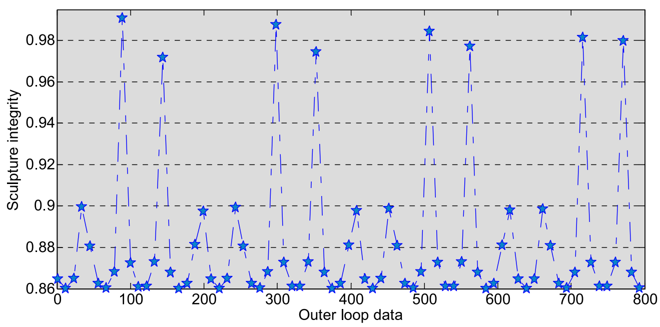


Fig. 7. Adjustment of green city garden landscape sculpture parameters.

representing the number of horizontal and vertical divisions. The default value in the calculator is 100, so it can be judged that the more split the surface, the smoother the surface. R represents the radius, T is the degree of twist, and S is the ratio of the outer ring to the inner ring. So here only the size of the model is changed, but the shape is not changed. So we adjust from the morphological parameters. The morphological parameter refers to the factors of the original state of the image model. The dynamic shape of the model is affected by the point bisect, so find the bisector calculator to adjust. But in the modeling, the Mobius ring was divided into two parts, so the halving point was divided into the outer ring and the inner ring. The adjustment results of the green city garden landscape sculpture parameters are shown in Fig. 7.

4.2. Plan and space of the main square

The users of the space are people, and different spaces bring different feelings to people, but different people will choose different spaces according to actual needs. Therefore, space and human beings are a pair of contradictory bodies, which have opposites of mutual choice and the commonality of mutual dependence. In addition, the space does not simply allow people to feel the narrowness in the enclosure and the wideness in the openness. The structural forms, patterns, colors, and textures of the materials in the space all give people different feelings. Therefore, the bionic of space is a synthesis of bionic design.

(a) Modeling extraction (b) Linear element feature extraction (c) Main square forming

Fig. 8. Design renderings of the main square.

The bionic design of the space needs a theme. We use abstract composition and logical structure to express according to the design theme. Therefore, the bionic design of space requires the combination of human and computer. It means that after people get inspiration from the natural world, they imitate and learn the generative logic of computers, and then deliberate and design the plan. There are four specific steps: the source of inspiration, the selection of computer logic, the deliberation of the plan, and the computer-generated plan. This article discusses the plan composition and spatial structure of the main landscape area in the landscape design of a certain urban green area as an example.

The first step is the source of inspiration. The sculpture in the center shows the theme of QiFei in the form of a peacock spreading its wings. The spatial layout is based on the ripples of peacocks flying on the water instantly. We take the amplitude of the ripple as the element of the enclosed space.

The second step is the choice of computer logic. The bionic design mentioned above should be a combination of natural and artistic features. The artistic features can be generated according to the digital logic of the computer. The gradient logic in the machine learning Grasshopper parameter platform is very popular in the design industry. It is the result produced by the pull point operator. Its logical principle is to influence the points around the object according to a certain gradual law, affecting the data between them. This kind of data is scalar, only size. So in Machine Learning Grasshopper, you can increase the vector direction to create more shocking laws. But not all bionic designs are realized through calculations. All designs realized through computers are dead and soulless, so we can learn from computer logic.

The third step is the deliberation of the plan. We imitate the logic of pull point to influence the hydrology in the theme concept. First we determine the influence point on the graph, and then according to the computer logic, the closer the influence point, the more drastic the change. In the process of scrutiny, it needs to give people a sense of flow. Therefore, in addition to logical imitation, artificial adjustment is also required. At the same time, the line segments cannot be stacked, so that others can see the clear structure lines and smooth composition. This is the effect produced by humans imitating digital logic.

In the fourth step, the computer generates the plan. That is, with the help of machine learning Grasshopper's logic, the opposite is considered and the plan is determined. The space of the facade is based on the layout pattern of low inside and high outside. The effect picture produced by using lumion is shown in Fig. 8.

4.3. Road curve design

The design of the main park road of this project does not form a circular lane like a general park, but is designed as a road system connected with the original sidewalk under the influence of the terrain and environment. The road design is mainly to consider the local area with larger space, and set up sightseeing battery cars, minivans and fire trucks. Therefore, the design speed of this project is 20–40 kilometers per hour. In order to make the design more scientific and reasonable, the road system of this project is constructed by machine learning Grasshopper, and when the turning position is designed, the parameters are restricted according to the requirements of the ''Road Route Design Specification''. This includes normative requirements for circular curves, superelevation of roads, superelevation easement sections, and widening easement sections.

For the road design steps of the machine learning Grasshopper parameter platform, it should conform to the above-mentioned machine learning Grasshopper design step principle of ''first overall and then partial, first restriction and then opening''. First, we determine the form and direction of the road. The form and direction of the road are generally determined from three aspects. The first is the centerline of the road, which determines the direction of the road. The second is the width of the road, which determines the grade of the road. For example, the width of the first-level road is more than 5 m. The third is the form of paving. The road centerline of this project is planned according to the design

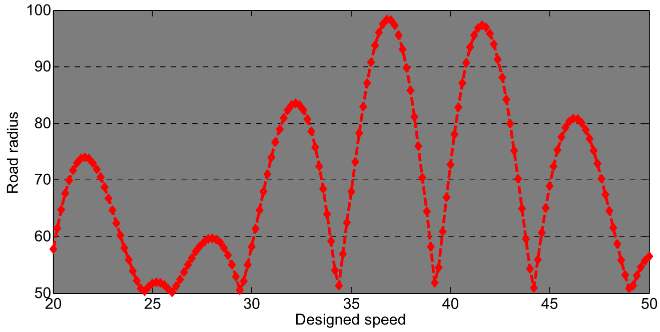


Fig. 9. The relationship between the radius of the circular curve of the green urban garden road and the design speed.

concept, and the plane data mainly comes from Auto CAD drawings. The width of the main road is designed to be 5 m in order to satisfy the entry of fire trucks. In order to reflect the natural ecology and embody the concept of "sponge city", the road pavement is paved with permeable concrete. The superelevation transition section refers to the smooth transition area from the road level to the superelevation cross slope of the road. The super-elevation transition section stipulates a gradient of 1:75. The most important factor is the radius, which is also limited by the speed of the vehicle and so on. The relationship between the radius of the circular curve of the green urban garden road and the design speed is shown in Fig. 9.

The design idea should first model the modeling, and then enter the parameter constraints. First, we sort out the design steps of the park road, and then add parameter restrictions in the middle. The steps are road center line, road width, radius, and design speed.

(1) We captured the middle line of the road in Rhino into Grasshopper for machine learning. In order to facilitate drawing, the road is generally drawn as a polyline in the process from manuscript to software assistance. This can determine the intersection of the road or the direction of the turn. (2) We use the offset calculator according to the required offset road width. (3) The basic form of the road has been built, and the visual parameter logic is constructed according to the core limiting factor "design speed". (4) According to the regulations, the speed of Gongyuanyuan Road is 4% for the super high section between 40 and 60, and 2% for the super high section between 20 and 40. According to the changing law of vehicle speed, it can be concluded that they conform to the changing law of arithmetic series with a tolerance of 2%. The super high road is to produce centrifugal phenomenon, so it is related to the radius. But the radius can be changed in the design according to the needs of the designer. In order to ensure that the specification requirements are met during the design process, two important limit operators will be introduced, namely Minimum and Maximum. Minimum means that the maximum cannot exceed a certain value, and Maximum means that the minimum cannot be lower than a certain value. If the design requires the road speed to be 40, no matter how small the turning radius is, the road superelevation cannot be higher than 2%. (5) According to the design requirements of vehicle speed, the turning radius is limited by parameter logic. The change rule of the minimum radius without superelevation is a proportional sequence. When the vehicle speed is 50, the limit of 400 can meet the design requirements.

5. Conclusion

Based on the discussion of the existing Agent model and supporting platform, this article points out its shortcomings in real-time simulation applications. The existing model is improved, the time granularity of machine learning is introduced, the related synchronization problems are studied, and the agent platform JADE is extended to provide Agents that support the time granularity of machine learning and support related time management mechanisms. On the basis of the Agent platform JADE, the Agent model related to the time granularity of machine learning is realized, the original Agent class is extended, the time granularity of machine learning and its binding and switching are realized, and the timetable in JADE is modified and improved. The corresponding time management mechanism in real-time simulation is introduced. Machine learning Grasshopper not only promotes designers' inspirational thinking, but also revolutionizes the garden landscape in terms of data logic and modeling of special-shaped surfaces. Machine learning Grasshopper has a logical relationship between "tree-shaped data" and "one-to-one correspondence" between data. Different data levels can produce different results and provide countless possibilities for designers' reference. A comparative analysis of machine learning Grasshopper and SU and other software, concluded that machine learning Grasshopper can replace all the functions of SU in garden landscape design, and can quickly and accurately generate special-shaped surfaces that SU cannot generate. We summarized the design steps of parametric design "whole to part, restricted to open". It makes Grasshopper machine learning different from traditional computer-aided software, and it is more unique, artistic and scientific.

CRediT authorship contribution statement

Jing Luo: Designed the research framework, Wrote the manuscript, Proofreading, Optimization of the results.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Belesky, P., 2020. A perennial practice: Designing between urban landscape and Urban Network. Archit. Des. 90 (3), 100–107.

Bird, R., 2020. Stalin's Well-Kept Garden: Horticulture, aesthetics, and soviet statecraft. Russ. Rev. 79 (3), 432–452.

D'Odorico, P., Besik, A., Wong, C.Y.S., et al., 2020. High-throughput drone-based remote sensing reliably tracks phenology in thousands of conifer seedlings. New Phytol. 226 (6), 1667–1681.

Jahani, A., Rayegani, B., 2020. Forest landscape visual quality evaluation using artificial intelligence techniques as a decision support system. Stoch. Environ. Res. Risk Assess. 34 (10), 1473–1486.

Oishi, Y., 2019. The influence of microclimate on bryophyte diversity in an urban Japanese garden landscape. Landsc. Ecol. Eng. 15 (2), 167–176.

Park, Y., Guldmann, J.M., 2019. Creating 3D city models with building footprints and LIDAR point cloud classification: A machine learning approach. Comput. Environ. Urban Syst. 75, 76–89.

Paynter, R., Ziegenbein, L., Lewis, Q., 2019. Excavating the Garden of the North: Five centuries of material and social change in western massachusetts: An introduction. Hist. Archaeol. 53 (2), 236–250.

Pérez-Rave, J.I., Correa-Morales, J.C., González-Echavarría, F., 2019. A machine learning approach to big data regression analysis of real estate prices for inferential and predictive purposes. J. Prop. Res. 36 (1), 59–96.

Reviglio, U., 2019. Serendipity as an emerging design principle of the infosphere: challenges and opportunities. Ethics Inform. Technol. 21 (2), 151–166.

Sweatt, A.J., Hedlin, H.K., Balasubramanian, V., et al., 2019. Discovery of distinct immune phenotypes using machine learning in pulmonary arterial hypertension. Circ. Res. 124 (6), 904–919.

Tang, L., 2019. Design of Natural Children's recreational spaces and facilities in Urban Neighborhood Parks. J. LandSc. Res. 11 (3), 1–4.

Tobey, M.B., Binder, R.B., Chang, S., et al., 2019. Urban systems design: A conceptual framework for planning smart communities. Smart Cities 2 (4), 522–537.

Ullah, K., Rashid, I., Afzal, H., et al., 2020. SS7 Vulnerabilities—A survey and implementation of machine learning vs rule based filtering for detection of SS7 network attacks. IEEE Commun. Surv. Tutor. 22 (2), 1337–1371.

Valdivia, A., Martínez-Cámara, E., Chaturvedi, I., et al., 2020. What do people think about this monument? Understanding negative reviews via deep learning, clustering and descriptive rules. J. Ambient Intell. Humaniz. Comput. 11 (1), 39–52.

Vartiainen, H., Tedre, M., Salonen, A., et al., 2020. Rematerialization of the virtual and its challenges for design and technology education. Tech. Ser.-Forsk. I Slöjdpedagogik Och Slöjdvetenskap 27 (1), 52–69.

Vij, A., Vijendra, S., Jain, A., et al., 2020. IoT and machine learning approaches for automation of Farm Irrigation System. Procedia Comput. Sci. 167, 1250–1257.

Yang, L., van Dam, K.H., Majumdar, A., et al., 2019. Integrated design of transport infrastructure and public spaces considering human behavior: A review of state-of-the-art methods and tools. Front. Archit. Res. 8 (4), 429–453.

Zhang, C., Atkinson, P.M., George, C., et al., 2020. Identifying and mapping individual plants in a highly diverse high-elevation ecosystem using UAV imagery and deep learning. ISPRS J. Photogramm. Remote Sens. 169, 280–291.

Zhang, C., Huang, Z., Cao, F., et al., 2019. Recognise me from outside to inside: Learning the influence chain of urban destination personalities. Tour. Manag. 70, 390–403.

Zimmerman, H.T., Land, S.M., Maggiore, C., et al., 2019. Supporting children's outdoor science learning with mobile computers: integrating learning on-the-move strategies with context-sensitive computing. Learn. Media Technol. 44 (4), 457–472.