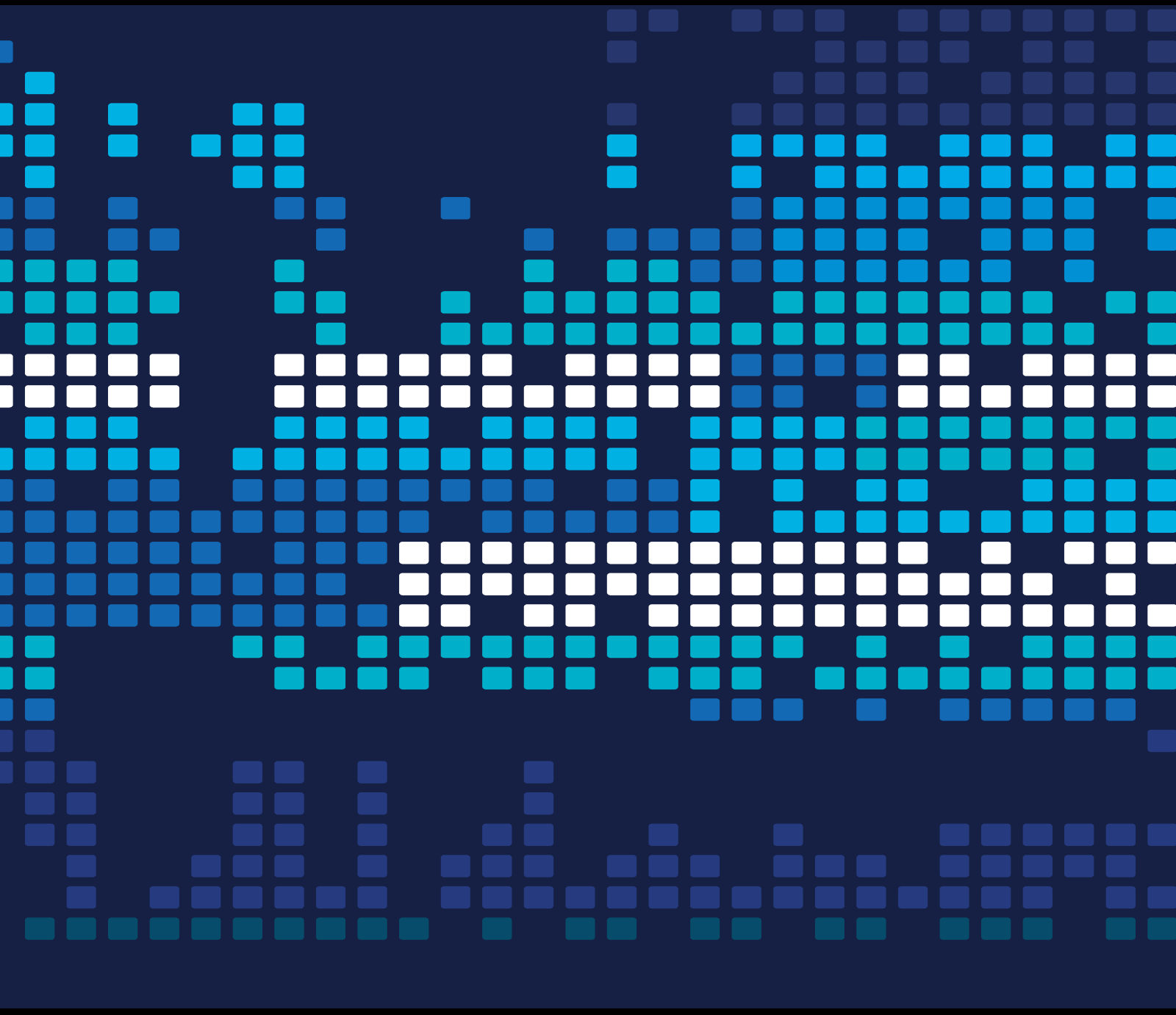
Scientific Programming

Service Automation with Data-driven Decision Analytics

Lead Guest Editor: C.H. WU
Guest Editors: Daniel Mo, Wai Hung Ip, and Yung Po Tsang



Service Automation with Data-driven Decision Analytics

Scientific Programming

Service Automation with Data-driven Decision Analytics

Lead Guest Editor: C.H. WU
Guest Editors: Daniel Mo, Wai Hung Ip, and Yung Po Tsang

Contents

Hindawi
Scientific Programming
Volume 2022, Article ID 1557302, 19 pages
https://doi.org/10.1155/2022/1557302

Research Article

Automatic Generation of Real-Time Animation Game Learning Levels Based on Artificial Intelligence Assistant

Rong Zhang

School of Artificial Intelligence, Dongguan Polytechnic, Dongguan, China

Correspondence should be addressed to Rong Zhang; 443798430@qq.com

Received 27 July 2022; Revised 7 September 2022; Accepted 19 September 2022; Published 13 October 2022

Academic Editor: Y. P. Tsang

With the development of computer animation games, now many animation games use the automatic generation of animation game levels method to generate animation game content. Using the automatic generation levels of the animation game method can quickly generate many different levels, saving labor costs in manual production. And with the combination of other fields, more accurate results can be obtained. In this paper, the goal is to automatically generate the level of the animation game. This research proposed animation game objects and victory conditions to derive the concept of the clearance path and uses artificial intelligence genetic algorithms to obtain the clearance path that best meets the needs of the developer to recombine new levels. In addition, since the generated levels are regenerated from the clearance path, there are still too many blank areas. In order to make full use of these blank areas, this research provides a level complete system. The system has formulated its placement rules for existing animation game objects. These rules fill in these blank areas and produce a more complete level. The contributions of this search proved that the genetic algorithm can converge quickly due to the correction effect in the mutation. In the complete system, it has successfully achieved a complete and playable level. In the system running time, this research adjusted the method of selecting elites and the use of permutations and combinations to mate, so that the system's operation can be stabilized on an average value.

1. Introduction

With the development of advanced animation games, more and more animation games also use ALG methods to generate animation game levels. Through ALG, different scene distributions can be easily explored. After the corresponding screening, a large number of qualified levels can be easily obtained, which greatly reduces the labor cost of manual production of levels and saves the storage capacity required for animation games. In Algwiki, this research has compiled a list of animation games using ALG technology, which contains many different types of animation games. There are single-player animation games and even multiplayer animation games. It can be seen that the development of ALG has gradually matured. And it also contains some animation game masterpieces that also use this technology. Therefore, it is also hoped that ALG technology will be introduced into the animation game, through the automatic generation of levels to generate many different levels, and to further complete the generated levels, so that the level can be improved and reduce the excessive probability of simple or unplayable levels. The article mainly consists of three parts: (1) create an animation game platform that can be loaded into a custom level for players to play, (2) research and develop an automatically generated level suitable for animation games and reduce the labor cost required for manual production of levels, and (3) provide a level complete system to correct oversimple or unplayable levels, and through corrections and adjustments, more different combinations of levels can be generated [1–3].

The goal of this research is to automatically generate the level of the animation game. First, we set the name and shape of the animation game object in the configuration file. The shape is represented by a 2D matrix, and then, we set the probability table to define the animation game object in the level. The position of the appear in the setting file is set in the

configuration file, which will be divided into three parts: upper, middle, and lower. After the setting is completed, the system will randomly generate a set of initial levels. This research uses the concept of clearance path. Applying this concept to the genetic algorithm, the system will obtain the clearance path data of these initial levels and pass the level. Crossover and mutation of the path generate multiple sets of new clearance path combinations. These new clearance paths can be reconstituted into new levels [4–6]. Finally, these new levels will be evaluated to obtain the corresponding scores, and the system will take them out according to the relevant settings in the configuration file. The corresponding number of levels will be included in the mating pool to participate in the next generation. Since the levels generated by the genetic algorithm have undergone many mating and mutations, the playability and completeness of the levels are not reliable. Therefore, this research provides a complete level system. After this system, this research aims at the level. The Rule-Base has been revised and adjusted, and multiple sets of different possible correction levels have been generated to obtain a more complete level [7–9].

The architecture of the whole system mainly consists of four parts. (1) Web server provides an API for better communication on the client side and the animation game database. (2) Animation game database mainly stores player information, models used by the level, and level information. (3) Client-side provides an animation game of unity of player version. (4) Generator provides automatic level generation and level integration system. As a contribution to the platform of the animation game, this research has implemented an animation game platform that can load custom levels for players to play and automatic levels generation based on AI genetic algorithms. Secondly, this research also proposed a level complete system to modify and adjust the generated levels. According to the experimental results, the completed level has a significant difference in the completeness of the scene compared with the incomplete level. Completeness still guarantees playability. Through the complete system, more different levels can be generated after the completion, so that developers can have more choices, also increase the system's exploration of levels, and increase the probability of the appearance of levels that meet the designer's needs [10–13]. The novel contributions of this search proved that the genetic algorithm can converge quickly due to the correction effect in the mutation. In the complete system, it has successfully achieved a complete and playable level. In the system running time, this research adjusted the method of selecting elites and the use of permutations and combinations to mate, so that the system's operation can be stabilized on an average value [14, 15].

2. Related Work

With the development of computer animation games, and in order to save labor costs, more and more animation games adopt ALG methods to generate animation game content, such as some classic animation games, elf, Angry Birds, and Diablo [16–18]. Therefore, this research also hopes to introduce ALG technology into the animation game.

2.1. Hyperautomation. Hyperautomation is a true digital transformation with the help of advanced techniques such as Robotic Process Automation (RPA), Machine Learning (ML), and Artificial Intelligence (AI). It automates complicated business processes, even where topic specialists were formerly needed. This is an expansion of the processes of traditional business process automation. Hyperautomation allows automation to do virtual tasks performed by business people by merging AI technologies with RPA. This takes us to the next level for detecting and generating automation processes dynamically. It allows companies to combine business intelligence systems, undertake complex needs, and increase human expertise and automation experience. This paper briefly discusses hyperautomation and its need in the current scenario. Then, it elaborates the significant roles of sensors to enhance hyperautomation. Various versatile technologies, such as dedicated workflow processes and specific domains of solicitations associated with hyperautomation, are also discussed diagrammatically. Then, this study further identifies and discusses the capabilities of hyperautomation for industries. Hyperautomation is being utilized to increase the efficiency and human enhancement of automated operations substantially. It comprises several automated tools, including analytics, discovery, design, measurement, monitoring, and complex automation components. Thus, it ideally utilizes to integrate state-of-the-art tools and develop new methods of working [19–21].

Robotic Process Automation (RPA) deals with the automation of rule-based process tasks to increase process efficiency and reduce process costs. Due to the utmost importance of business process automation in industry, RPA attracts increasing attention in the scientific field as well. This paper presents the state of the art in the RPA field by means of a Systematic Mapping Study (SMS). In this SMS, 63 publications are identified and analyzed. From the SMS findings, a framework for systematically analyzing and comparing RPA works is derived. The discovered thematic clusters suggest further investigations to develop a more elaborated structural research approach for RPA [22–24].

The vision of factories of the future is motivating many industrial companies to modernize their existing portfolio of systems and services to maintain market share and improve business agility. For long-lived industrial systems, it is challenging to adapt legacy assets to a service-oriented stream in cloud computing and Internet-of-things contexts. For this reason, many research studies have proposed techniques and methodologies to migrate legacy industrial functions and systems at different hierarchy levels of automation control. This paper presents an overview of these techniques and methodologies, as well as industry practices to achieve the vision of factories of the future. A better understanding of the challenges encountered in legacy migration processes will help researchers and practitioners in their further efforts [25].

2.2. Animation Game. The background of the animation game came to the world and learned that a few behaved apples lost. He had to send the lost apple home, but he would

encounter them on the way. When you go to some terrible institutions is different from the traditional puzzle-solving animation game, which only controls a single object to perform the animation game, such as traditional classic puzzle-solving animation games such as warehouse fan, push music, Mario and Tetris, and a simultaneously an animation game that controls two objects to complete the animation game level and combines physics to make collision behaviors with other objects. In the animation game, players use both hands to control two s at the same time and transport the apple back to the bag [26–29]. Completing the animation game level requires not only good control ability but also excellent hand-eye coordination [30–32].

Players operate two s in different directions to move in two directions (left and right). In the level, by controlling the transported to the apple bag, then you can enter the next level. The player can use the keyboard keys *Z*, *X* and *N*, *M* to control two s at the same time. There will be some obstacles in the process of apple to prevent the player from delivering the apple. These obstacles are as follows:

Gap: it may cause the apple to fall on a more difficult route, and the player must accelerate across the gap or may need to swap the positions of the apple to reach the gap smoothly.

Trap ball: it may swing around the player's path. When the apple hits the trap ball, it will be knocked away. The player must calculate the swing time of the trap ball or use collide the trap ball multiple times to get the trap ball [33].

2.3. Automatic Level Generation (ALG). This article mainly uses genetic algorithms to automatically generate animation game levels. The following are related documents for automatically generating animation game levels: the method proposed by Ferreira (Ferreira and Toledo, 2014) uses genetic algorithms to generate Angry Birds levels, treats all elements in the Angry Birds levels as an object, introduces the object into the concept of stacking, and proposes a fitness function to evaluate the balance and quantity of the level elements. This method can generate levels in real time, and the generated levels can be balanced without collapsing. The method proposed by Snodgrass cuts the map of Super Mario Bros into H∗W Tiles, given some existing Mario Maps, learns the occurrence probability of elements in the map through Markov Chains, and finally uses these chances to generate new Mario Maps. The method proposed by Dahlskog divides the map of Super Mario Bros into three different levels of patterns, namely, Micropatterns, Mesopatterns, and Macropatterns. Mesopatterns are composed of Micropatterns, and Macropatterns are composed of multiple Mesopatterns. The Mesopatterns have different classifications, including Enemies, Gaps, Valleys, Multiple Paths, and Stairs. Finally, through Evolution Algorithm, Micropatterns are combined to generate new Mario Maps, and meaningful combination levels are left through the fitness function. The method proposed by Horn and Dahlskog uses a variety of related Mario AIs that have been published and sets different indicators to obtain the different characteristics of these AIs. Finally, these characteristics can be used to generate levels. It is also possible to combine multiple different AI features to obtain a better level or meet the designer's expectations. This research refers to the method proposed by Ferreira. The level is cut into a 32∗24 Grid, and the concept of the customs clearance path is derived based on the relationship between the elements and the customs clearance conditions. The concept is brought into the genetic algorithm and then applied to the automatic generation of animation game levels. Different levels can be obtained by controlling the system parameters and the probability table of the relevant element data and the element placement position for a given level.

Also, levels are generated using 16 patterns in the system to combine multiple levels with different configurations. The difference from this is that this system only needs to set the system parameters and the animation game object and the animation game object placement probability table to generate the level. Its ALG has been used in many different types of animation games. In the article "The Death of the Level Designer," the ALG used in animation games is subdivided into the following categories: runtime random level generation, design of level content, dynamic world generation, instancing of in-game entities, user mediated content, dynamic systems, procedural puzzles, and plot generation from the above classification; it can be seen that ALG has been flexibly used in animation games, such as animation game content, maps, and animation game plots. In addition, Togelius also mentioned that the production methods of ALG can be divided into Online or Offline, Necessary or Optional Content, Random Seeds or Parameter Vectors, Stochastic or Deterministic Generation, and Constructive or Generate-and-Test. ALG is also integrated with other fields and therefore has more room for development and applications, such as Learning-Based and Search-Based. The following will explain the related documents of automatic level generation and genetic algorithm, respectively [34, 35].

2.4. Genetic Algorithm (GA). A genetic algorithm (Genetic algorithm, 2017) is developed based on the evolutionary phenomenon of the biological world, which includes heredity, mutation, natural selection, mating, etc. This research uses computer simulations to implement this algorithm. This research treats parameters as genes, allows these genes to evolve, and then explores and obtains parameters that are more suitable for solving the problem. Genetic algorithms are good at solving global optimization problems. For example, to solve the problem of scheduling timetables. Many types of software for scheduling timetables use genetic algorithms. Nowadays, genetic algorithms are also used in many different fields, such as circuit design and neural networks [36, 37] for manufacturing systems/processes [38, 39]. This paper derives the concept of the customs clearance path based on the animation game clearance conditions, uses genetic algorithms to generate better clearance paths, evaluates these clearance paths through the evaluate function, and retains the levels that meet the expectations. As the generations evolve, we will get the most consistent level of evaluation [40–43].

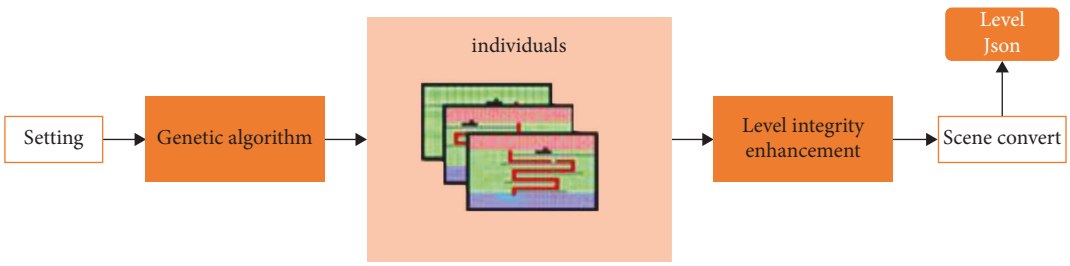


Figure 1: The framework of our proposed method.

Table 1: System basic parameter setting.

Main Main setting	width	Grid preset width
	Height	Grid preset height
	Top	Defines the top area
	Bottom	Defines the bottom area
	init_gen_num	Initialize the number of genes
	Population_Elitism	Parent selected number
	Elitism	Mating pool elite selection (%)
Ground initialize ground-related settings	min	Ground minimum length
	max	Ground maximum length
	Count	Ground out of the count initialization level quantity (min, max)
Trap Trap ball-related settings	max_count	Maximum number of trap spheres
Export output conversion-related settings	save_path	Output file placement location
	background_res	Unity_prefab related path settings
	Prefab_psth	
	_Res	
	floors_res	
	apple_res	
	bag_res	
	trap_res	

3. Research Methods

This article mainly uses genetic algorithms to generate animation game levels; at the same time, a level complete system is made to reduce the blank area in the level, so that the level can be closer to a complete level, as shown in Figure 1. The method can be divided into two parts: (1) genetic algorithm and (2) level integrity enhancement.

3.1. Genetic Algorithm. This article is inspired by Ferreira, referring to its system architecture combined with genetic algorithms to generate animation game levels, and uses the path of the level as the main axis of the level to ensure that the level can be cleared. This section disassembles the genetic algorithm into multiple small sections to explain it one by one [44–47].

3.1.1. Scenario Structure. In this article, referring to the concept of level division in Ferreira (Ferreira and Toledo, 2014), the level scene is vertically divided into three parts, the top, the middle, and the bottom. This research regards the level as a grid, and this grid has a total of 32∗24 Cells, and its animation game screen resolution is preset to 1024∗768; that is, each cell occupies 32∗32 animation game screens. *Pixel.* The entire level is divided into three layers. From top to ground, there are three layers: top (red area), middle (green area), and ground (blue area); top and ground are set by parameters in the configuration file, and the remaining areas belong to the middle.

3.1.2. System Parameter Setting. Before the system starts to operate, some necessary parameters must be set so that the system can execute smoothly and obtain results that meet the expectations of users. There are 3 configuration files in the system. Table 1 is the basic parameter settings of the system. The parameters set here are the system defaults.

Since the system uses the grid to represent levels, it needs to put the objects in the animation game level as grid representation as Figure 2 and separately describe the cell occupancy pattern of its animation game object in the grid.

The representation method is shown in Table 2. There are 2 types of representation methods for the area description. (1) The area description is N or a number, which means the X-axis length of the object is N. (2) Take the object as an example; the area description is 4∗2, which means the X-axis length is 4, and the Y-axis length is 2 for the object. If the area is described in the above-mentioned type 2 notation, the pattern must be defined separately. In the pattern, 0 represents a blank area, 1 represents the physical area of the

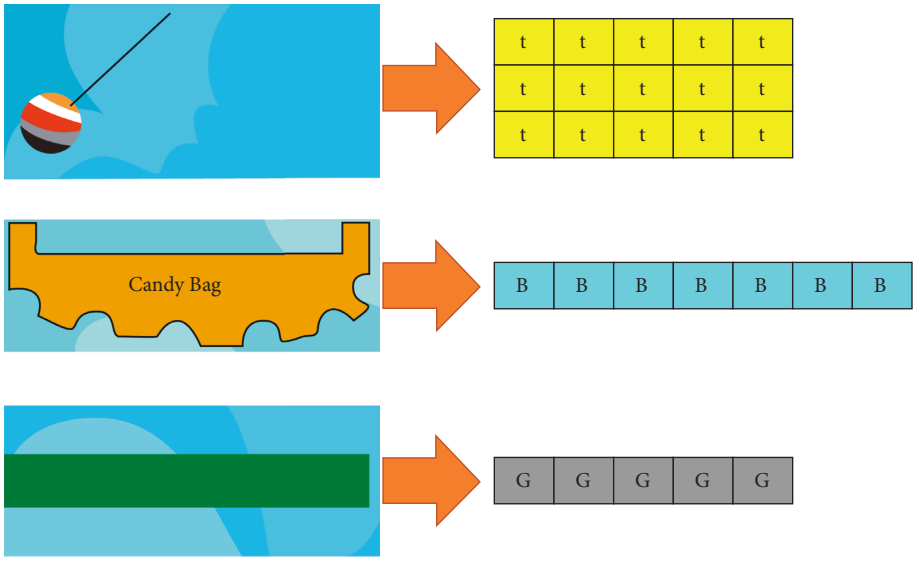


Figure 2: Animation game object pattern.

Table 2: Animation game object setting table.

Ground	G	N	1
Candy	C	1	1
Road	$p1$	$4 * 2$	[0, 0, 1, 0], [1, 1, 2, 1]
Trap	t	$5 * 3$	[1, 1, 1, 1, 1], [1, 1, 1, 2, 1], [1, 1, 1, 1, 1]
Bag	B	7	1
Empty	E	1	1

object, and 2 represents the reference point for placing the object when the system determines that an element is placed at a certain coordinate, its coordinate is the reference point of the element, and the Y-axis separation is indicated by a semicolon.

3.1.3. Probability Table Setting. In the initial population phase of the genetic algorithm, the system will refer to the probability set here to place the corresponding animation game object. Here, the probability normalize is a value of 0~1 (0%~100%), and the sum of the three floors' probability of a single object must be equal to 1.

3.1.4. Initialization. Since when the system is first executed, there are no individuals in the mating pool that can mate with each other, so it is necessary to initialize some available individuals to allow the system to execute smoothly. First, obtain the configuration information of each element through the probability table and randomly place it at a corresponding position in a random number manner to generate N individuals, where N refers to the init_gen_num parameter in the system setting, and the rules of ground will also be based on the system setting. The parameters are set. Since the animation game uses gravity physics, the individuals generated will first go through the gravity function. This function will give all elements except the floor ground the gravity effect so that the individual elements are placed in a position that conforms to the gravity effect. After the gravity function, a clearance path search will be performed to obtain the clearance path and apple information of the individuals. If the clearance path or apple information cannot be successfully obtained, the level will be determined as unavailable. Regenerate a new individual until the value of the init_gen_num parameter in the system setting is satisfied.

3.1.5. Clearance Path Search. The clearance path search function is mainly used to obtain the clearance path and apple information. First, the system will first obtain the bag element, take the cell occupied by the element as input, and perform reverse path tracking one by one. The system coordinate system here is shown in Figure 3. Taking a 32∗24 Grid as an example, the upper left cell coordinate is (0, 0), and the lower right cell coordinate is (31, 23).

The tracking method of the clearance path mainly uses two methods.

Method 1 is responsible for the Y-axis search. Each time it is executed, it will perform an upward search action (Method 1, line 5) until it encounters the ground (G) element or exceeds the coordinates outside the scene end.

Method 1: Scan Y

(1) *cell* ← [*X*, *Y*];

(2) while *cell.Y* ! = "*G*" *and cell.Y in Scene* do

(3) if (*cell*.LowerRight = "*G*") ScanX(*cell*, *Right*);

(4) else if (*cell*.LowerLeft = "*G*") ScanX(*cell*, *Left*);

(5) *cell.Y* ← *cell.Y*—*1*;

(6) end while

Method 2 is responsible for the X-axis search. It will determine the calculation of X according to the direction parameter (Method 2, line 2) every time it is executed and search to the left or right until there is no ground (G) element support below the position or out of the scene The coordinates are over.

Method 2: Scan X

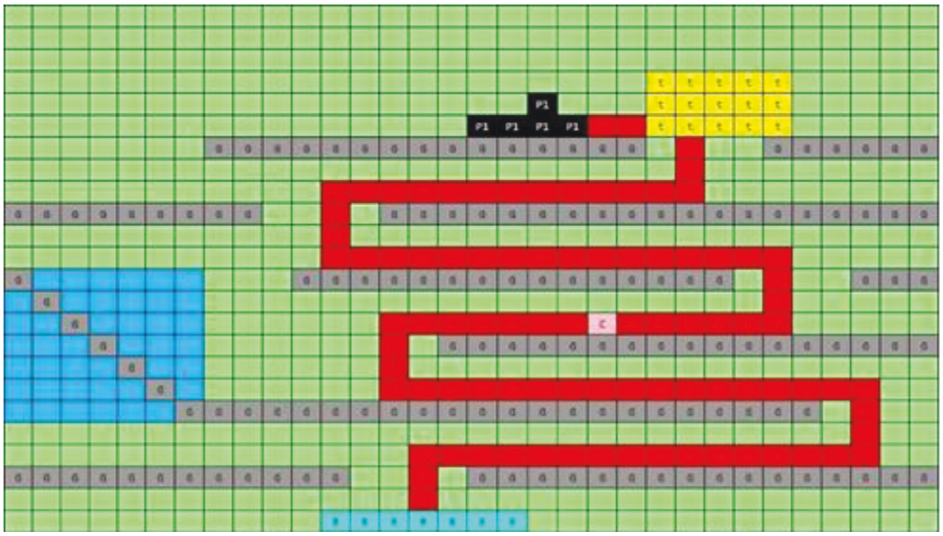


Figure 3: The result of placing the trap ball.

(1) *cell* ← [*X*, *Y*];
(2) *direction* ← (Left or Right)
(3) while *cell.down* = *"G" and cell.X in Scene* do
(4) ScanY(*cell*);
(5) if (*direction* = *Left*) *cell.X* ← *cell.X*—*1*;
(6) else if (*direction* = *Right*) *cell.X* ← *cell.X* + *1*;
(7) end while

After the above two methods, the blue path will be judged by the system as the customs clearance path.

3.1.6. Evaluation. The fitness function is used to find the optimal path from the source node to the destination node to reduce the energy consumption in multipath routing. The generated levels will be scored through the evaluate function (1). Here, the information of the pass path (clearance path) and apple obtained after the clearance path search is used.

Equation (1) is the fitness function (Ttito and Huacho, 2021):

$$f_{\text{in d}} = \frac{f_1(f_2 + f_3)}{2}. \tag{1}$$

Equation (2) ensures that apple exists:

$$f_1 = \begin{cases} 1, & [\text{apple}] \neq NULL \\ 0, & \text{Otherwise} \end{cases}. \tag{2}$$

Use (2) to ensure the existence of the apple. If the apple information is empty, the score will be 0.

$$f_2 = \frac{\text{Count}(\text{PassPath}, \text{CheckPoint})}{\max(\text{PassPath}, \text{CheckPoint})}\left(1 - \frac{|\text{Left.count} - \text{Right.Count}|}{\text{Count}(\text{PassPath}, \text{CheckPoint})}\right). \tag{3}$$

Equation (3) is the clearance path complexity.

In (3), this research uses the maximum Check Point of the level (equation (4)) and the number of Check Points currently owned by the clearance path as weights. The maximum Check Point uses the scene height to accommodate several s as the value, and the height of the element is increased by 1. It is due to the placement of the floor elements, and the final maximum value minus one is to deduct the reserved layer of the bottom bag.

Equation (4) is the Check Point Maximum:

$$\max(\text{PassPath}, \text{CheckPoint}) = \frac{\text{Scene.Height}}{\text{Monkey.Height} + 1} - 1. \tag{4}$$

Secondly, use the Check Point with the left direction and the Check Point with the right direction to calculate, so as to obtain the average result of the left and right times of the Check Point; that is, the left and right changes are more, which can increase the complexity of the customs clearance path.

$$f_3 = \frac{\text{PassPath}_{\text{countmovement}}}{\text{PassPath}_{\text{totallength}}}. \tag{5}$$

Equation (5) obtains the percentage of the total length of the clearance path and the number of steps that can be moved as the scoring benchmark. The number of steps that can be moved is defined by the support of the floor element under this coordinate. The higher the score, the longer the

length and the movable step. The number is large, so as to obtain a longer customs clearance path.

3.1.7. Selection of Mating Group. Here, the mating chromosomes will be selected for genetic evolution. After evaluation, all individuals will have a score, which will be ranked from high to low according to the score, starting from the highest score, and M individual individuals will be taken out, where M is the Population Elitism parameter in the system setting and will be selected. The individual will be put into the mating pool and used by crossover and mutation later.

3.1.8. Mating. The individual system selected by a select population takes its clearance path as the chromosome. This research uses the permutation formula P_n, where n is the population Elitism parameter in the system setting. Arrange every 2 clearance paths in the list as a group and each group as a uniform crossover. Take an example, the two sets of chromosomes have different lengths, and both have the first three sets of genes, then the first three sets of genes have a 50% chance to exchange each other, and the first set of chromosomes lacks the fourth and fifth sets of genes. There is a 50% probability that the second set of genes will be copied. The example is the result of copying the last set of genes. Since there may be genes that can be merged after the crossover, it uses the merge method to integrate the genes. Crossover here uses a small number of high-scoring maternal populations to exchange their gene combinations to generate chromosomes with a larger number of genes or different gene combinations. Secondly, the number of genes can be increased to increase the probability of a better combination.

3.1.9. Merge Method. After the crossover, the gene system will be in the chromosome, and the two adjacent genes will be checked one by one in a group. The judgment conditions are as follows:

(1) The same direction (←, ← OR ⟶, ⟶ OR ↑↑) will synthesize a group of genes
(2) The opposite direction (←,⟶ OR ⟶,←) values cancel each other and finally combine into a group of genes
(3) If the directions are not the same and one group of genes is "↑", the next group will be judged directly

3.1.10. Mutation. In the crossover part, it is mainly by exchanging genes to generate more possible combinations and chromosomes with longer gene lengths. Make rational corrections and probability mutations for each group of genes to obtain more different possibilities, so as to explore possible better results. The mutation of this system is mainly divided into four parts. Among them, direction mutation and Check Point mutation are probability mutations, and fixed length and fixed path are rationalization correction directions,

3.1.11. Direction and Check Point Mutation. The first part of gene mutation is the chance of changing the direction of the gene. The mutation probability of each group of genes is 50%. The system will replace the originally left gene with the right gene with a 50% probability. If it is an original right direction gene, it will be replaced with the left direction, and if it is an upward direction gene, the system will obtain the remaining blank area above the scene to randomly change the upward direction value but will not change the upward direction.

The second part of the mutation is for genes in the upward direction. The system will base on the height of the (player) element. If a group of genes with an upward direction is found to be higher than the height of the element, the system will respond with a 50% probability. Splitting the upward gene will split the upward gene into two groups of upward genes; inserting a set of random left or right genes between the two groups of upward genes, this action will increase the coloring of the number of genes in the body used to increase the complexity of the customs clearance path. An example of gene splitting: the genes marked in red font on the left are split into 3 genes on the right.

3.1.12. Correction Function. Here, the system will perform correction actions for each group of genes. The main purpose is to make more combinations that were previously unavailable into useable combinations and to explore better combinations through corrections. The system will make correction judgments for each group of genes one by one. The correction is divided into two parts; the first part is the length correction. The system will adjust the genes that do not meet the basis value based on the width and height of the (player) element. The gene of the basis is given a value that satisfies the basis again. The value basis for the left and right directions is 6, and the value for the upper direction is 3. If the value in the left and right direction is less than 6 and the value in the up direction is less than 3, a new value will be assigned.

The second part is the direction modification. After crossover and mutation, this research found that many levels have undergone the step of modifying the gene length. Although the goal of increasing the length has been achieved, the path of clearance may go beyond. Therefore, the system calculates according to each group of genes and at the same time checks whether the calculated result is out of the scene; if it is out of the scene, replace it with the gene in the opposite direction as a correction.

3.1.13. Selection of Elites. After crossover and mutation, the system ranks the scores of all individuals from high to low. Refer to the Elitism score threshold parameter in the system setting to calculate P_{score}. The formula is as follows.

Equation (6) is the Elitism score threshold calculation formula:

$$P_{\text{score}} = \max\,(\text{mathingPoolScore}) - \frac{\max\,(\text{maxtingPoolScore}) * \text{Elitism\_score\_threshold}}{100}. \tag{6}$$

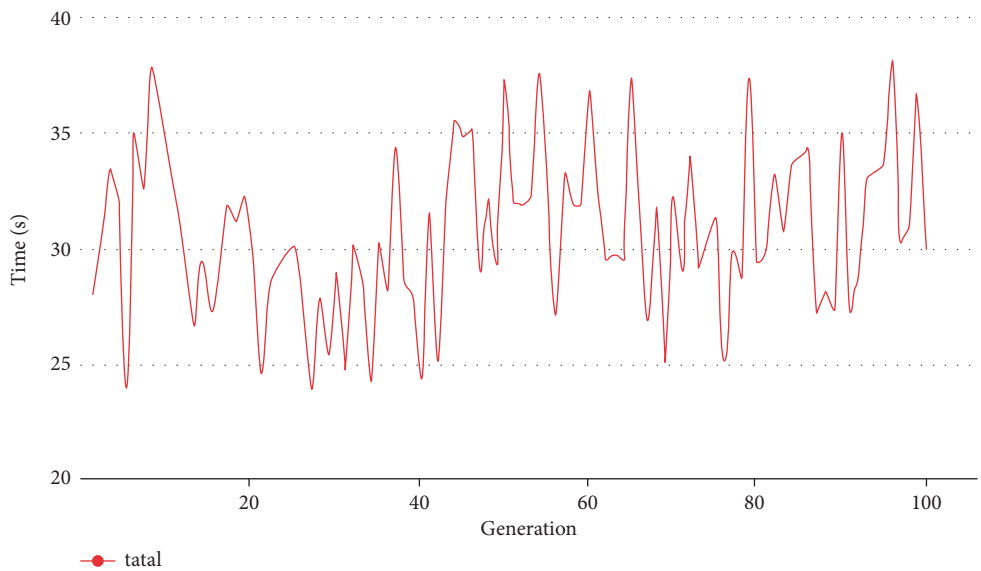


Figure 4: Line chart of the total time spent in the execution of the 100-generation system.

If the current level score is higher than P_{score}, the level will be transferred to the mating pool to participate in the next generation; otherwise, the level will be discarded after all actions of the generation are over.

3.2. Level Integrity Enhancement. After the genetic algorithm, it can obtain multiple sets of levels, and these levels all have a path to pass the level in Figure 4 (red path) that can guarantee its playability, but this research found that there are still many unused levels in these levels. The entire level does not look complete enough. Therefore, it has developed a system to strengthen the integrity of the level and set a number of rules to fill in the blank areas in the level and increase the diversity of the level.

3.2.1. Completion of Customs Clearance Path. The clearance path is the main reference core. The purpose of this research is to ensure that the clearance path is unobstructed, which means that the level is playable. The level generated by the genetic algorithm level generation system is based on the relative position of the clearance path endpoint and determines the generation of the top floor, so the system first connects to the end of the clearance path. It needs to define a complete clearance path. The clearance path is used to ensure level playability, and the clearance path is blocked or covered arbitrarily. Such actions mean the need to bear the risk of reduced playability.

3.2.2. Floor Filling. After completing the clearance path, the system will then fill in the blank area in the level. First, you need to analyze the level and divide the floors in Figure 5. If ground appears on the Y-axis of the scene, the system will regard it as a floor, and the floor where the element is located is the top floor, and the bottom floor is the bottom floor. The top and ground floors are called for the middle layer.

After the floor is cut, the system will be divided into two parts for processing. The top part of the system will detect the direction of the path through the customs and open a hole on the other side of its direction. This action can increase different path lines so that this level has more than a single path to choose from at the beginning of the game. For the middle layer and the bottom layer, each layer will fill its ground but retain its clearance path.

3.2.3. Ensuring Customs Clearance Path. After the system fills up each floor, although the clearance path is still maintained, the gap must be greater than the width of the clearance path to be truly effective. Therefore, the system will check the width of the gap in the clearance path here to ensure that the clearance path can really let the element pass. First, the system will first obtain all the coordinates on the path of the clearance, and each coordinate will return a set of 9 grids pattern; this research does individual processing for 3 different patterns.

The center point of all patterns is the detection point. If the pattern belongs to Figure 6(a), the system will convert the 2 cells on the left and the 2 cells on the right of the detection point into blank blocks to ensure that the element can be smoothly followed. The clearance path drops down. In Figure 6(b), the system converts 1 cell on the left and 2 cells on the right of the detection point into blank blocks.

3.2.4. Placing a Sloped Floor. In order to increase the diversity of the animation game, this research added a sloped floor, and because it has a gravity physics system, adding a sloped floor can also relatively improve the animation gameplay and difficulty. In order to avoid the problem of the joints of the diagonal floor and whether they can go up to the slope, the sloped floor is divided into two types: single-joint slope and double-joint slope. The single-joint slope in Figure 6(a) is classified in this research. It is defined as a 30-

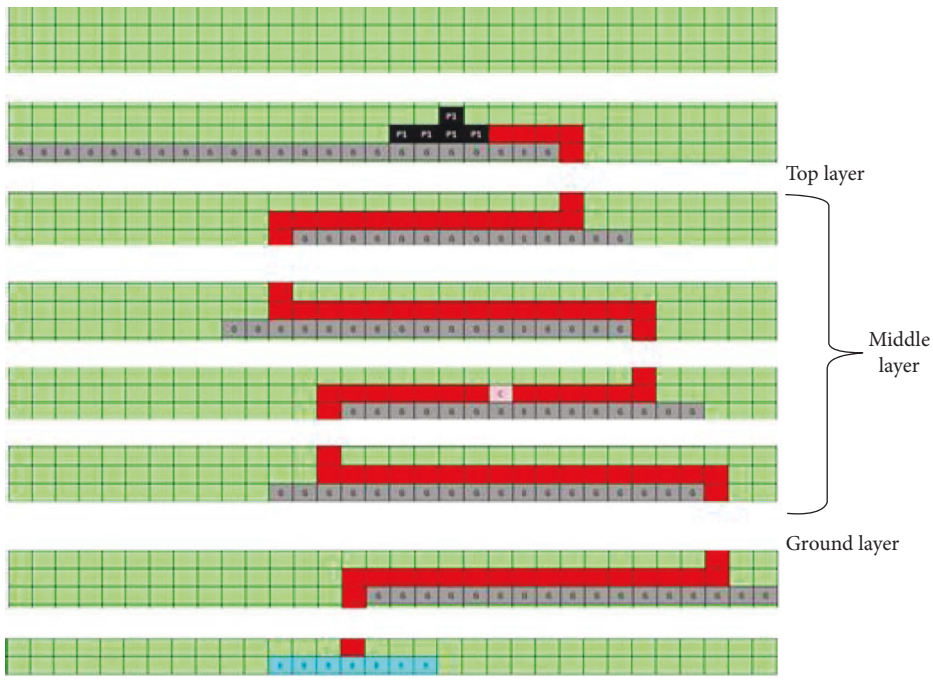


Figure 5: Floor cut plan.

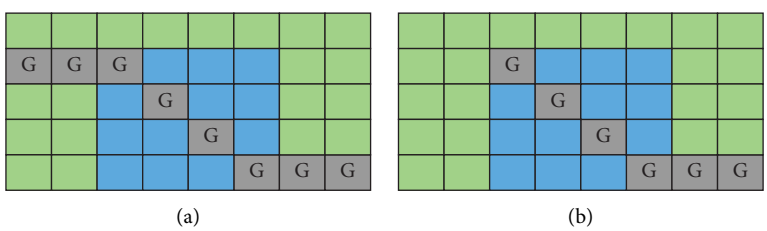


Figure 6: Sloping floor. (a) Double-joint slope. (b) Single-joint slope.

degree slope. It has been tested that can climb uphill, but the 45-degree slope cannot be climbed because it is too steep. The double-joint slope in Figure 6(b) system uses a 45-degree slope, which means it cannot climb.

Next, the system will detect the area that is not affected by the clearance path on the premise where it does not affect the clearance path, and treat this area as a variable area; that is, slope elements can be placed. The method is to detect the shortest distance between the left and right edges of the scene and the clearance path to infer the variable area. Therefore, this research will obtain the minimum X (CMin) and maximum X (CMax) two parameter values; the formula is as follows.

Equation (7) is the calculation formula for variable area:

$$\begin{aligned} \text{CMin} &= \min_{\text{PassPoint}\in\text{PathPass}} (\text{PassPoint}.X) \\ \text{CMax} &= \text{Scene.width} - \max_{\text{PassPoint}\in\text{PathPass}} (\text{PassPoint}.X) \end{aligned}. \quad (7)$$

The value of CMin is the minimum X value of Pass Point on all clearance paths, and the value of CMax is taken as Pass Point. The maximum X value is subtracted from the scene width. Taking Figure 7 as an example, CMin is 11 and CMax is 2.

Because the sloped floor element will occupy a large area, and it is easy to break the rules of the element arrangement in the scene, the system will calculate the clearance path as the center and determine which side of the scene is the largest area and then place the sloped floor in the largest area. The method is to take the middle layer and the bottom layer and calculate the moving cell of each layer from the left and right edges of the scene to the customs clearance path. The formula is as follows:

$$\begin{aligned} \text{Right} &= \sum_{n-1}^{\text{cnt}} R_n \\ \text{left} &= \sum_{n-1}^{\text{cnt}} L_n \end{aligned}. \quad (8)$$

Equation (8) is the formula for calculating the azimuth of the largest open area in the scene. In the formula, cnt is the total number of the middle layer and the bottom layer, L is the cell occupied by each layer from the left of the scene to the customs clearance path, and R is the cell occupied by each layer from the right to the customs clearance path. The

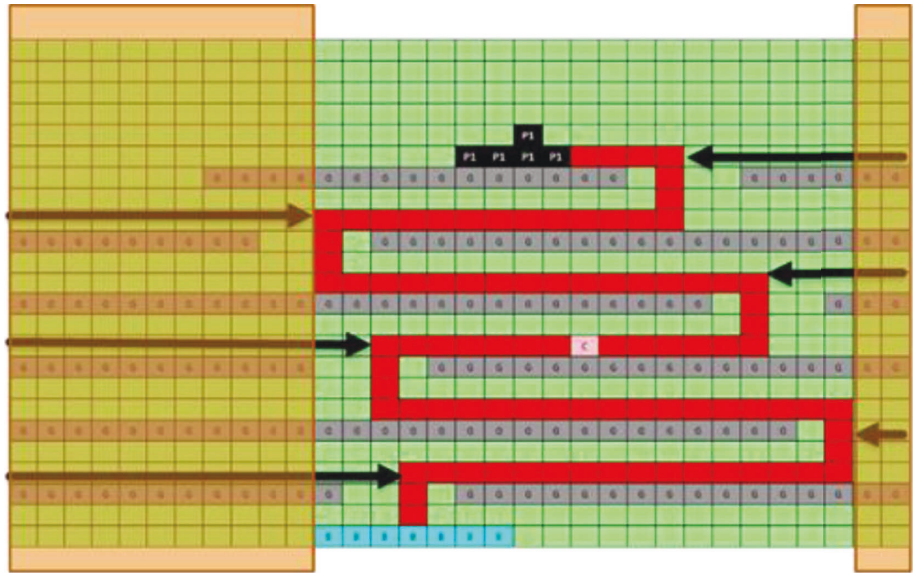


Figure 7: Schematic diagram of the variable area.

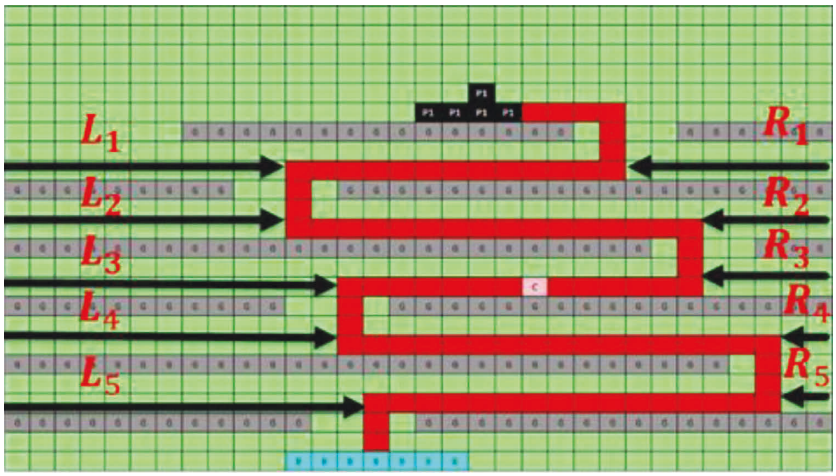


Figure 8: Schematic diagram of calculating the position of the largest open area in the scene.

schematic diagram is shown in Figure 8. After calculating the left and right values, the system will select the larger value as the placement direction.

After obtaining the position of the largest open area and the variable area of the scene, the system will randomly select the ground element of the two floors in the middle layer, namely, two Y, in the position of the largest open area of the scene. The X calculation formula is as (9) slope placement point X value calculation, connecting two points, namely, the slope floor entity; when placing the slope, the system will place the nonphysical area in a square area as a blue block, as shown in Figure 9.

Equation (9) is the calculation of X value of slope placement point:

$$X = \begin{cases} \text{Left} > \text{Right}, & X < \text{Min} \\ \text{Right} > \text{Left}, & X > \max \end{cases}. \tag{9}$$

3.2.5. Placing the Trap Ball. The trap ball element is added to the level here. The system will search for the hole on each floor and randomly choose a place close to the ground at the hole to place its trap ball. This research believes that proximity to the ground element can increase the chance of contact with the player. Regarding the scene, the maximum number of trap balls is set in the trap.max_count parameter in the system setting. After placing trap balls, the result is shown in the yellow area in Figure 3.

3.2.6. Opening the Second Path. This research has successfully added two new elements (slope floor and trap ball), but because the current level mainly retains the ground element according to the path of the level, the path of the level is easy to be seen by the player at a glance, so the system provides a method to open different routes from floor to floor to confuse the customs clearance route, which is to improve the players in each floor to have left and right routes to choose

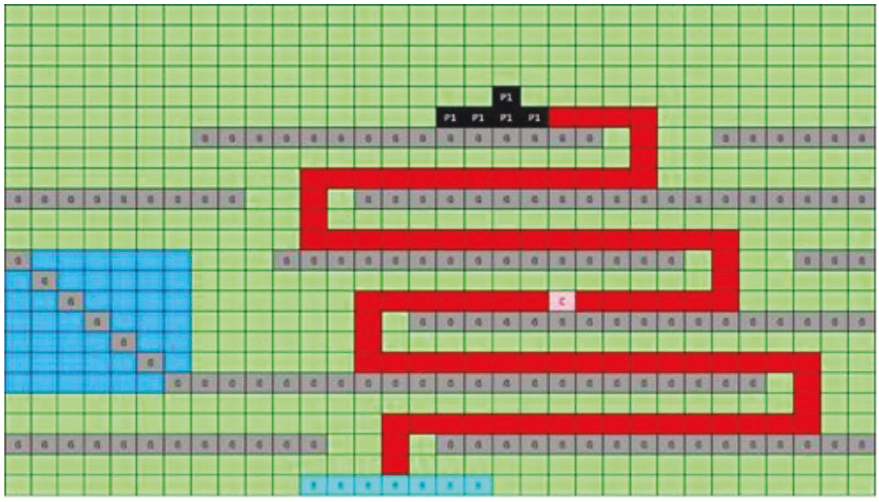


Figure 9: Result of placement of sloping floor.

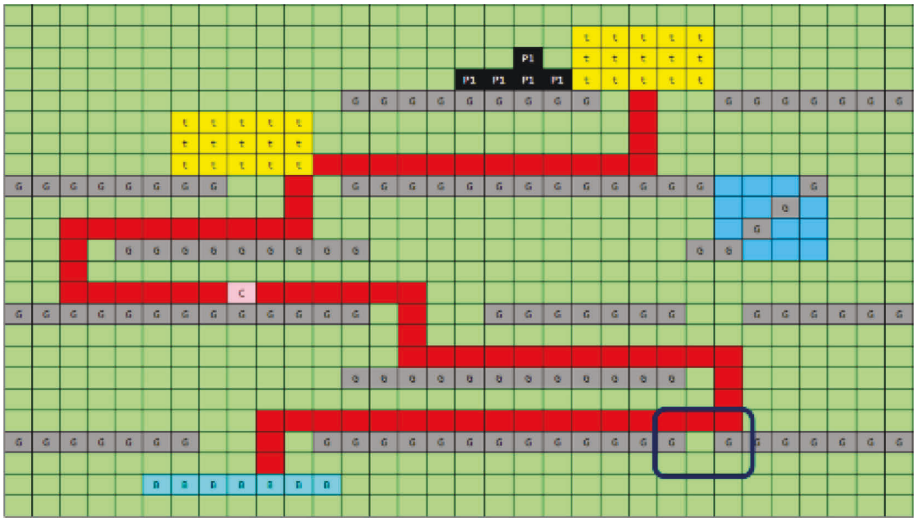


Figure 10: Example of obstructing apple.

from instead of just the clearance route options. The system will take out the Y-axis position of all floors and open the second route in the middle layer one by one. The system will start from the top layer, and there will be one set for every three floors. If there is ground on the first and third floors, delete it. The second layer is the ground. Based on experience, this research found that the level route can be complicated to an S-shaped route, which is also the main concept of this method.

The following randomly takes out an area in the scene as an example. If the floor structure of this area is not the same, it is divided into two blocks for illustration. Small blocks will be found. The ground element is in the second step because the ground element is bound by the clearance path (red square); that is, the second route is still based on the principle of not destroying the clearance path. Refer to the process. After opening the second route, the checkpoint result appears.

3.2.7. Apple Trap Settings. In order to make the animation game a little more difficult, the system provides a method to hinder the transfer of apple. The system first obtains the floor position of the apple element in the level, from the ground floor to the floor where the apple exists, searches for the nonfloor cells of the floor, and obtains the nine-square grid pattern of these nonfloor cells one by one. If the pattern meets the rule in Figure 3, calculate its landing point and create a small hole below the landing point in Figure 10. This hole only allows the apple element to pass, and the element can pass through the hole when accelerating to a certain speed. The existence of this hole causes the player to control the impact. Try to keep the apple element from falling into a hole to increase the difficulty of the animation game.

3.2.8. Level Floor Correction. Since the system has some damage to the distribution of level elements after the actions in the above few subsections, the system will modify the distribution of ground in this step. There are three main processing steps for the modification. The first step is to correct the floor at the edge of the scene in Figure 11. The system will ensure that the width of this element can be used by the (player) element from the ground elements on each

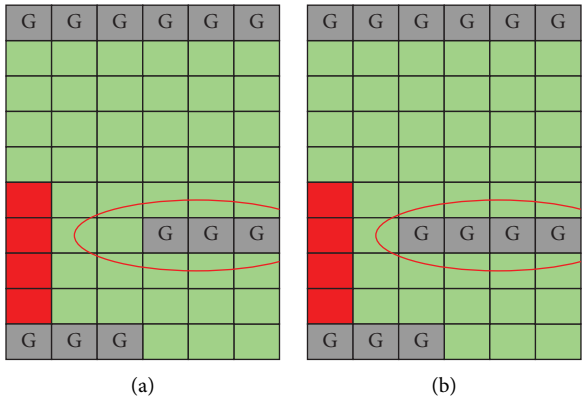


Figure 11: Correcting the edge floor. (a) The length of the edge floor before correction is only 3, which cannot accommodate the length of the element. (b) After correction, the length is 4, which can accommodate elements.

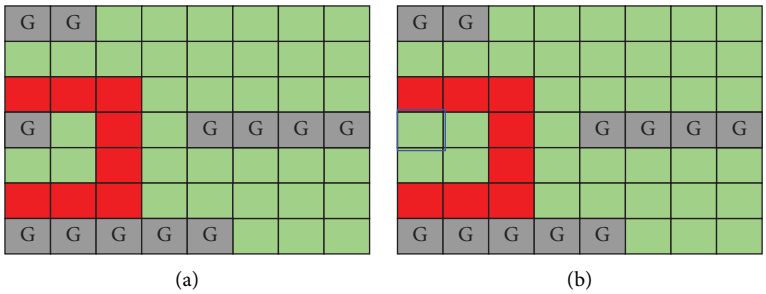


Figure 12: (a) Clearance path hole correction. (b) After the correction, the system decides to keep the edge and delete the floor element in the opposite direction to make the hole. The width of the mouth satisfies the element. Finally, the system will search for all floor elements on the level and check whether it is greater than the width of the element. Floors that are less than the width will be filled to the required width to reduce the existence of unusable floors. The edge floors will also be avoided here.

floor on the left and right sides of the scene. The purpose is to make the floor at the edge of the scene more meaningful. The floor, that is, the element, can stay.

Second, the system will modify the ground element of the landing point position of the customs clearance path to ensure that the landing point of the customs clearance path can allow the (player) to land normally as in Figure 12. At the same time, the system will also ensure that the customs clearance path is smooth, that is, each Pass Point. The sum of the left and right widths must be greater than or equal to the width of the element. This step ensures that the level is playable in Figure 12.

3.3. Output. After the level is complete, the level must be converted to a level file. First, coordinate conversion is required. In the system, its (0, 0) position is in the upper left corner of the cell, but it is at the center of the scene, and the length and width of the elements refer to the width and height in the system setting and the level resolution size (1024∗768) preset and take the aspect ratio value for calculation; then, the conversion can be successful. Regarding the seams of sloped floors, this article also provides a set of equations to calculate the correct placement coordinates. To use this equation, the sloped floors must be divided into two categories, the slope of the single joint is 30 degrees, and the slope of double joints is 45 degrees. For the single-joint slope part, first, convert the slope floor to the animation game coordinates, calculate the X and Y correction values in equation (10), and then add and subtract actions according to the joint points in Table 3 to smoothly change the slope. The floor is connected to the horizontal floor at the joint point.

Equation (10) is the coordinate adjustment of slope floor output conversion:

$$\begin{aligned} X_{\text{fix}} &= 0.683 + (0.433 * (\text{slash.length} - 1)) \\ X_{\text{fix}} &= -0.183 + (-0.25 * (\text{slash.length} - 1)) \end{aligned}. \quad (10)$$

The double-joint sloped floor rotates the horizontal floor by plus or minus 45 degrees, and this article solves the joint problem by calculating the correction value based on experience as shown in Table 4. Under this formula, the placement coordinate of the sloped floor is the center position of the two joints and its coordinates.

4. Experimental Results

4.1. System Architecture. The animation game system architecture in Figure 13 is mainly divided into four parts, including (1) server, (2) animation game database, (3) client, and (4) generator. Among them, the web server mainly

Table 3: Conversion table of single-joint slope correction coordinates.

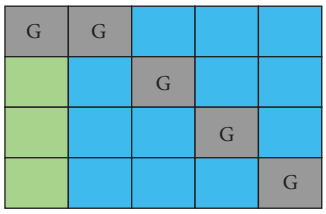
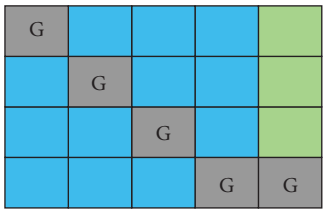
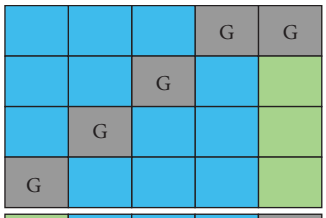
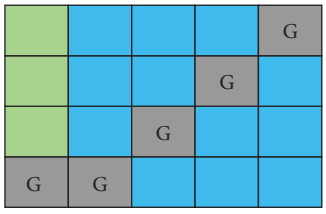
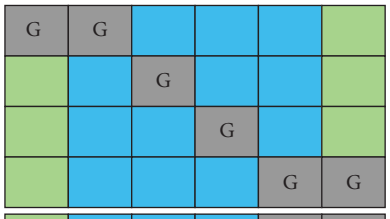
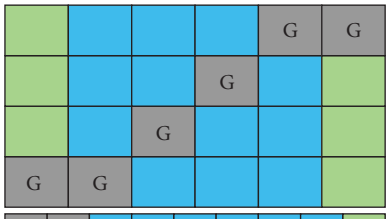
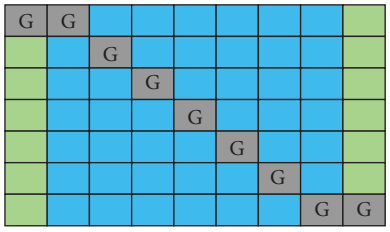
	$Xcur = X + X_{fix}$ $Ycur = Y - Y_{fix}$
	$Xcur = X - X_{fix}$ $Ycur = Y + Y_{fix}$
	$Xcur = X - X_{fix}$ $Ycur = Y - Y_{fix}$
	$Xcur = X + X_{fix}$ $Ycur = Y + Y_{fix}$

Table 4: Corrections.

	$Length_{cur} = 4.5$	Xcur $= X + 0.7375$ Ycur $= Y - 0.0455$
		Xcur $= X + 0.2375$ Ycur $= Y - 0.0455$
	$Length_{cur} = 8.5$	Xcur $= X + 0.65$ Ycur $= Y - 0.145$

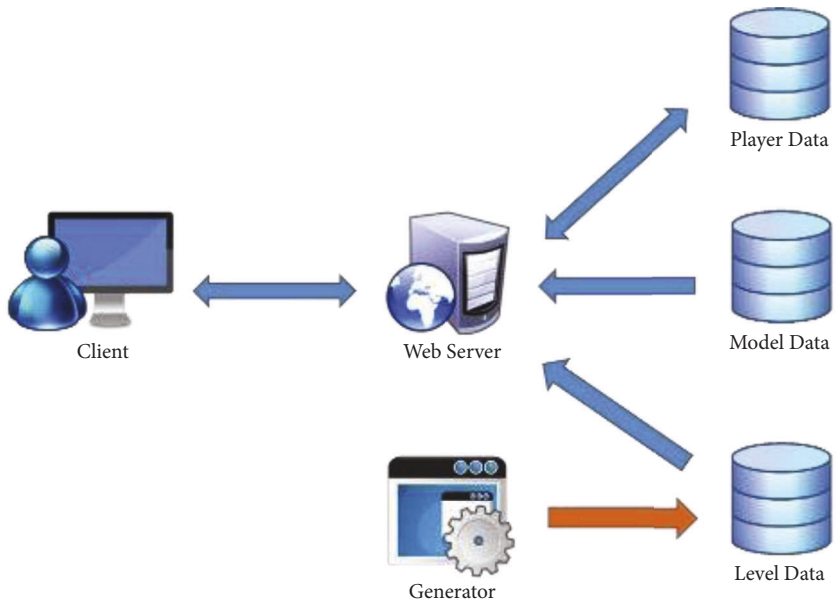


Figure 13: Animation game system architecture diagram.

provides APIs for the client to obtain the resources required by the animation game and the API for uploading player data. The animation game database mainly contains player information, models used by the level, and information about the level. Generator mainly provides automatic generation of levels. The client-side mainly provides the real operation and screen of the player's animation game, obtains the required resources, and uploads the player's data through the API provided by the web server.

In the server and animation game database part, it sets XAMPP as a cross-platform and includes a package of MySQL, Apache web server, and PHP. The environment is set up in CPU: I5-44403, 10 GHz, memory: 20 G, hard disk: 1TB, OS: Windows 10, 64 Bit. This research part of the animation game server uses PHP language to write the animation game interface, which includes animation game login, get animation game theme information, get level data, upload player records, etc. functions. The animation game database mainly stores player information and the models used in the level, and the level information is divided into the following three parts:

(1) Player information: player's animation game account, level clearance record, player's animation game time, the number of replays, the player's control buttons, and the movement of the apple.

(2) Models used in the level: use the Prefab format provided by Unity to package the models used in the animation game (apple, floor, apple bag, trap ball), and the database will store relevant information such as the placement path of the corresponding model.

(3) Level information: store relevant information about each theme level, such as the name of the theme, whether the theme is activated, and the name of the theme file used. The theme file stores the information of the level components.

Table 5: Probability setting table in the experiment.

Object name	Ground	Middle	Top
Ground	0.01	0.89	0.1
Candy	0	0.8	0.2
car1	0	0	1
car2	0	1	0
Bag	1	0	0

The client uses the unity platform and JavaScript language to implement the animation game. Its operation process mainly includes the following four parts:

(1) Log in to the animation game: the animation game will first log in to the animation game account. If it is the first time to play, the system will automatically register a new account based on the MIME code (unique code) of the device.

(2) Load theme information: when the login is completed, the system will automatically load the theme information to form a menu, and players can choose different level themes for the animation game.

(3) Load the level information and the models required by the level: when the player enters a theme to play, the system and the server will obtain the level information file of the theme and the models required by the level and initialize it according to the level information

(4) Record: during the animation game, the system will record the player's operation behavior and clearance data such as animation game time, replay times, operations, and apple movement. These data will be automatically uploaded to the animation game data when passing the level in the library.

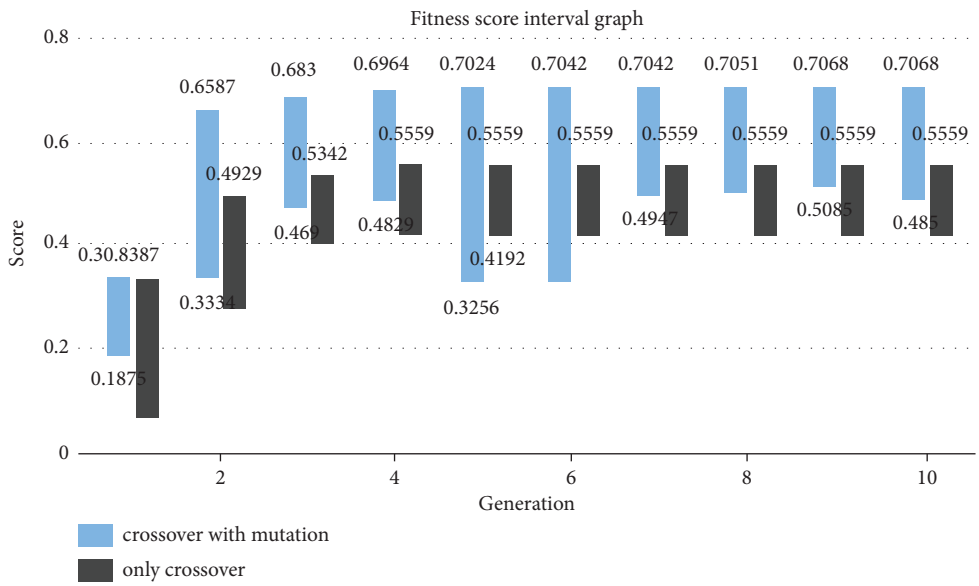


Figure 14: Fitness score change interval graph (10 generations).

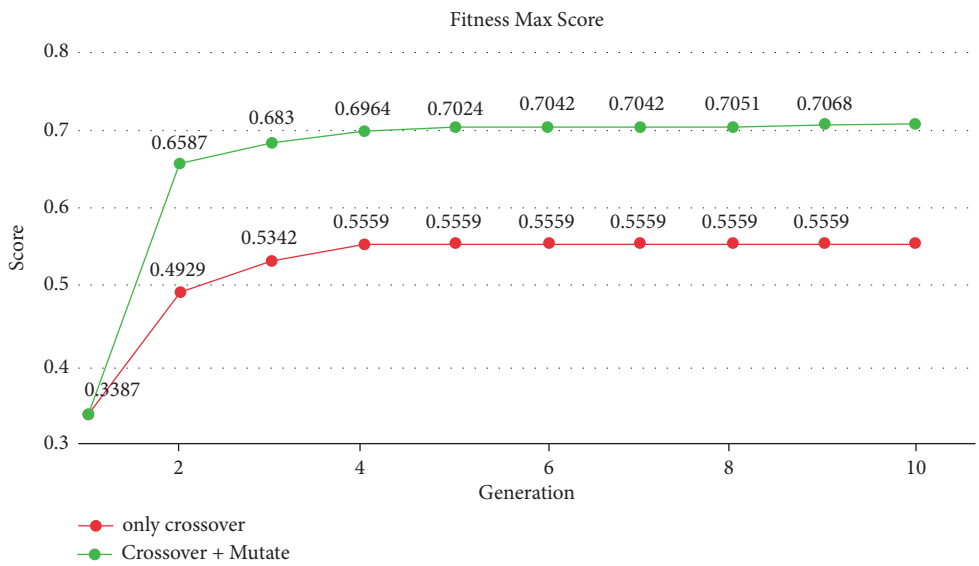


Figure 15: Fitness maximum score change line chart (10 generations).

4.2. Experimental Results. First, use custom-set parameters to generate levels through the system's genetic algorithm, and finally display the experimental results through graphs. Secondly, this research lists the average time spent by the system in each stage of execution. This result will also affect the result of whether it can be operated in real time. Finally, several completed levels will be generated, and they will be verified manually by means of manual inspection to ensure the feasibility of completeness. The experimental parameters are set as in Table 5

4.2.1. Level Generation Verification. This research uses the parameter settings in the previous section to perform genetic calculations for 10 generations and uses mutation to watch the fitness score changes of the cards in each generation (Figure 14). You can see the scores that use mutation. It is much higher because there are many turning points in the clearance path after passing through mutation. Each turning point also means the joint between one floor and the next. This research uses this feature to obtain levels with more floors.

In Figure 15, it can be seen that the genetic algorithm of this research converged rapidly after the first generation. It was because of the correction of mutation, the turning point of the clearance path, and the growth of the clearance path, which greatly improved the fitness score, but due to the limited space of the level scene, the location of the element will affect the maximum value of the turning point and the length of its clearance path.

In Figure 16 shows the standard deviation of the fitness scores of the cards in each generation. The red line segment is the result of not using mutations. It can be seen that the

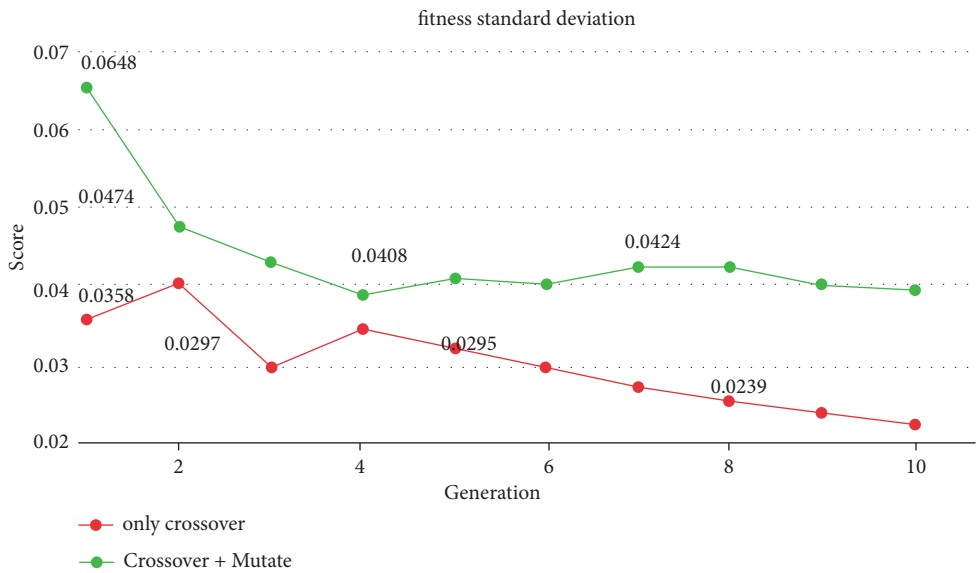


Figure 16: Fitness score standard deviation line chart (10 generations).

Table 6: Average system running time (100 generations).

Process	Average spent time (s)
Init population	17.06
Evaluate population	0.18
Crossover	2.89
Mutation	9.93
Evaluate mating pool	0.30
Elitism	0.33
Total	30.73

standard deviations have a gradual decrease, which means that the scores are getting closer and closer. The reason is that crossover easily affects the length of the gene and indirectly affects the total length of its customs clearance path and the number of turning points, which drives the overall fitness score to gradually increase, and the length only increases but does not decrease in the mating process, which makes its standard. The difference gradually decreases. After the mutation function is added to the green line in Figure 16, it can be seen that the standard deviation will gradually decrease in the previous generations. This is because the mutation of this system has a corrective effect to remedy some low-scoring levels and make them one of the levels. The scores in between can be pulled in quickly, and in the end, the standard deviation will fluctuate within a certain range due to a sudden change in probability.

4.2.2. System Running Time. Here, it allows the system to evolve for 100 generations and records the operating time of the system in each stage. Finally, it averages these data, as shown in Table 6.

It can be seen from Table 6 that the system spends the most time in the process of mating pool initialization and mutation. In the mutation process, the calculation time increases due to the correction function, but the adjustment function can also be seen in the previous section, making its genetic algorithm converge quickly. However, in the mating pool initialization, because the system does not have any genes that can mate with each other at the beginning, the system will determine the placement position according to the probability table and randomly place the animation game objects in the relative position to generate the level and the generated level. It is also necessary to obtain the clearance path through the clearance path search function, and the genetic algorithm can only be executed after the clearance path is obtained. Therefore, the operation process in this part is more complicated and therefore increases the time cost. The calculation time of crossover is affected by modifying Population_Elitism in its system configuration file. Parameter: the system will obtain the permutation and combination whose Population_Elitism parameter takes two in the crossover stage to mate one by one to obtain the mating result and mutation, but because the permutation and combination will cause the number of results to be factorial multiplied, so in the setting of this parameter, it should not be set too high; otherwise it may take up a lot of memory.

Figure 4 shows the total time spent by the system in 100 generations. The line chart shows that the time spent in the system is not proportional to the generations, but the average time spent drifting. The main reason is that the crossover in the system is mating. The top 10 high-scoring genes are selected in the pool for mating, and the mating generation is mutated. Therefore, even if more and more levels pass the threshold in the mating pool, only the first 10 levels will participate in the evolution of the next generation. Process: the main purpose is to use the highest-scored gene to mate a longer gene combination, thereby increasing the fitness score. Of course, the number of mating mothers can be changed by modifying the system parameter settings, but this also means that the execution time of crossover and mutation will increase, leading to increased time costs.

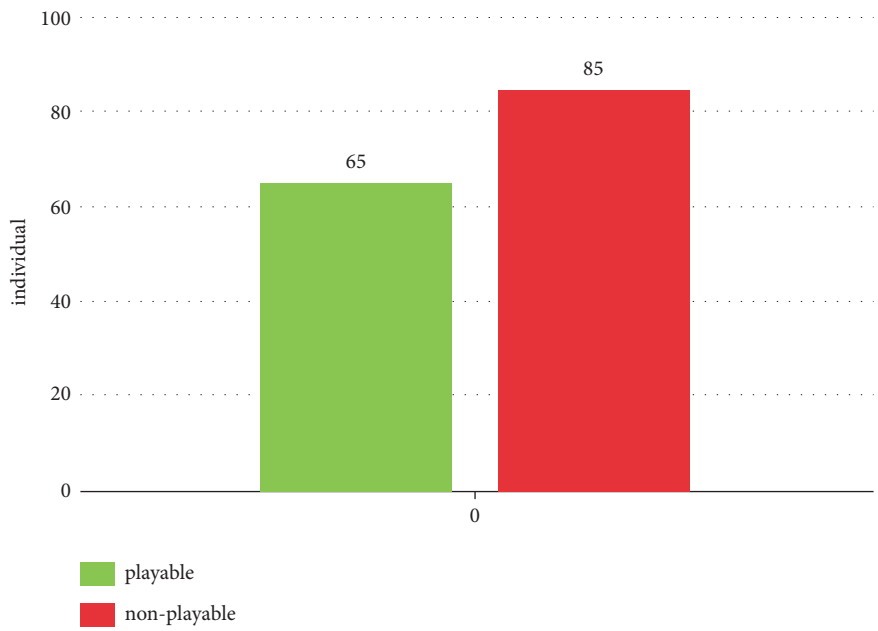


Figure 17: Total playability after the level is complete.

4.2.3. Checkpoint Complete Verification. This research uses genetic algorithms to calculate the results of 10 generations. Each generation takes the level with the highest score of 3, a total of 30 levels. These levels are completed one by one, and each level produces 5 complete levels. Finally, play these 150 levels manually to verify the playability of the completed level. The results are shown in Figure 17. The results show that there are still many levels that cannot be played after the level is completed. The main reason is that even if the system guarantees a smooth path to the level, it puts more animation game objects in the process of completion and fills up the sloped floor. With more horizontal floors, this actually indirectly interferes with the integrity of its clearance path, but if it is not completed, although more than 90% of the playable levels will be playable, there will be too many blank parts in the level. This leads to monotonous levels, so if you want to improve the current situation, you can only define more or more rigorous correction conditions, so the number of playable levels will have room for improvement.

4.2.4. Sloping Floor Correction. Through the provided equation and fixing its single-joint slope to a 30-degree slope, after the double-joint slope is a 45-degree slope, the joint problem between the sloped floor and the horizontal floor can be solved. The research shows the output results of its single-joint and double-joint floor.

5. Conclusion

5.1. Conclusion. This research used unity development tools and JavaScript language to actually create automatic generation levels of animation game systems that can load files for custom levels. The contribution of this article mainly proposes using a genetic algorithm to automatically generate levels and provide a complete level system to process the generated levels, so that the level can have more elements and improve the animation game of the level. This research method is based on the principle of ensuring that the customs clearance path is unobstructed. The research believes that as long as the customs clearance path is unobstructed, it means that the level is playable. In the genetic mutation part, this research has added some corrective mutations to make the next generation more ideal and complete. The system has also formulated many rules to make the level more perfect, and the principle of the path of clearance also allows the complete system to be better played [48].

5.2. Limitations and Future Works. Although this research has completed the generation of levels through genetic algorithms and successfully achieved a more ideal level through the level integration system, it still has limitations and future works are following:

(1) Method improvement: the currently defined fitness function is calculated for the path of the customs clearance, but in fact, the random level generated by the initialization in the entire system actually allows other animation game objects to be placed, but this research only uses the information of the customs clearance path in the end. It does not make full use of all the information possessed by initialization, so there should be a better use space here.

(2) Online real-time generation of levels: if the system can generate levels in real time, it may be necessary to set up a level database and complete the generation of complete levels by obtaining the existing levels in the database. This can achieve real-time effects.

(3) Enhancement of level integrity system: if its playability should reach more than 80%, the optimization system still sets rules for the placement of different animation game objects, but as animation game objects increase.

(4) Sloping floor: since the grid is used to represent a level in the system, there will eventually be joint problems on the sloped floor. At present, this research fixes the slope of the slope and then uses the calculated equation to calculate the correct placement position to solve the joint problem. But this research believes that the most perfect solution should support any sloped floor, so there should be more changes in a level generation.

Data Availability

The raw data supporting the conclusions of this article will be made available by the author.

Consent

The voluntary agreement is free of coercion and undue influence on research participation.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] Algwiki, "What ALG is retrieved 6 29, 2016, from procedural content generation Wiki," 2014, https://ALG.wikidot.com/what-ALG-is.

[2] J. Roberts and K. Chen, "Learning-based procedural content generation," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 1, pp. 88–101, 2015.

[3] A. Angry Birds, "Retrieved from Angry Birds," 2017, https://www.angrybirds.com/.

[4] H. Pei Breivold, "Towards factories of the future: migration of industrial legacy automation systems in the cloud computing and Internet-of-things context," *Enterprise Information Systems*, vol. 14, no. 4, pp. 542–562, 2020.

[5] J. Chen and K. Tai, *Pattern-Based Automatic Game Level Generation and Difficulty Evaluation*, National Dong Hwa University, Taiwan, 2016.

[6] H. Cheng, L. Wu, R. Li, F. Huang, C. Tu, and Z. Yu, "Data recovery in wireless sensor networks based on attribute correlation and extremely randomized trees," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 245–259, 2021.

[7] Y. Cheng, H. Jiang, F. Wang et al., "Using high-Bandwidth networks Efficiently for Fast graph computation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 5, pp. 1170–1183, 2019.

[8] S. Dahlskog and J. Togelius, "A multi-level level generator," in *Proceedings of the Computational Intelligence and Games (CIG) 2016 Conference on Computational Intelligence and Games*, pp. 1–8, IEEE, Dortmund, Germany, August 2014.

[9] Y. Dai, S. Wang, X. Chen, C. Xu, and W. Guo, "Generative adversarial networks based on Wasserstein distance for knowledge graph embeddings," *Knowledge-Based Systems*, vol. 190, Article ID 105165, 2020.

[10] Diablo, "Retrieved from Diablo III Official Game site," 2017, https://tw.battle.net/d3/zh/.

[11] A. Doull, "Ascii Dreams: the Death of the level designer: procedural content generation in Games. Retrieved from the Death of the level designer: procedural content generation in Games - Part One," 2008, https://roguelikedeveloper.blogspot.tw/2008/01/death-of-level-designer-procedural.html.

[12] L. Ferreira and C. Toledo, "A search-based approach for generating Angry Birds levels. Computational intelligence and Games (CIG)," in *Proceedings of the 2014 IEEE Conference on Computational Intelligence and Games*, pp. 1–8, IEEE, Dortmund, Germany, August 2014.

[13] Y. G. Fu, H. Y. Huang, Y. Guan, Y. M. Wang, W. Liu, and W. J. Fang, "EBRB cascade classifier for imbalanced data via rule weight updating," *Knowledge-Based Systems*, vol. 223, Article ID 107010, 2021b.

[14] Y. G. Fu, J. F. Ye, Z. F. Yin, L. J. Chen, Y. M. Wang, and G. G. Liu, "Construction of EBRB classifier for imbalanced data based on Fuzzy C-Means clustering," *Knowledge-Based Systems*, vol. 234, Article ID 107590, 2021a.

[15] Y. G. Fu, J. H. Zhuang, Y. ., P. Chen, L. K. Guo, and Y. M. Wang, "A framework for optimizing extended belief rule base systems with improved Ball trees," *Knowledge-Based Systems*, vol. 210, Article ID 106484, 2020.

[16] Genetic algorithm, "Genetic algorithm retrieved from Wiki," 2017, https://en.wikipedia.org/wiki/Genetic_algorithm.

[17] A. Haleem, M. Javaid, R. P. Singh, S. Rab, and R. Suman, "Hyperautomation for the enhancement of automation in industries," *Sensors International*, vol. 2, Article ID 100124, 2021.

[18] B. Horn, S. Dahlskog, N. Shaker, G. Smith, and J. Togelius, "A Comparative evaluation of procedural level Generators in the Mario AI framework," in *Proceedings of the International Conference on the Foundations of Digital Games*, pp. 1–8, Ft Lauderdale, Florida, U.S.A, April 2014.

[19] X. Y. Li, W. Lin, X. Liu, C. K. Lin, K. J. Pai, and J. M. Chang, "Completely Independent Spanning trees on BCCC data center networks with an application to Fault-Tolerant routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 8, pp. 1939–1952, 2022.

[20] G. Liu, X. Chen, R. Zhou, S. Xu, Y. C. Chen, and G. Chen, "Social learning discrete particle Swarm optimization based two-stage X-routing for IC design under intelligent edge computing architecture," *Applied Soft Computing*, vol. 104, Article ID 107215, 2021.

[21] G. Liu, Z. Chen, Z. Zhuang, W. Guo, and G. Chen, "A unified algorithm based on HTS and self-adapting PSO for the construction of octagonal and rectilinear SMT," *Soft Computing*, vol. 24, no. 6, pp. 3943–3961, 2020a.

[22] G. Liu, X. Zhang, W. Guo et al., "Timing-aware layer Assignment for advanced process technologies Considering via pillars," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 6, pp. 1957–1970, 2022.

[23] G. Liu, W. Zhu, S. Xu, Z. Zhuang, Y. C. Chen, and G. Chen, "Efficient VLSI routing algorithm employing novel discrete PSO and multi-stage transformation," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–16, 2020b.

[24] G. Liu, Y. Zhu, S. Xu, Y. C. Chen, and H. Tang, "PSO-based power-Driven X-routing algorithm in Semiconductor design

for predictive intelligence of IoT applications," *Applied Soft Computing*, vol. 114, Article ID 108114, 2022b.

[25] N. Liu, J. S. Pan, C. Sun, and S. C. Chu, "An efficient surrogate-assisted quasi-affine transformation evolutionary algorithm for expensive optimization problems," *Knowledge-Based Systems*, vol. 209, Article ID 106418, 2020c.

[26] G. Van Den Bergen, *Collision Detection in Interactive 3D Environments*, CRC Press, Boca Raton, 2003.

[27] A. K. Inkulu, M. R. Bahubalendruni, A. Dara, and K. SankaranarayanaSamy, "Challenges and opportunities in human robot collaboration context of Industry 4.0-a state of the art review," *Industrial Robot: The International Journal of Robotics Research and Application*, vol. 49, no. 2, pp. 226–239, 2021.

[28] V. S. S. V. Prasad, M. Hymavathi, C. Rao, and M. A. Bahubalendruni, "A novel computative strategic planning projections algorithm (CSPPA) to generate oblique directional interference matrix for different applications in computer-aided design," *Computers in Industry*, vol. 141, Article ID 103703, 2022.

[29] G. A. Kumar, M. Bahubalendruni, V. Vara Prasad, D. Ashok, and K. Sankaranarayanasamy, "A novel Geometric feasibility method to perform assembly sequence planning through oblique orientations," *Engineering Science and Technology, an International Journal*, vol. 26, Article ID 100994, 2022.

[30] Z. Lu, G. Liu, and S. Wang, "Sparse neighbor constrained co-clustering via category consistency learning," *Knowledge-Based Systems*, vol. 201-202, Article ID 105987, 2020.

[31] Pacman, "Retrieved from pacman," 2017, https://www.gametop.com/download-free/pacman/.

[32] S. Shen, Y. Yang, and X. Liu, "Toward data privacy preservation with ciphertext update and key rotation for IoT," *Concurrency and Computation: Practice and Experience*, Article ID e6729, 2021.

[33] L. Lattanzi, R. Raffaeli, M. Peruzzini, and M. Pellicciari, "Digital twin for smart manufacturing: a review of concepts towards a practical industrial implementation," *International Journal of Computer Integrated Manufacturing*, vol. 34, no. 6, pp. 567–597, 2021.

[34] S. Snodgrass and S. Ontañón, "Experiments in map generation using Markov super Mario Bros (2017) retrieved from super Mario Bros," 2014, https://www.supermariobrosx.org/.

[35] J. Togelius, G. Yannakakis, O. Stanley, and C. Browne, "Search-based procedural content generation," in *Proceedings of the 2010 international conference on Applications of Evolutionary Computation, Volume Part I*, pp. 141–150, Springer, Berlin, Heidelberg, April 2010.

[36] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation:A Taxonomy and Survey," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, pp. 172–186, 2011.

[37] L. Ttito and J. Huacho, "Support vector machine for the implementation of the fitness function of genetic algorithms," in *Proceedings of the 2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 23–26, IEEE, Chaves, Portugal, June 2021.

[38] X. Wu, C. H. Chu, Y. Wang, and W. Yan, "A genetic algorithm for cellular manufacturing design and layout," *European Journal of Operational Research*, vol. 181, no. 1, pp. 156–167, 2007.

[39] A. Kumar Gulivindala, M. Raju Bahubalendruni, R. Chandrasekar, E. Ahmed, M. Haider Abidi, and A. Al-Ahmari, "Automated disassembly sequence prediction for industry 4.0 using enhanced genetic algorithm," *Computers, Materials & Continua*, vol. 69, no. 2, pp. 2531–2548, 2021.

[40] B. Vajgel, P. L. P. Correa, T. Tossoli De Sousa et al., "Development of intelligent Robotic process automation: a utility Case study in Brazil," *IEEE Access*, vol. 9, pp. 71222–71235, 2021.

[41] S. Wang, Z. Wang, K. L. Lim, G. Xiao, and W. Guo, "Seeded random walk for multi-view semi-supervised classification," *Knowledge-Based Systems*, vol. 222, Article ID 107016, 2021.

[42] J. Wewerka and M. Reichert, "Robotic process automation - a systematic mapping study and classification framework," *Enterprise Information Systems*, pp. 1–38, 2021.

[43] Z. Yu, X. Zheng, F. Huang, W. Guo, L. Sun, and Z. Yu, "A framework based on sparse representation model for time series prediction in smart city," *Frontiers of Computer Science*, vol. 15, no. 1, pp. 151305–151313, 2021.

[44] Y. Zhang, G. Huang, X. Liu, W. Zhang, H. Mei, and S. Yang, "Refactoring android Java code for on-demand computation offloading," in *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, pp. 233–248, ACM, October 2012.

[45] H. Zhang, J. L. Li, X. M. Liu, and C. Dong, "Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection," *Future Generation Computer Systems*, vol. 122, pp. 130–143, 2021a.

[46] Y. Zhang, Z. Lu, and S. Wang, "Unsupervised feature selection via transformed auto-encoder," *Knowledge-Based Systems*, vol. 215, Article ID 106748, 2021.

[47] X. Zheng, C. Rong, and T. Badarch, "Foreword to the special issue of green cloud computing: Methodology and practice," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 23, Article ID e5425, 2019.

[48] W. Zou, L. Guo, P. Huang, G. Lin, and H. Mei, "Linear time algorithm for computing min-max movement of sink-based mobile sensors for line barrier coverage," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 2, Article ID e6175, 2022.

Hindawi
Scientific Programming
Volume 2022, Article ID 3943442, 13 pages
https://doi.org/10.1155/2022/3943442

Research Article

Concept Tree-Based Event Matching Algorithm in Publish/Subscribe Systems

Zhi Yuan Zhang,[1] Yu Jie Wang,[2] Xue Hu Huang,[1] and Kai Leung Yung[3]

[1]*School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China*
[2]*State Key Laboratory of Air Traffic Management System and Technology, Nanjing 210014, China*
[3]*Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong 999077, China*

Correspondence should be addressed to Zhi Yuan Zhang; zyzhangcauc@163.com

Received 8 March 2022; Revised 23 June 2022; Accepted 12 August 2022; Published 31 August 2022

Academic Editor: Daniel Mo

Semantic-based publish/subscribe system has attracted a lot of attention in recent years due to its powerful description ability in message dissemination scenarios. As a key part of semantic-based publish/subscribe systems, event matching needs to understand the semantic meaning of subscriptions, especially the hierarchy of concepts. However, existing event matching algorithms are severely affected by the complexity of concept hierarchy trees, some even cannot run due to high memory occupation. This article proposes an event matching algorithm called CTPS (concept tree-based publish/subscribe system) to address it; specifically, we build subscription indexes on concept hierarchy trees for the first time, employ bit arrays to avoid unnecessary matches, and use faster bit operations to accelerate the matching speed. Experiments show that, compared with previous algorithms, CTPS has less memory occupation and shorter event matching time, and its performance is not sensitive to the height of the concept hierarchy tree.

1. Introduction

Sending and receiving messages are very common in many scenarios, such as pushing restaurant or hotel information to online users in location-based service systems. Different from peer-to-peer message delivery methods, publish/subscribe (pub/sub) systems use a decoupled messaging architecture [1], in which a broker collects subscribers' subscription interests and publishers' published messages (often referred to as events in pub/sub systems) and route the events to corresponding subscribers automatically. In this way, an event can be received by multiple subscribers at the same time. Pub/sub systems can be classified into three types according to the subscription model, namely topic-based, content-based, and semantic-based. In topic-based pub/sub systems, events need to carry topic information when publishing, and subscribers also need to submit their interesting topics, and the broker matches them simply through topic information. IBM's MQ [2] and Apache's Kafka [3] are popular topic-based pub/sub system products. Event matching in topic-based pub/sub systems is highly efficient, but its subscription granularity is coarse and its description ability is weak. In content-based pub/sub systems, subscribers are no longer confined to topics but can specify constraints on the content of events. For example, if one wants to purchase a computer with a price less than 400$, in content-based pub/sub systems, he may send a subscription message of (<string, target, =, "computer"> ∧ <string, currency, =, "$"> ∧ <int, price, ≤, 400>) to the broker, where each constraint has a form of <*data type, attribute name, operator, value*>, and ∧ is a conjunction (logical and), which means every constraint must be satisfied. Content-based pub/sub systems such as SIENA [4, 5], REIN [6], GEM [7, 8], H-Tree [9], and BE-Tree [10]greatly enhance the description ability and make the system more flexible and have been a research focus in recent years.

Content-based pub/sub systems can only match subscriptions according to the structure information of events,

which is not enough to understand their semantic meanings, and thus may deviate from users' original intentions. For example, if there is a provider who is selling laptops with a price of 350$, he may publish an event message of (<string, target, "laptop"> <string, currency, "$"> ∧ <int, price, 350>) to the broker. However, in content-based pub/sub systems, without knowing the semantic relation between *computer* and *laptop*, the broker only finds that *laptop* ! = *computer*, thus will inevitably fail to deliver this message to the above subscriber. To address this problem, some researchers try to incorporate semantic information into pub/sub systems, such as S-ToPSS [11], which uses a very intuitive three-step process to solve the semantic matching problem: At first, it translates all attributes with different names but same meaning to a *root* attribute. Then, it uses a concept hierarchy tree to organize the hypernym-hyponym relations of concepts, that is to say, if the attribute/value in the subscription is the ancestor of the corresponding attribute/value in the event, the matching condition is also satisfied. And finally, it uses functions to express more flexible semantic equivalence relations, such as "professional experience = present date – graduation year." S-ToPSS replaces each attribute of an event with all its possible hypernym; in this way, one event may become multiple new events when matching. Obviously, this method artificially increases matching times and is inefficient. In subsequent research of semantic-based pub/sub systems, ontology is often used to model events, and RDF (resource description framework) graph, which is similar to ontology querying, is often used to model subscriptions. As ontology is a general framework for knowledge and concepts, the description ability of semantic-based pub/sub systems such as OPS [12], G-ToPSS [13], MIC [14], and iBroker [15]is further improved, and the users' subscription intention is better understood; therefore, it has got much attention in recent years. However, all these methods are severely affected by the complexity of the concept hierarchy tree: S-ToPSS [11] expands one event to many events with all attributes' hypernym (ancestors), thus increasing matching times; MIC [14] expands one subscription to many subscriptions with all attributes' hyponym (descendants), thus increasing the memory usage of subscription index and cannot manage a large number of subscriptions; and G-ToPSS [13] and iBroker [15] traverse the hierarchy tree and check them one by one when matching, thus the matching efficiency is low.

To address these problems, this article proposes CTPS (concept tree-based pub/sub system) model, which makes the matching efficiency effectively improved. The main contributions of this article are as follows: (1) We propose a new event matching algorithm called CTPS, which is the first time to build a subscription index on concept hierarchy trees. By CTPS index structure, a subscription does not need to store multiple times, thus memory occupation is dramatically reduced. Also by CTPS index structure, an event does not need to match multiple times either, and unmatched subscriptions can be quickly filtered, thus event matching efficiency is greatly improved. (2) Instead of using complex RDF graphs, we use simple <*attribute*, *value*> and <*attribute*, *constraint*> pairs as event and subscription models, respectively, and together with the concept hierarchy tree, the description ability is powerful enough for semantic-based pub/sub systems.

2. Related Works

In semantic-based pub/sub systems, events and subscriptions are usually modeled by RDF graphs. OPS [12] takes event matching as a subgraph isomorphism problem, in which all nodes in the subscription graph are thought to be variables, so the algorithm complexity is very high. In G-ToPSS [13], only ?*x* in form of (?*x*, *op*, *v*) is regarded as a variable, where *op* may be a relational operation used for literal value comparison or may be a hypernym-hyponym relational operation used for concept/class comparison. For example, "laptop" < "computer" means the former is a hyponym or descendant concept of the latter. G-ToPSS builds a two-level hash index for all edges in the subscription graphs. The first level is the start and end vertices of an edge, and the second level is the edge label, together with a subscription list. G-ToPSS applies two processing stages for event matching: In the first stage, a completed graph is constructed for each edge (*s*, *p*, *o*) in the event graph, which means no matter the corresponding secondary index contains p or not, G-ToPSS will always search the following pairs in the index space:so, s^*i, $^*io$, and $^*i$ $^*j$ ($^*i$ and $^*j$ are variables), and the matched subscriptions are added to a processing set. In the second stage, for each subscription in the processing set, the variables are set with proper values to check whether the constraints are satisfied. If the checks are about hypernym-hyponym constraints, it will take a lot of time to traverse the concept hierarchy tree. MIC [14] (multidimensional index matching count) algorithm builds a three-level hash index for subscriptions, different from G-ToPSS, the edge labels are stored in the first level, and the vertex pairs are stored in the second level. If the second-level index contains variable vertices, the variable range will be divided into multiple intervals organized by a binary tree as the third level. For hypernym-hyponym constraints, MIC uses the hyponym expansion operation, that is, for any edge (*s*, *p*, *o*) in the subscription graph, new edges in the form of (*Ts*, *Tp*, *To*) will also be added to the index, where *Ts, Tp,* and *To* are descendants of *s*, *p*, and *o*, respectively. Therefore, there is no need to worry about the hypernym-hyponym problems when matching events; however, the index space increases dramatically, especially when the concept tree is high. In [16], subscriptions are translated into SPARQL queries, and the query condition is also extended as in MIC. In iBroker [15], both classes and attributes are stored in the first-level hash index, and the second-level index stores the next class or attribute to be matched, thus pulling the constraints of each subscription into a linked list. In essence, it merges identical vertices/edges in different subscription graphs into the same node, and the node value is then set to be true or false in the subsequent matching process. Finally, the list is traversed to get the matching result.

The above pub/sub systems are all centralized, that is, there is a broker who manages all subscriptions and is responsible for event message routing. There are also

decentralized peer-to-peer pub/sub systems, such as OpenPubSub [17], which proposes a hybrid event routing model that combines rendezvous routing and gossiping over a structured peer-to-peer network. The network is built based on a high-dimensional semantic vector space. This article only focuses on centralized pub/sub systems.

3. CTPS Model

To better understand the relation of concepts involved in events and subscriptions, concept trees must be defined at first, then events and subscriptions can use terms defined in these concept trees. Thereafter, the matching algorithm can use the concept tree to say whether an event's attribute/value is satisfied with a subscription constraint. The concept model is used to express concept relations, and the event/subscription model is used to express the event/subscription message structure sent to the broker.

3.1. Concept Model. The concept model reflects the hierarchical relation of classes and attributes, such as "computer" is the parent of "laptop PC," as shown in Figure 1. Similarly, attributes may also have hierarchical relations. For example, both "*cellphone number*" and "*email*" are subattributes of contact information. If there is only one node in the concept tree, it is called an independent concept, such as "MoneyValue" in Figure 2. In this way, each attribute could be an independent attribute or a hierarchical attribute, and also the attribute value could be an independent class or a hierarchical class. The xsd:string and xsd:decimal, which are frequently used in ontology, are regarded as independent classes. We specify that the attribute name is unique, and the attribute value type is determined by the attribute name, which can be obtained by domain(attribute) function, for example domain(target) = product, domain(contact information) = xsd:string. In addition, root(name) function can be used to obtain the corresponding root node's name, for example root (computer) = product. As for independent concepts, the root node's name is itself, for example root(MoneyValue) = MoneyValue.

3.2. Event Model. In semantic-based pub/sub systems, events are usually represented as RDF graphs. Figure 2 shows an example, indicating that John is selling a HP desktop PC with a 40G IBM hard disk for $450. This article uses a more concise event model, which is composed of <*attribute*, *value*> pairs, where *attribute* is the attribute name defined by the attribute hierarchy tree, and *value* is the attribute value, which can be a class name defined by the class hierarchy tree or a literal value. With a specification that attribute name is unique and it can determine its value type, for example price.value and hardDisk.size.value are both numeric, and *target* is of "*product*" type (see Figure 1), we can then use the following nine pairs to represent the event graph in Figure 2. Note that if the value type is an independent class, which means that we can exactly know its value type by the attribute name, then the pair for describing its value type can be omitted, such as < price.type, MoneyValue>. It can be seen that, compared with the original RDF event graph, our model is more concise. In addition, because event data are often drawn from relational databases in most cases, our model reduces the workload for there is no need to additionally convert events into RDF graphs.

<seller.name, John>
<seller.cellPhoneNumber, 123456789>
<price.value, 450>
<price.currency, units:$>
<target, Desktop PC>
<target.manufacture.name, IBM>
<target.hardDisk.size.value, 40>
<target.hardDisk.size.units, units:*G*>
<target.hardDisk.manufacture.name, IBM>

3.3. Subscription Model. In semantic-based pub/sub systems, subscriptions are often represented as graphs. As an example, Figure 3 shows a subscription graph about purchasing a computer with a price less than $400. In the graph, each vertex consists of 2 or 3 components; the first is the vertex ID, the second is the value type, and the third is an optional constraint. The subscription model in this article is also < attribute, constraint > pairs, where attribute is the attribute name and constraint is the constraint condition. There are two kinds of constraint: one is literal value constraint on basic data types such as numeric or string, and the other is type value constraint on class data types. For type value constraints, it is specified that the constraint conditions are only satisfied when the event value is the constraint value itself or is its subclass. For example, let "Desktop PC" and "computer" be event and constraint value, respectively, since "Desktop PC" is a subclass of "computer," the constraint is satisfied. The subscription graph in Figure 3 can be represented as the following four pairs, where interval [*lower*, *upper*] is used to denote numeric constraints. As we assume that the attribute name can be used to determine its value type, so if the attribute value type is unique, its type value constraint will be excluded.

<target, Computer>
<price.currency, units:$>
<price.value, [0,400]>

4. Index Structure

There may be several ways to express one thing, such as "Desktop PC" and its Chinese term "台式机," or "US" for short of "United States." To translate attributes and values in different names but with the same meaning, a synonym hash table in the form of <key, value> is designed, where "value" is the root words and "key" is all possible statements of them. In this way, the attribute names/values in events and subscriptions are easy to be replaced with their corresponding root words before processing. In addition, the unit types in events and subscriptions are also unified before processing,

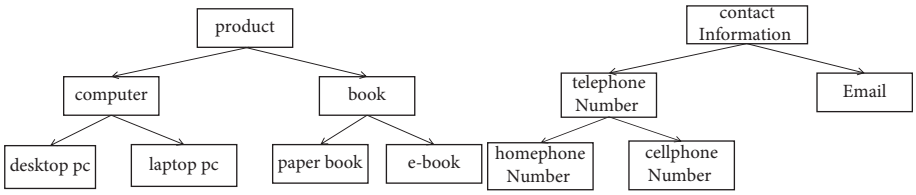


Figure 1: Example of hierarchy tree for classes (a) and attributes (b).

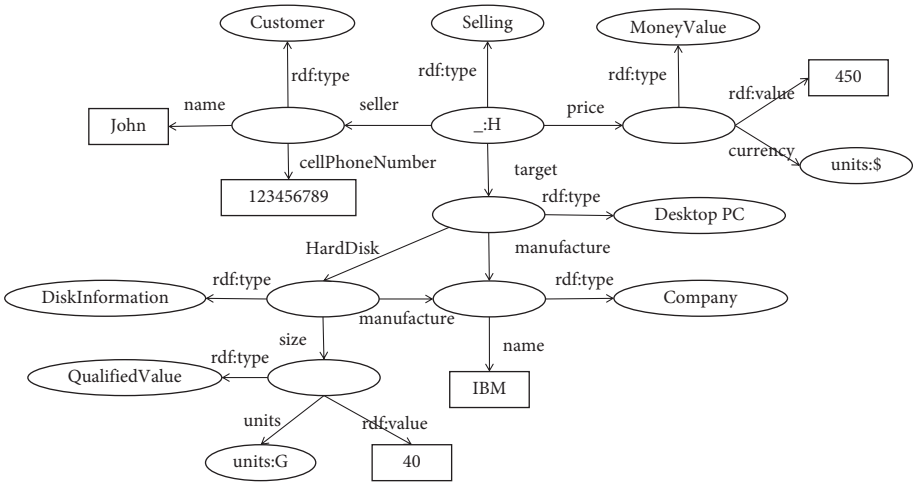


Figure 2: Example of an RDF event graph.

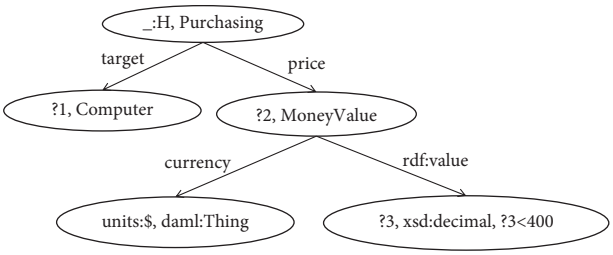


Figure 3: Example of a subscription graph.

such as "target.hardDisk.size.units" is transformed to *G*. By doing this, pairs about attribute unit can also be omitted, such as <price.currency, units:$> and <target.hardDisk.size.units, units:*G*> in the above event and <price.currency, units:$> in the above subscription. At the same time, a unique ID is assigned to each subscription.

4.1. Hierarchy Tree Index. An index is built on each concept hierarchy tree. As shown in Figure 4, the index consists of three parts: The left part is a hash table, where the *key* is attribute name/value and the *value* is linked to its corresponding node of the tree; the right part is a hierarchy tree, where each node consists of an attribute name/value and a list of subscription IDs, and each node is linked to its parent node; and the last part is the total subscription list of the hierarchy tree, as shown in the figure, *SID*: {1,4,5,10}. The subscription lists of nodes and the total subscription list *SID* are all stored in bit arrays, like *BitSet* in Java. Different from *integer list* which needs to be compared one by one, the bit array uses bit operations and greatly speeds up the matching speed. It should also be noted that, unlike the child node representation used in Figure 1, we use the parent node representation here, because only ancestor searching is needed in our algorithms.

Data Structure:

```
TreeNode{
    String name;
    TreeNode parentNode;
    BitSet sid;
}
TreeIndex{
    Map < String, TreeNode > hashTable;
    BitSet SID;
};
Map < String, TreeIndex > m1, m2;
```

The hierarchy index tree is built by Algorithm 1: *buildTreeIndex*. For each concept hierarchy tree, if it is an attribute tree, such as the right part of Figure 1, we build and initialize the corresponding index tree, and put it into *m*1 so we can quickly find it by the attribute name (2–6). If it is an attribute value tree, such as the left part of Figure 1, we build and initialize the corresponding index tree, and put it into *m*2 for subsequent search (7–11). The algorithm *initTreeIndex* recursively traverses the concept

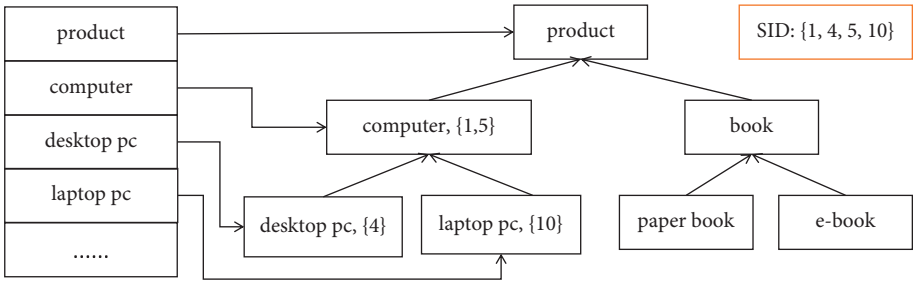


Figure 4: Example of a hierarchy tree index.

```
      buildTreeIndex( ){
(1)     for each hierarchy tree{
(2)       if it is an attribute tree pt{
(3)         TreeIndex ptree = new TreeIndex();
(4)         initTreeIndex(pt, null, ptree);
(5)         for each node name pname of pt, m1.put(pname, ptree);
(6)       }
(7)       if it is an attribute value tree vt{
(8)         TreeIndex vtree = new TreeIndex();
(9)         initTreeIndex(vt, null, vtree);
(10)        for each node name vname of vt, m2.put(vname, vtree);
(11)      }
(12)    }
      }
      initTreeIndex(ct, pNode, tree){
(1)     node = new TreeNode( );
(2)     node.name = ct.name;
(3)     node.parent = pNode;
(4)     tree.hashtable.put(node.name, node);
(5)     for each childNode of ct{
(6)       initTreeIndex(childNode, node, tree);
(7)     }
      }
```

Algorithm 1: Build and initialize a hierarchy index tree.

hierarchy tree, uses the parent node representation to initialize the index tree, and adds hash table entries as shown in the left part of Figure 4 so as to quickly access the corresponding index node and its ancestors by the attribute name.

Algorithm 2: insertSubscribe adds each constraint of subscription s into the index structure. First, we add the attribute name into the attribute index tree (2–3), and then add it into different index structures according to the type of the attribute value: string and numeric types are added to the literal constraint index (5–10), and other types are added into the attribute value index tree (11–13). Algorithm 3: *insertTreeConstraint* adds subscription ID into the subscription list of the node according to the attribute name/value and also adds it into the total subscription list in the index tree (2–4). As an example, for constraint <*target*, *computer*>, there is no need to add an index for *target* because it is an independent attribute (line 1), while "*computer*" is not, so we find its node in the *hashtable* (line 2), and then for the node subscription list and the total subscription list of the tree, we set its corresponding element to be 1(3–4).

4.2. Literal Constraint Index. We only consider two literal types: string and numeric. For string types, we only consider the equivalent constraint, and its index is a two-level hash table, where the first level is the attribute name, and the second level is the string value of the attribute with a subscription list, which is also stored in a bit array, as shown in Figure 5. A special item $*$ is added to the second-level index to match all possible string values of the attribute name. Algorithm 4 shows how to insert string constraints. Although there may be many nodes in an attribute hierarchy tree, they have the same value type, so we build only one string index by its root node.

Data structure is as follows:

Map < String, Map < String, BitSet >> $m3$;

```
insertSubscribe(s){
  for each pair < attribute, constraint> of s{
    ptree = m1.get(attribute);
    insertTreeConstraint(ptree, attribute, s.id);
    switch(domain(attribute)){
    case xsd:string:
      insertStringConstraint(attribute, constraint, s.id);
      break;
    case xsd:decimal:
      insertGEMConstraint(attribute, constraint, s.id);
      break;
    default:
      vtree = m2.get(attribute);
      insertTreeConstraint(vtree, constraint, s.id);
    }
  }
}
```

Algorithm 2: Insert subscription.

```
insertTreeConstraint(tree, pvName, id){
  if tree is null, return;
  node = tree.hashtable.get(pvName);
  node.sid.set(id);
  tree.SID.set(id);
}
```

Algorithm 3: Insert attribute name/value class constraint.

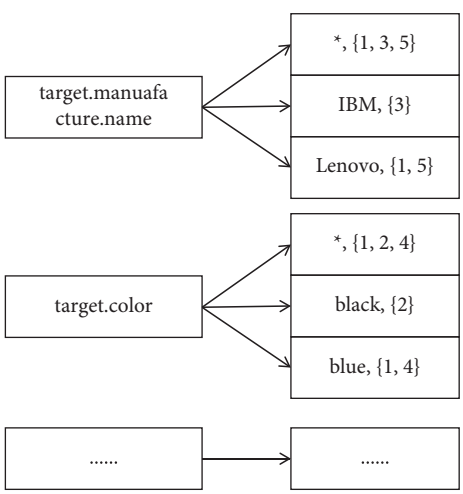


Figure 5: Example of string constraint indexes.

The numerical constraint index is also a hash table, in which the *key* is an attribute name, and the *value* is a GEM [7] index, as shown in Figure 6. Each cell of the GEM index has a subscription list, which is also stored in a bit array. The x-axis of the GEM index is the lower bound of interval constraints, and the y-axis is the upper bound of interval constraints. Since the lower bound must be less than or equal to the upper bound, constraints satisfying this condition all fall into the upper triangular region in the plane. The value range R_m is divided evenly into T boxes both on the x-axis and y-axis, and in this way, a constraint will only fall into one *cell*. Each *cell* is denoted as a coordinate (y, x), where y and x are calculated as follows:

$$y = \begin{cases} \dfrac{\text{upper}}{R_m/T}, \text{upper} \neq R_m \\ T-1, \text{otherwise} \end{cases} \quad x = \begin{cases} \dfrac{\text{lower}}{R_m/T}, \text{lower} \neq R_m \\ T-1, \text{otherwise} \end{cases}. \tag{1}$$

Assuming a constraint of [0, 20], $R_m = 100$, and $T = 5$, then the *cell* index is (1,0). Since the calculation of the *cell* index is independent of the number of constraints, the subscription insertion speed of GEM is a constant.

Data structure is as follows:

```
GEMCell{
  BitSet sid;
  Map < id, constraint > list;
}
GEMIndex{
  int T, Rm; GEMCell cell[T][T]; }
  Map < String, GEMIndex > m4;
```

```
insertStringConstraint(attribute, value, id){
    name = root(attribute);
    map = m3.get(name);
    if map is null{
        map = new Map < String, BitSet>( );
        m3.put(name, map);
    }
    sid = map.get(value);
    if sid is null{
        sid = new BitSet( );
        map.put(value, sid);
    }
    sid.set(id);
    aid = map.get(*);
    if aid is null{
        aid = new BitSet( );
        map.put(*, aid);
    }
    aid.set(id);
}
```

Algorithm 4: Insert string constraint.

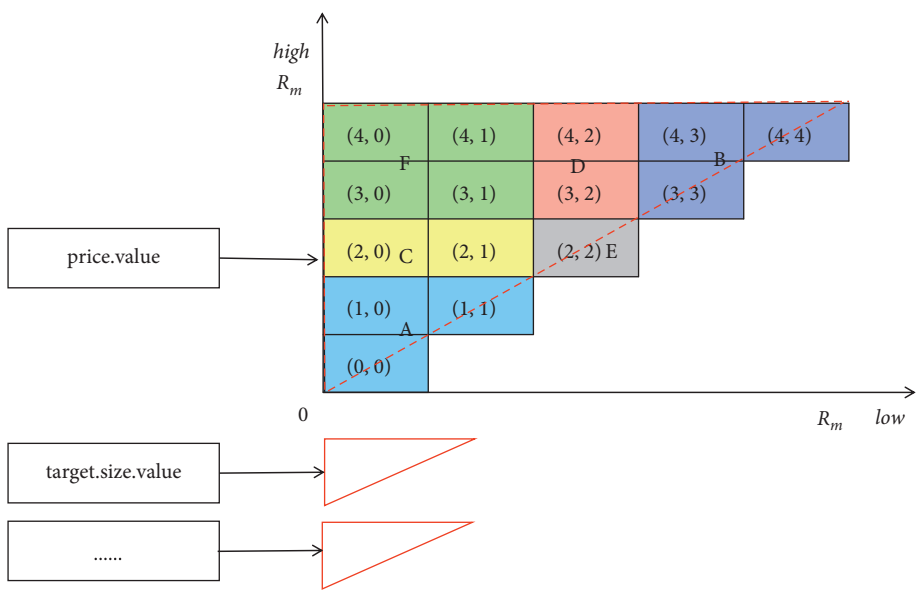


Figure 6: Example of a numerical constraint index.

Algorithm 5: insertGEMConstraint inserts a numerical constraint in the form of [*lower*, *upper*] into the *cell* of GEM. First, we need to find the corresponding GEM index by the root name of the attribute in *m*4 (1–2). If it is not found, a new hash table entry should be inserted (3–6). Then, we calculate *cell* index according to the formula and set the corresponding ID of the subscription list to be 1(7–8).

5. Event Matching Algorithm

In event matching, we delete unmatched subscriptions as early as possible. First, we initialize a bit array with a length of subscription number and set all its elements to be 1. Then we match each pair of the event in turn; if a constraint is not satisfied, then the location of its subscription ID in the bit array is set to be 0, which means it would not be matched anymore, and, finally, those subscriptions that are 1 in the bit array are matched. Since subscriptions are no longer matched after being set to 0, lots of invalid matching operations are avoided, so the matching efficiency is improved. As shown in Algorithm 6, for each pair <*attribute*, *value*> of an event, at first we find the index tree by the root name of the attribute (3–4), then we match the *attribute* if the index tree is not null (5–7), and then we match the *value* according

```
insertGEMConstraint(attribute, constraint, id){
    name = root(attribute);
    gem = m4.get(name);
    if gem is null{
        gem = new GEMIndex( );
        m4.put(name, gem);
    }
    compute index y, x by [lower, upper] of the constraint
    gem.cell[y][x].set(id);
}
```

Algorithm 5: Insert numeric constraint.

```
(1)  match(e){
(2)      initial a BitSet sid with a length of subscription number, and set all its elements to be 1
(3)      for each pair <attribute, value> of e{
(4)          name = root(attribute);
(5)          ptree = m1.get(name);
(6)          if ptree is not null{
(7)              sid = treeMatch(ptree, attribute, sid);
(8)          }
(9)          switch(domain(attribute)){
(10)         case xsd:string:
(11)             sid = stringMatch(name, value, sid);
(12)             break;
(13)         case xsd:decimal:
(14)             sid = gemMatch(name, value, sid);
(15)             break;
(16)         default:
(17)             vtree = m2.get(name);
(18)             if vtree is not null{
(19)                 sid = treeMatch(vtree, value, sid);
(20)             }
(21)         }
(22)     return sid;
         }
```

Algorithm 6: Event matching algorithm.

to the value type of the attribute (8–20), and, finally, the matching result is returned (line 22).

5.1. Hierarchy Tree Matching Algorithm. If the constraint's attribute name is the child node of the event's attribute name, they cannot match. Similarly, if the constraint's attribute value is the child node of the event's attribute value, they cannot match either. Algorithm 7: treeMatch performs event matching on attribute names/values. First, we find the corresponding node of the attribute name/value in the index tree and save the subscription list of the node and its ancestors in a temporary bit array tmpid (1–6). Then we perform bit operations and get the matching result sid (7–9). The resize function on line 7 sets tree.sid to the same length as sid by padding zeros. For example, tree.sid = {0,1,1}, sid = {0,1,1,1}, which means there are 4 subscriptions and only the first and second subscriptions (index starting from 0) are indexed in the tree, and after resize function, tree.sid = {0,1,1,0}. Supposing that tmpid = {0,1}, that is, only the first subscription matches, so the second subscription should be set to 0, and the result of the calculation is (~{0,1,1,0} | {0,1,0,0}) & {0,1,1,1} = {0,1,0,1}. If the length alignment is not performed, the system will take the ~ operation first, which may get a wrong result.

5.2. String Matching Algorithm. Algorithm 8: stringMatch is used for attribute values of string type. First, we get the hash table from $m3$ according to the attribute name, and if we get nothing, further operation will not be required, for there is no string constraint on it at all (lines 1–2). Then, we get subscriptions matching the attribute value (line 3) and subscriptions matching all possible values of the attribute

```
    treeMatch(tree, key, sid){
(1)     node = tree.hashtable.get(key);
(2)     tmpid = new BitSet( );
(3)     while node is not null{
(4)         tmpid = tmpid | node.sid;
(5)         node = node.parentNode;
(6)     }
(7)     resize(tree.sid, sid.length);
(8)     sid = (~tree.sid | tmpid) & sid;
(9)     return sid;
    }
```

Algorithm 7: Tree matching algorithm.

```
    stringMatch(key, value, sid){
(1)     map = m3.get(key);
(2)     if map is null, return sid;
(3)     tmpid = map.get(value);
(4)     aid = map.get(*);
(5)     resize(aid, sid.length);
(6)     sid = (~aid | tmpid) & sid;
(7)     return sid;
    }
```

Algorithm 8: String matching algorithm.

(line 4) and perform bit operations similar to algorithm 7 (lines 5–7).

5.3. Numeric Matching Algorithm. Algorithm 9: gemMatch is used for attribute values of numeric type. First, we get the GEM index according to the root name of the attribute (line 1). If it does not exist, which means that there is no numeric constraint on this attribute at all, we just return without doing anything (line 2). Otherwise, unmatched subscriptions should be removed. Using formula in 4.2, the cell index r of the *value* is calculated (2–4), and the index structure is divided into 5 parts by r, labeled as A to E as shown in Figure 6. As an example, let $R_m = 100$, $T = 5$, and $value = 45$, it is easy to know that r is 2.

(1) For cells in part A, $y < 2$, that is, $upper < 40$, it is impossible to match the event, so subscriptions in these cells will be removed (5–7);

(2) For cells in part B, $x > 2$, that is, $lower > 60$, it is impossible to match the event, so subscriptions in these cells will also be removed (8–10);

(3) For cells in part C, $x < 2$, that is, $lower < 40$, there is no need to check the lower bound, so only subscriptions with constraint whose upper bound is less than 45 will be removed (11–17);

(4) For cells in part D, $y > 2$, that is, $upper > 60$, there is no need to check the upper bound, so only subscriptions with constraint whose lower bound is greater than 45 will be removed (18–24);

(5) There is only one cell ($x = y = 2$) in part E, each constraint in it should be checked, and if it does not meet the condition, its corresponding subscription will be removed (25–30);

(6) For cells in part F, $y > 2$ and $x < 2$, that is, $upper > 60$ and $lower < 40$, the event is certainly matched, so there is no need to deal with it.

Different from the original GEM event matching algorithm, when processing parts A and B, the original algorithm needs to traverse all subscriptions in each *cell*, while it is changed to bit operations in this algorithm to speed up matching. And also, when dealing with parts C, D, and E, the original algorithm needs to check every subscription in each *cell*, regardless of whether it is invalid or not, while this algorithm first finds the intersection of *sid* and the subscription list of the current *cell* (lines 12, 19, 26), and then only valid subscriptions are checked. Thus, the number of comparisons is reduced and the matching efficiency is improved.

5.4. Algorithm Analysis. Index maintenance complexity analysis: For any subscription, *attribute* of every constraint should be inserted into the hierarchy tree index $m1$, and *value* of every constraint should be inserted into $m2$ (for class value), $m3$ (for string value), or $m4$ (for numeric value). For $m1$ and $m2$, each insertion includes 1 hash operation, 2 bit operations; for $m3$, each insertion includes 3 hash operations and 2 bit operations; for $m4$, each insertion includes

```
gemMatch(key, value, sid){
  gem = m4.get(key);
  if gem is null, return sid;
  if value = gem.Rm, r = gem.T - 1;
  else r = value/(gem.Rm/gem.T);
  for each pair (x,y) that 0 ≤ x<=y < r{
     sid = sid & ~gem.cell[x][y].sid;
  }
  for each pair (x,y) that r + 1 ≤ x<=y < gem.T{
     sid = sid & ~gem.cell[x][y].sid;
  }
  for (x = 0; x< r; x++){
     tmpid = sid & gem.cell[x][r].sid;
     for each bit 1 index of tmpid{
        if gem.cell.[x][r].list.get(index).upper < value
           sid.set(index, 0);
     }
  }
  for(y = r + 1; y < gem.T; y++){
     tmpid = sid & gem.cell[x][r].sid;
     for each bit 1 index of tmpid{
        if gem.cell.[r][y].list.get(index).lower > value
           sid.set(index, 0);
        }
  }
  tmpid = sid & gem.cell[r][r].sid;
  for each bit 1 index of tmpid{
     c = gem.cell[r][r].list.get(index);
     if c.lower > value ||c.upper < value
        sid.set(index, 0);
  }
  return sid;
}
```

Algorithm 9: Numeric matching algorithm.

1 hash operation, 1 GEM index computation, and 1 bit operation. Operations are same for subscription deletion. Note that none of these operations are proportional to subscription numbers, and the maintenance complexity is nearly $O(1)$.

Space complexity analysis: It mainly includes the hierarchy tree index of attribute names $m1$, the hierarchy tree index of attribute values $m2$, the string index $m3$, and the numeric index $m4$. Suppose the number of root attributes is a. There are a_1 of them whose attribute name has a hierarchical class, a_2 of them whose attribute value has a hierarchical class, a_3 of them whose attribute value type is a string, and a_4 of them whose attribute value type is a numeric. There is $a_1 \le a$ and $a_2 + a_3 + a_4 \le a$. We suppose that there are n subscriptions in total. Each hierarchy tree has at most k nodes, and each string attribute takes at most b values. Then the space complexity is $a_1kn + a_2kn + a_3(b+1)n + a_4T^2n = ((a_1 + a_2)k + a_3(b+1) + a_4T^2)n = O(n)$, which is proportional to the number of subscriptions.

Time complexity analysis: Suppose the height of the hierarchy tree is h. Matching an attribute name of hierarchical class requires 2 hash operations, and a maximum of $h+1$ "|" operations, one "~"operation, and one "&" operation, that is, a total of 2 hash operations and $h+3$ bit operations are required to match the attribute name hierarchical class. The match of an attribute value hierarchical class is similar to this procedure. Matching a string value requires 3 hash operations and 3 bit operations. For numeric index matching, if the constraints are evenly distributed, each *cell* contains about $2n/T^2$ constraints: for each *cell* of sections A and B, 2 bit operations are required. Considering that A and B account for half of all *cells* on average, a total of $2\,T^2/4 = T^2/2$ bit operations are required. In the worst case, one comparison operation is required for the constraints in each *cell* of sections C and D, and two comparison operations are required for the constraints in each *cell* of section E, that is, a total of $(T-1+2)*2n/T^2 = 2(T+1)n/T^2 \approx 2n/T$ comparison operations. Supposing that the cost of one hash operation is p, the cost of one bit operation to is βn (the longest bit array is n, so the cost of bit operations is proportional to n), and the cost of one comparison operation is q. Hence, the time cost is $(a_1 + a_2)\ (2p + (h+3)\beta n) + a_3(3p + 3\beta n) + a_4(\beta n * T^2/2 + q * 2n/T) = (2a_1 + 2a_2 + 3a_3)p + [(a_1 + a_2)\ (h+3)\beta + 3a_3\beta + a_4T2\beta/2 + 2a_4q/T]n$, and the

Table 1: Parameter settings.

Parameter name	Parameter value	Memo
Attribute number with concept tree	6	
Attribute value number with concept tree	4	
Attribute value number with basic type	2	xsd:decimal
Height of concept tree	1~5	Tree with only root node has height 0
Width of concept tree	2	Each nonleaf node has 2 child nodes
Nodes (edges) of subscription/event graph	7(6)	7 nodes, 6 edges
Subscription number	500 ~ 10000	
Event number	1000	

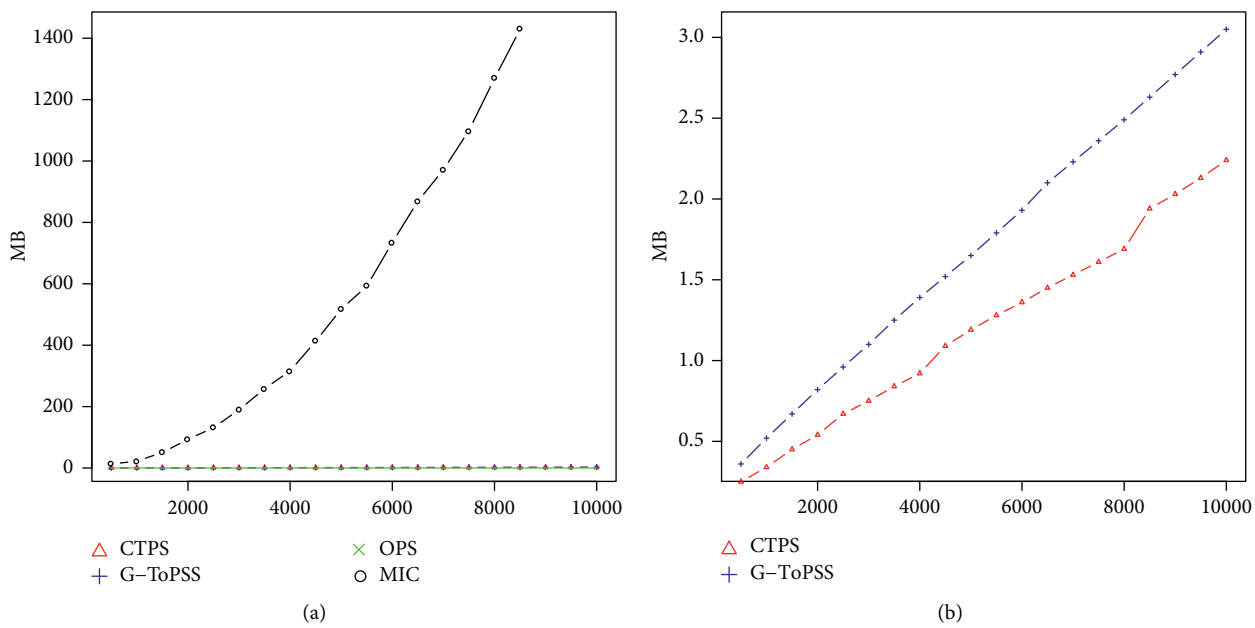


Figure 7: Comparison of memory usage: (a) subscribe (height = 1); (b) subscribe (height = 5).

complexity is $O(n)$. Due to the fast speed of the bit operation, the β value is small and the matching time will grow relatively slowly with the number of subscriptions.

6. Experiments

The CTPS prototype system is developed using JDK1.8. Experiments are run on a Windows10 laptop with an Intel i5-6200 2.40 GH CPU and 8G memory, and the experiment data are synthetic. Parameter settings are listed in Table 1.

In Figure 7(a), we compare the memory occupation of the four algorithms CTPS, G-ToPSS [13], OPS [12], and MIC [14] and the height of tree is 1 (one root node with two child nodes). Since MIC algorithm expands the original subscription, even when the tree height is only 1, it will expand $3*3*3 = 27$ times. It can also be seen that with the increase in subscription number, the memory space of the MIC increases sharply. When the subscription number is greater than 8500, the memory of MIC overflows, while the growth of CTPS and G-ToPSS is relatively slow. It should be pointed out that the OPS index only stores all possible vertices and edges, and the memory occupation has nothing to do with the subscription number and only grows rapidly with the height of the tree. When the tree height is 4, OPS occupies about 505 MB (Mega Byte), and it overflows when the tree height is 5. As the tree height increases, the number of subscriptions that MIC can handle also decreases rapidly. When the tree height is 4, only 1000 subscriptions can be handled in MIC, and when the tree height is 5, this number drops to less than 500. Therefore, in Figure 7(b), we only compare the memory occupation of CTPS and G-ToPSS when the tree height is 5. It can be seen that the memory occupation of the two increases linearly with the number of subscriptions. The growth rate of CTPS is smaller than that of G-ToPSS, and the total amount is relatively small. When the number of subscriptions is 10000, the memory is less than 3 MB.

In Figure 8, we compare the matching time of the four algorithms CTPS, G-ToPSS [13], OPS [12], and MIC [14]. The number of events is 1000 and the number of subscriptions varies from 500 to 10000. In Figures 8(a) and 8(b), we compare the matching time of the four algorithms when the tree height is 1. Since OPS matches subscriptions one by one, its matching time is so much longer than others, so graph 8(a) is drawn separately. It can be seen from Figure 8(b) that when the number of subscriptions is greater

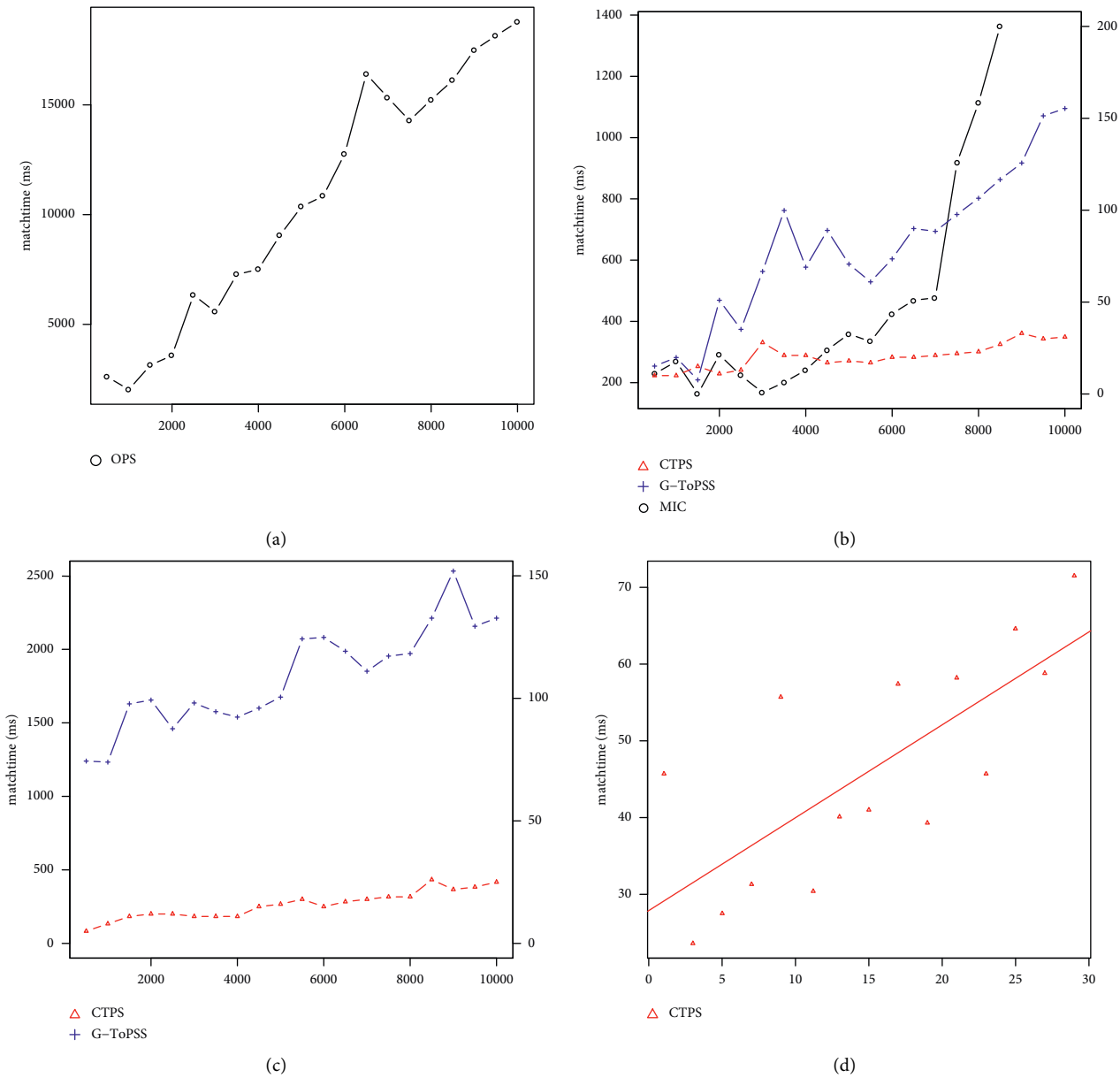


Figure 8: Comparison of matching time (number of events = 1000): (a) subscribe(height = 1); (b) subscribe(height = 1); (c) subscribe(height = 5); (d) matchrate(%) and subscribenum = 10000.

than 7000, the matching time of the MIC rises sharply, possibly due to the sharp increase in its memory usage. The matching time of G-ToPSS grows slowly with the number of subscriptions, while CTPS does not change much. It should be noted that since the matching time of CTPS is much shorter than other algorithms, Figure 8(b) sets a right vertical axis for CTPS alone. In Figure 8(c), we compare the matching time of CTPS and G-ToPSS algorithms when the tree height is 5, where G-ToPSS uses the blue vertical axis on the left and CTPS uses the red vertical axis on the right, and we can see that CTPS is about 70 times faster than G-ToPSS. Compared with Figure 8(b), the matching time of G-ToPSS increases as the tree height increases, while CTPS remains basically unchanged. In Figure 8(d), we compare the time used for CTPS under different matching rates. It can be seen that the matching time of the CTPS algorithm grows slowly with the increase of the matching rate, but the overall matching time is still very short. To sum up, CTPS is superior to other algorithms in terms of memory occupation and matching time, and the performance is not affected by the height of the concept tree.

7. Conclusion

Semantic-based pub/sub system has powerful description abilities, but its event matching algorithm is severely affected by the complexity of concept trees. We propose a new index structure called CTPS, which builds indexes on the concept hierarchy tree, though which unmatched subscriptions can be quickly filtered. By using bit arrays in the index, invalid

matches are avoided, and by fast bit operations, the matching time is further shortened. Experiments show that the algorithm in this article is significantly superior to existing algorithms in terms of memory occupation and matching time.

Data Availability

Data are available on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Zhiyuan zhang proposed CTPS algorithm and completed the manuscript; Yujie Wang performed the experiments; Xuehu Huang completed the coding of G-ToPSS algorithm; and Kai Leung Yung improved the editing quality of this article.

Acknowledgments

This work was supported by the States Key Laboratory of Air Traffic Management System and Technology of China (grant no. SKLATM201902).

References

[1] P. T. Eugster, P. A. Felber, R. Guerraoui, and A. M. Kermarrec, "The many faces of publish/subscribe," *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131, 2003.

[2] I. B. M. RedBook, *Internet application development with MQSeries and Java*, IBM, Armonk, NY.USA, 1997.

[3] J. Kreps, N. Narkhede, and J. Rao, "Kafka: a distributed messaging system for log processing[C]," *6th International Workshop on Networking Meets Databases (NetDB)*, vol. 11, 2011.

[4] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Design and evaluation of a wide-area event notification service," *ACM Transactions on Computer Systems*, vol. 19, no. 3, pp. 332–383, 2001.

[5] A. Carzaniga and A. L. Wolf, "Forwarding in a content-based network[C]," *Proceedings of ACM SIGCOMM*, pp. 163–174, ACM, Karlsruhe, Germany, 2003.

[6] S. Qian, J. Cao, Y. Zhu, and L Minglu, "REIN: a fast event matching approach for content-based publish/subscribe systems," in *Proceedings of the IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014.

[7] W. Fan, Y. Liu, and B. Tang, "GEM: an analytic geometrical approach to fast event matching for multi-dimensional content-based publish/subscribe services," in *Proceedings of the IEEE INFOCOM 2016*, April 2016.

[8] W. Fan, P. Xiong, F. Wu, and Y Liu, "GEM-tree: tree-based analytic geometrical multi-dimensional content-based event matching," *IEEE Access*, vol. 7, pp. 164089–164101, 2019.

[9] S. Qian, J. Cao, and Y. Zhu, "H Tree an efficient index structure for event matching in publish/subscribe systems," in *Proceedings of the IFIP Networking Conference, 2013*, April 2013.

[10] M. Sadoghi and J. H. A. . BE-Tree, "An index structure to efficiently match Boolean expressions over high-dimensional discrete space," *Acm Sigmod International Conference on Management of Data*, ACM, Athens, Greece, 2011.

[11] M. Petrovic, I. Burcea, H. A. . S Jacobsen, and S. S. ToP, "Semantic Toronto Publish/Subscribe System," *VLDB 2003*, pp. 1101–1104, Berlin, Germany, 2003.

[12] J. Wang, B. Jin, L. I. Jing, and S Dan Hua, "Data model and matching algorithm in an ontology-based publish/subscribe system," *Journal of Software*, vol. 16, no. 9, pp. 1625–1635, 2005.

[13] M Petrovic, L Haifeng, and J Hans Arno, "G-ToPSS: Fast filtering of graph-based metadata," *WWW 2005*, pp. 539–547, ACM, Chiba, Japan, 2005.

[14] X. Hu, "Matching algorithm for semantic-based publish/subscribe system[J]," *Journal of Zhejiang University*, vol. 43, no. 1, pp. 63–68, 2009.

[15] M. J. Park and C. W. Chung, "iBroker: an intelligent broker for ontology based publish/subscribe systems," *2009 IEEE 25th International Conference on Data Engineering*, in *Proceedings of the*, pp. 1255–1258, Shanghai, China, March 2009.

[16] H. Zhang, X. Zhang, K. Ding, and L Ming, "A fuzzy matching with reasoning publish/subscribe system based on ontology," in *Proceedings of the 2022 2nd international conference on consumer electronics and computer engineering (ICCECE)*, pp. 150–156, Guangzhou, China, January 2022.

[17] T. Zaarour, A. Bhattacharya, and E. Curry, "OpenPubSub: supporting large semantic content spaces in peer-to-peer publish/subscribe systems for the internet of multimedia things," *IEEE Internet of Things Journal*, vol. 1, 2022.

Hindawi
Scientific Programming
Volume 2022, Article ID 5106036, 10 pages
https://doi.org/10.1155/2022/5106036

Research Article

Digitized Visual Transformation of Grotto Art Using Deep Learning and Virtual Reality Technology

Hao Xie [1,2]

[1] *College of Fine Arts, Sichuan University of Science and Engineering, Zigong 643000, China*
[2] *The Catholic University of Korea, Bucheon, Republic of Korea*

Correspondence should be addressed to Hao Xie; xiehao@suse.edu.cn

Received 15 March 2022; Revised 1 July 2022; Accepted 10 August 2022; Published 29 August 2022

Academic Editor: Yung Po Tsang

The purpose is to convert the contents of the grotto murals into a digital vision based on deep learning (DL) and virtual reality (VR) technology to protect the precious grotto murals. The grotto murals are analyzed and modeled by introducing the related concepts of the digital library, using VR technology, and taking the cultural relics data acquisition technology as the basic framework. Then, the cultural relics information collection technology under DL is combined with VR technology, the color and materialization concepts of grotto murals are integrated, and a digital recognition model of grotto mural restoration and simulation is constructed. The color analysis scheme of grotto murals is established. Through the analysis of color and material, the comprehensive recognition rate of this color model can reach 70%, and the material recognition of color painting can also ensure one-to-one restoration. Besides, the model can be applied to the simulation and restoration of grotto murals and provide some help for the protection of grotto murals. This paper has practical application value for the protection and restoration of mural art.

1. Introduction

Grotto murals have always been one of China's most precious artistic treasures. Chen et al. [1] believed that their artistic and historical reference values were immeasurable. Ma et al. [2] also pointed out that the grotto murals added rich color to China's long history. Stratford et al. [3] pointed out that most of the grottoes were located in harsh areas such as the Gobi and the desert, with more sandstorms all year round. Hence, the erosion of the grottoes becomes increasingly serious, causing severe damage to the treasure. Armitage et al. [4] held that how to protect the grotto murals better had always been a difficult problem for archaeologists in recent years. Some studies have adopted network video technology to record the grotto murals to let more posterity enjoy the charm of grotto murals. However, Yin et al. [5] found that this method only took videos and pictures that could not be appreciated from multiple angles. He et al. [6] believed that the type of technology adopted to restore the grottoes is crucial.

In studying wind sand erosion in grottoes, Liang et al. [7] pointed out that wind was the main dynamic condition affecting Gobi dust emission, and surface dust content was a significant factor determining dust emission. Besides, the contribution of wind to PM10 emission is greater than surface dust content. The higher the height is, the greater the weight of friction velocity is and the greater the damage to the grottoes is. Mogao Grottoes is one of the most representative grottoes in the Northern Wei Dynasty. Bao [8] pointed out that the "Crouching Tiger, Hidden Dragon" painting on the south wall of the grottoes had high artistic value. However, this painting's original appearance has changed with the changes in dynasties and the development of the times. In particular, the color of the whole grotto has undergone subversive changes, which has led to multiple misunderstandings, indicating that the preservation of grotto murals is significant. Wu et al. [9] found that the tomb mural was a special mural buried underground. Due to the narrow exit of the tomb passage, the tomb mural is excavated by dividing the whole mural into blocks, which loses a lot of information between blocks. Therefore, digital image technology is adopted to restore the missing part to achieve a certain purpose of preservation. For the restoration of

grottoes, Zhao et al. [10] combined semantic acquisition with knowledge association logic in information images and innovated the image resource analysis mode and the concept of digital restoration. The 4S-IAM model was proposed to connect historical graphics resources, local archaeological achievements, and mural image ontology resources effectively. However, Bi et al. [11] found that the tunnel wall's heat and moisture transfer process changed in space and time. The complex heat and moisture transfer directions change periodically. Besides, the water transfer in the tunnel wall reaches a stable state faster than the heat transfer. The damp heat cycle plays a vital role in the development of mural degradation. Hence, the research on the grottoes preservation technology is one of the most crucial works in archaeology. The digital visual transformation of grotto mural art contributes to psychological influence and affects subjective well-being. However, because residents have a less visual appearance to perceive and experience the global contemporary architecture and urban design trends, this paper uses the mobile virtual reality (VR) platform to study the art of grotto murals. The content of model inspection is the degree of digital transformation of grotto mural art.

The purpose is to realize the digital visual transformation of the mural art in grottoes and improve the probability of mural preservation and restoration. VR technology is used and combined with deep learning (DL). A virtual scene that is completely consistent with the real murals can be restored by analyzing the style and materials of grotto murals. The use of VR enables observers to observe and record grottoes anytime and anywhere through specific equipment. First, the concepts of grotto murals, VR, and DL are briefly introduced. The main research contribution is to identify the color and material of the grotto murals by combining DL and VR technology, judge the system's performance, and provide a reference for the preservation and recording of the grottoes. The main research innovation is to use VR technology to simulate real grottoes, rather than using a single image restoration and recording to complete the preservation of grottoes. This paper has crucial reference value for the preservation and restoration of murals.

2. Literature Review

Many scholars have conducted relevant research on DL technology. Zhou et al. [12] explored and studied the application of DL in food and investigated dozens of studies using DL as a data analysis tool to solve problems and challenges in the food field. It includes food identification, calorie estimation, and quality detection of fruits, vegetables, meat, and aquatic products. The survey results show that DL is superior to other methods. Using DL as a tool for food quality and safety inspection can improve the accuracy of food safety inspection. Van et al. [13] studied DL technology in histopathology, summarized its current application situation in this field, and described the application challenges of artificial intelligence in histopathology. Research shows that DL can improve the speed and accuracy of medical diagnosis. Jacob et al. [14] studied the application of DL in image acquisition of the Internet of Things. Good test results show that the accuracy of DL can improve the accuracy of image recognition. Besides, Samadbeik et al. [15] studied the application of VR technology in medical group teaching. The results show that using virtual education technology for laparoscopic surgery training can improve the accuracy of medical practice. Bashabsheh et al. [16] studied the application of VR technology in architectural teaching and used VR technology to present some three-dimensional models of construction stages. The results show that VR software can better realize model construction compared to traditional teaching methods. To sum up, grotto mural art's reproduction and digital transformation with the support of DL and VR technology can enhance the protective effect of precious cultural relics murals. The purpose is to establish a visual model of grotto mural art and compare the previous research methods with the research methods here. Wang et al. [17] studied the identity recognition algorithm based on DL, introduced the traditional method of person Re-ID, and analyzed its advantages and disadvantages. The results show that the combination of DL and traditional methods is vital. Laxton [18] evaluated and studied the surrealism of painting in the 19th century. The results show that modernist painting needs to be explored and recognized by DL technology. DL has crucial reference value for the digital transformation of grotto art. Li et al. [19] used DL to improve immune peptides in proteomics. The results show that compared with the existing methods, the DeepRescore rescoring method enhances the sensitivity and reliability of mass spectrometry binding peptides and new antigen identification. Farazmand [20] believed that grotto art was an ancient art exhibition integrating architecture, sculpture, and painting used for religious activities. It is a work with a high combination of artistry and practicality, so it has the role of artistic decoration and artistic protection of sculpture art. Duan et al. [21] found that colored sculptures existed in grotto temples in about 25 provinces, cities, and autonomous regions in China. Wang et al. [22] pointed out that China's most famous grotto art included Qiuci Caves, Mountain Grottoes, Mogao Caves, Yungang Grottoes, The Longmen Grottoes, and Dazu Rock Carvings. To sum up, combined with the relevant research on grotto art, the use of DL and VR technology can provide technical support for restoring and protecting grotto art.

3. Methods

3.1. Deep Learning. DL is a machine learning method, which integrates the prediction results of multiple classifiers into a learning model to improve robustness and classification performance. VR is a computer simulation system that combines virtual and reality to create and experience a virtual world. The integration method of DL can improve the classification from any single classifier, but the combination classifier is not always meaningful. It is worth combining different classifiers only when the detection rate is high. The color classification of the grottoes is also obvious. Figure 1 shows several main types.

Dunhuang Mogao Grottoes is a place integrating modeling, color sculpture, and murals. Due to the

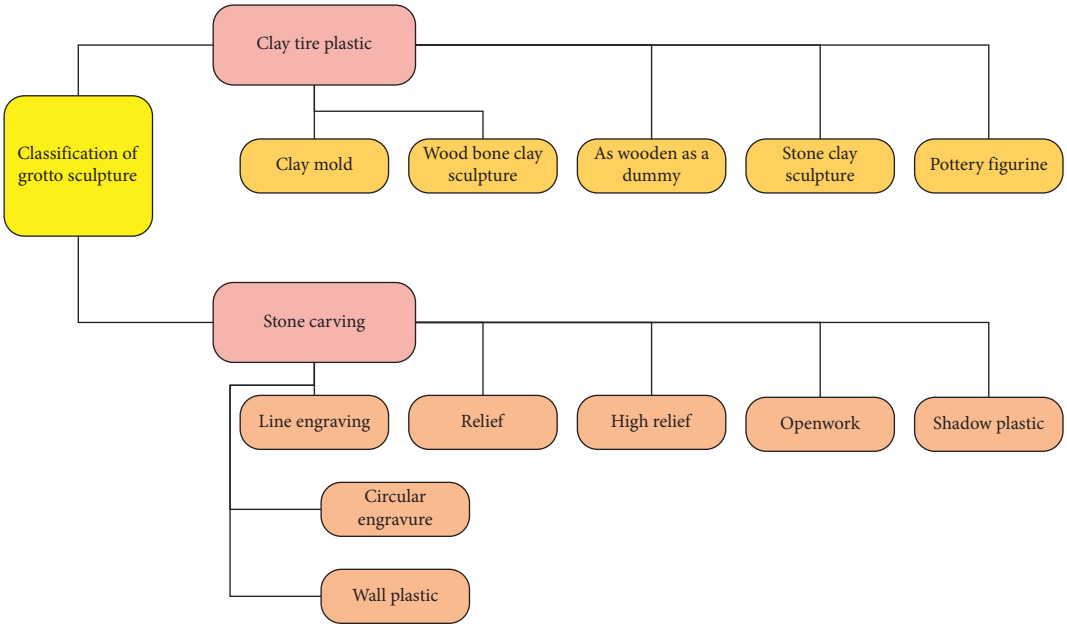


Figure 1: Main shapes of grotto murals.

constraints of the space environment and light conditions, the color structure and effect of murals in Mogao Grottoes are closely related to the grotto environment. According to the recorded stroke fragments, Singha and Manib [23] found that the colors used include black, yellow, bright red, vermilion, stone cyan, and stone green. Black accounts for the largest proportion and has high saturation. Figure 2 displays the main color proportions and classifications of the grotto murals.

Nishanthi [24] found that color impacts people's senses and psychology. The effect of Dazu Rock Carvings has both religious appeal and guiding effect. Qin and Song [25] pointed out that people used imperial cave statues in Dazu Rock Carvings to pray for blessings. The strong religious atmosphere created by the color of Dazu Rock Carvings makes believers feel immersive. In Buddhist art, red has the spiritual meaning of submission and appeal. Therefore, the colors of the grotto murals generally have two general meanings in Figure 3.

From the color art perspective, combined with religious culture, Chinese traditional culture, and field data, this paper summarizes and analyzes the color of Dazu Rock Carvings through the DL model, which lays a theoretical foundation for further displaying the color art of Dazu Rock Carvings.

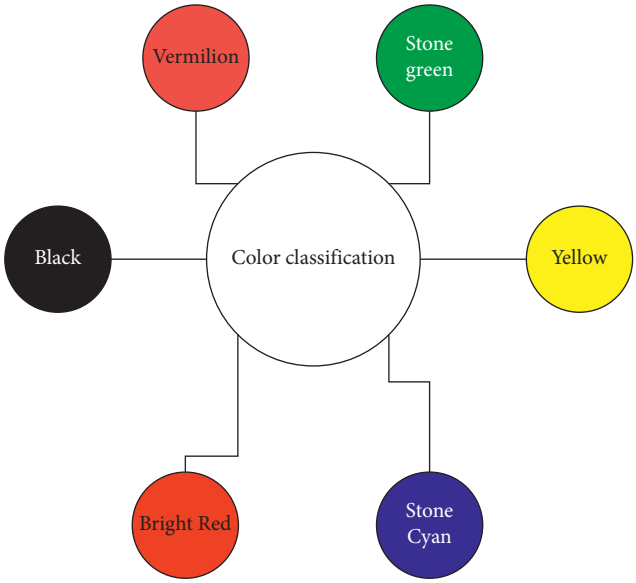


Figure 2: Main color classification of grotto murals.

3.2. Cultural Relics Information Acquisition Technology under DL. Han et al. [26] found that the work on the acquisition of two-dimensional cultural relics information had been carried out in worldwide cultural relics protection departments, transforming the pattern, color, and text information contained in cultural relics into digital information. After investigation, Xu et al. [27] found that the commonly used method was to take photos of cultural relics and obtain optical information using photography technology. Ordinary film can be used to take pictures and input them into the computer after shooting, or a digital camera can be directly used to obtain digital information. However, for multiple complex cultural relics, the research on how to shoot to ensure sufficient information is obtained and how to deal with the shooting information is not perfect.

Chang and Guo [28] found that an image-based rendering (IBR) technology was to generate scene images from different viewpoints based on some pregenerated images or environment maps. Compared with the traditional rendering technology, it has three distinct differences in Table 1.

Guan et al. [29] believed that, according to the graphics theory, the microscopic properties of the scene surface were

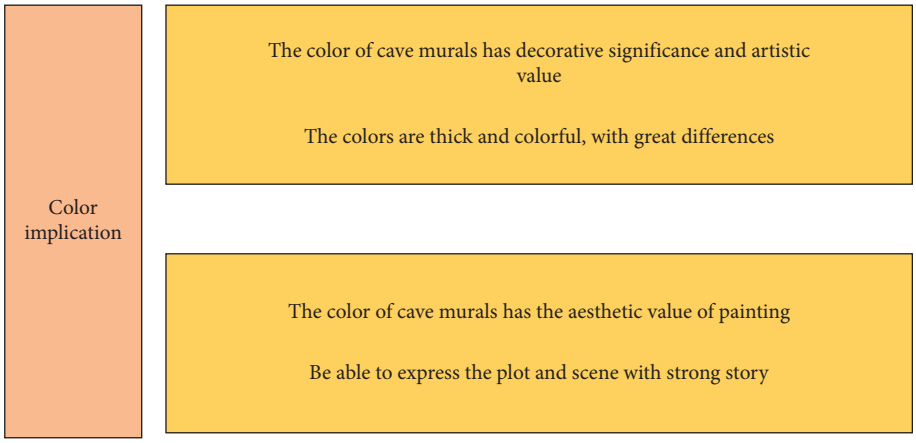


Figure 3: The color implication of grotto murals.

Table 1: The uniqueness of IBR technology.

The uniqueness of IBR technology
Images under IBR technology are generally drawn with separate scenes. The sharpness of the image depends on the scene's complexity.
The images adopted by IBR technology can be drawn manually or photographed on the spot and used comprehensively
IBR technology has a low computer load and can be applied and deployed on any multimedia device

finally reflected in the bidirectional reflectance at each point of the scene surface. The traditional realistic graphics rendering technology uses texture images to describe the reflective attributes of each point on the scene surface to simulate the rich texture details. IBR technology is based on this theory. Its core technology is to draw the scene with any viewpoint position using a group of images (specific viewpoint position) of related objects or environments. It uses these texture images to express the microinformation of the scene surface in the environment. IBR technology can be divided into three types in Figure 4:

Multiple models are stored in the form of multiresolution to meet the needs of real-time rendering or progressive transmission of network models. At this time, the multiresolution model representation needs to be compressed. Li et al. [30] used grids to generate regular or adaptive subdivision mechanisms, which could be used to generate models with multiple resolutions and provide an effective compression mechanism. However, they require that the input grid can meet the sub-subdivision connectivity.

Many grids cannot meet this requirement, so the three steps of image recording in Figure 5 are formulated.

Bashir et al. [31] believed that the visualization technology of adding time dimension information was a very crucial technology for multiple cultural relic visualization systems. For example, to demonstrate the damage or evolution process of cultural relics, or the evolution process of a site over time, time information needs to be combined with visual information.

3.3. VR Technology and Its Analysis and Operation in Grotto Murals. Bashabsheh et al. [16] pointed out that virtual image technology is usually a communication technology. It

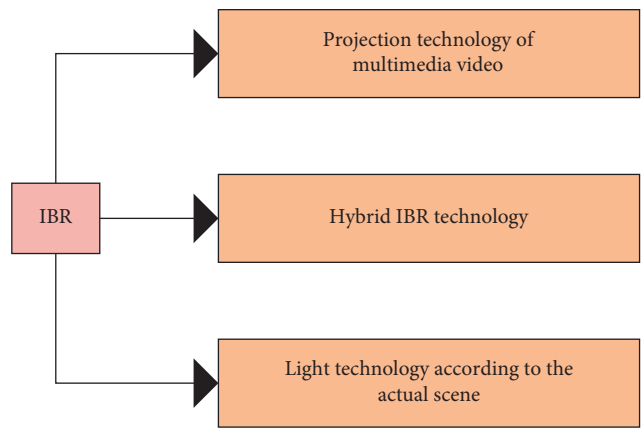


Figure 4: IBR technology classification.

is based on digital technology to combine vision and hearing. Virtual image technology includes various technologies, such as photography, projection, information transmission, multimedia display, sensory transmission, limb motion capture, and imitation. It provides users with virtual images through devices. Shin et al. [32] believed that virtual technology helps people understand abstract concepts, which are interactive images and scenes composed of text, images, videos, and other information. The main features are interactive, virtual, and immersive experiences. The rendering process of VR is analyzed. After collecting data elements in the image processing center, three-dimensional image modeling is conducted through sensors. Figure 6 shows the mapping process of VR technology.

Lungu et al. [33] found that the research on virtual image push algorithms in China is incomplete and not intelligent enough, the accurate push is difficult, and the processing

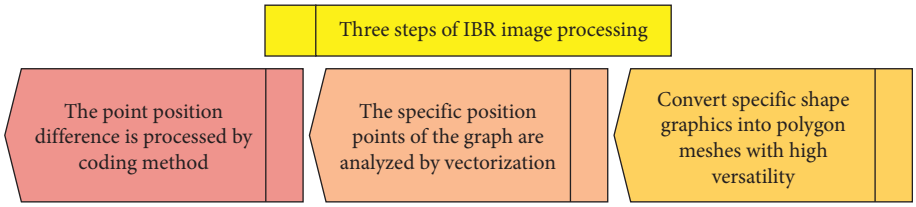


Figure 5: Three steps of IBR image processing.

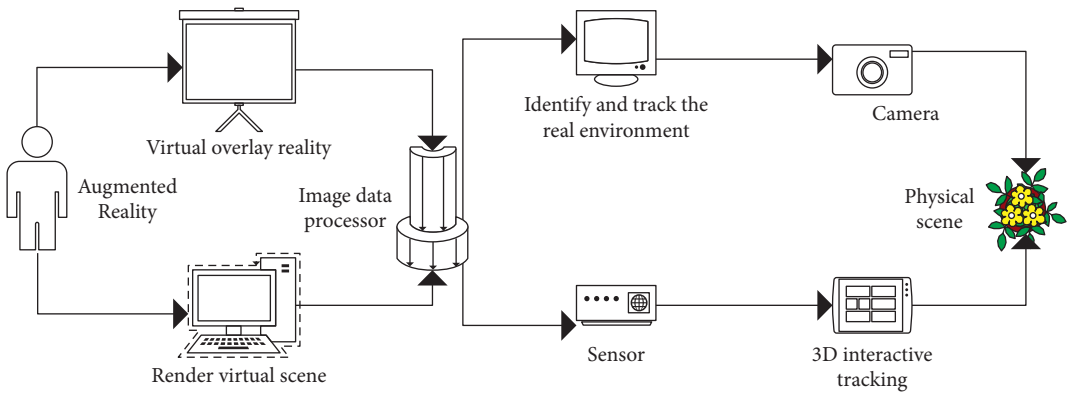


Figure 6: Mapping process of VR technology.

speed is low. There are lots of college students with different personal preferences. Many resources are wasted in massive virtual videos and images. Using a single algorithm cannot meet the needs of variability. Hence, a new hybrid recommendation algorithm is a feasible scheme to solve this problem.

The loss of resolution accuracy is caused in the splicing process, but the reason is in the shooting ring step. The main reasons for the loss are camera position error, lens aberration, projection distortion, and mural wall fluctuation. A complete operation step is calculated after considering these factors comprehensively. Liang et al. [34] revealed that, in image and video shooting calculation, $S_{\text{len-opti}}$ is generally adopted to represent the shooting pixel accuracy of the lens, dist means the distance, and len represents the lens type. Moreover, the fineness of the digital image must be determined before shooting. The digital simulation of grotto murals must have high resolution. However, too high precision and resolution will increase work difficulty and waste a lot of resources. Therefore, the minimum standard of resolution is to be able to clearly identify the lines on the mural. The specific situation needs to be judged according to the actual situation. If the inclination angle of the mural wall is θ and the final mural resolution is M, the shooting angle of the mural resolution is

$$PPI = \left(\frac{M}{\cos\theta}\right)^2. \tag{1}$$

When the maximum fluctuation angle of the mural wall does not exceed θ, the lens is LEN model, the elevation angle of the lens is defined as φ, and the resolution of the camera is $W \times H$, the distance L between the camera lens and the mural wall shall be

$$L = \frac{W\cos\theta}{2000M} ctg\frac{\phi}{2}. \tag{2}$$

At the last shooting, the horizontal or vertical moving distance L of the lens is calculated as follows:

$$L = \frac{S_{\text{len-opt}}(L, \text{Len})\cos\theta}{1000M}, \tag{3}$$

where $1000M$ represents the pixel size of the mural photographed by the camera. Many factors affecting murals' restoration cannot be accurately measured and calculated. This problem is effectively solved by obtaining lens distortion data through experiments and using the estimated value to determine the selection box to eliminate the influence of wall fluctuation. It also ensures the final results' accuracy and has high feasibility and practicability.

Lv et al. [35] believed that a general problem of uneven illuminationtook results in inconsistent brightness of the images. This situation can be improved by using the reflector to adjust the lighting direction of the flash lamp. However, it is still not ideal for preserving cultural relics.

The maximum light intensity is defined as $L_{\max}$. In a region, M points are adopted for calculation. If the nearest four points are L_0, L_1, L_2, and L_3, respectively, and point M is taken as the coordinate (x, y), the illumination intensity L of point M is

$$\begin{aligned} L = {} & \frac{(5-x)(5-y)}{100}L_0 + \frac{(5+x)(5-y)}{100}L_1 \\ & + \frac{(5-x)(5+y)}{100}L_2 + \frac{(5+x)(5+y)}{100}L_3. \end{aligned} \tag{4}$$

It is adjusted by using the light intensity L and the maximum light intensity L_{max}. For the most edge point, the brightness value LA is processed, and the new brightness value LB is

$$LB = LA \times \frac{L_{max} - L}{L_{max}}. \quad (5)$$

This adjustment method is efficient in processing speed and has little effect on the variation range of brightness. Color sculpture data acquisition includes three-dimensional structure data and surface texture data. These two data acquisition methods are different. The two-dimensional scanning method is adopted to directly obtain three-dimensional structure data of color sculpture. However, for the surface texture processing of color sculptures, the required data can be obtained only after field shooting.

3.4. Colour Analysis Scheme of Grotto Murals. When reconstructing murals with VR technology, it is essential to analyze the reasons for the rich colors of Dazu Rock Carvings, find out their comparison methods, and reproduce the original style of Dazu Rock Carvings as much as possible. Paakki et al. [36] found that Dazu Rock Carvings are rich in color due to the use of many complementary colors. Some studies mentioned that, from the perspective of chromatics, the contrast of complementary colors is the physiological and psychological response of human visual balance. Artists often use this visual instinct to apply complementary colors to the color structure of the image simultaneously or continuously in the process of color painting to enrich the image and reduce the visual fatigue of the viewer. In different cave murals, the structure and production materials of murals are different, which increases the color complexity of mural scenes and brings certain difficulties to mural protection. Hence, a reasonable scheme needs to be designed for the color analysis of the grotto murals.

The color of the Dazu Rock Carvings extracted from the scene is visible. Unlike the color displayed by computer monitors with operations, how to convert the color from reality to computer value is a difficult problem for this paper. There are two kinds of existing colors of Dazu Rock Carvings. First, the color is still visible; second, the color is missing in varying degrees. According to the above two situations, a comprehensive technical route is formulated through consulting massive literature and optical professionals to realize real and virtual color conversion. Figure 7 displays the specific technical route.

Figure 7 displays that the color analyzer further obtains color data by dividing light and color materials in mural materials, simulates and restores colors through VR technology, compares similar colors to form a color space, and fills the color image database. Shi et al. [37] proved that most of the pigments used in the color painting of grotto murals are mineral pigments, which have stable chemical properties and are not easy to change color. However, after a long history, the color change of the Dazu Rock Carvings is diversified. The same original color will have different degrees of color decay and falling off due to different distribution and influencing factors such as light, weathering, and humidity. Figure 8 presents the digital application process of cave murals established by VR technology and DL.

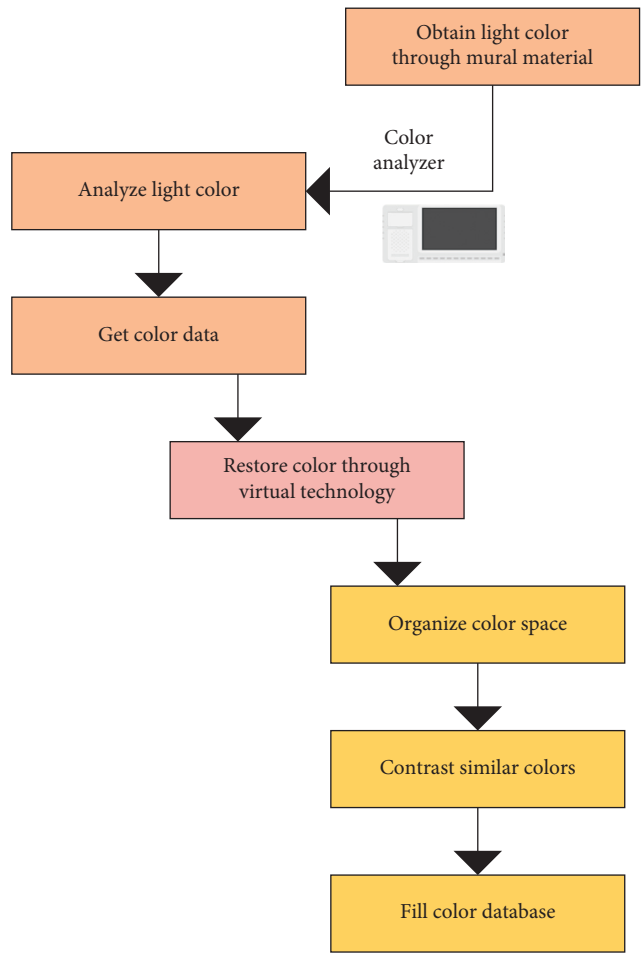


Figure 7: The digital color conversion process.

In the digital preview of grotto mural images, the principle of camera interactive control is to simulate the first-person perspective of human beings. Thereby, virtual characters can be set in virtual scenes. Moreover, the interactive control function of the camera can be realized by controlling the role perspective. Therefore, this digital simulation process will be used to analyze and identify the colors and materials of grotto murals. Murals in a cave are analyzed. Figure 9 presents the results.

Figure 9 displays that, for the images on the traditional grotto murals, first, the text is output through the sensor, then the mural repair and text extraction are carried out through the processor, and the digital simulation model is further established to analyze the mural text.

3.5. VR Technology and Artistic Modelling Analysis of Grotto Murals. The deep application of VR technology has expanded people's imagination and cognitive range. It can conduct highly simulated virtual reconstruction of the historical scenes of the grotto murals and build the scenes

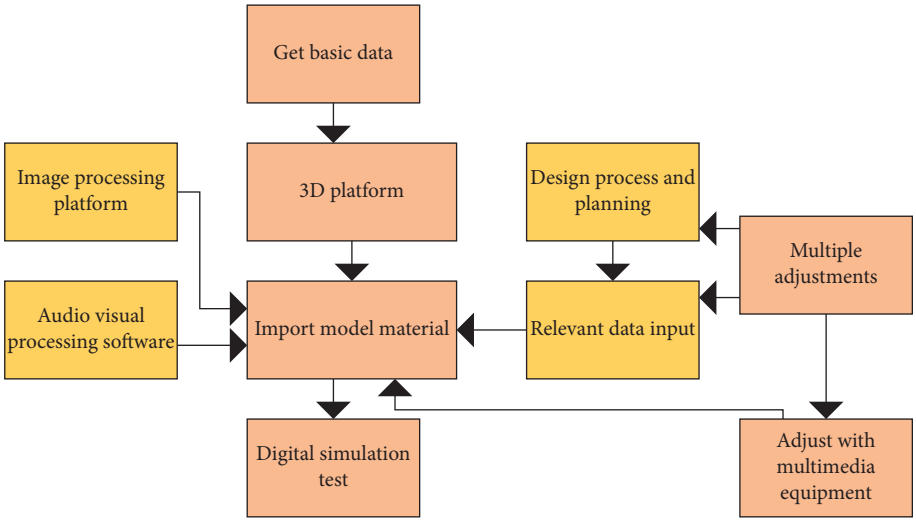


Figure 8: Digital simulation process of grotto murals.

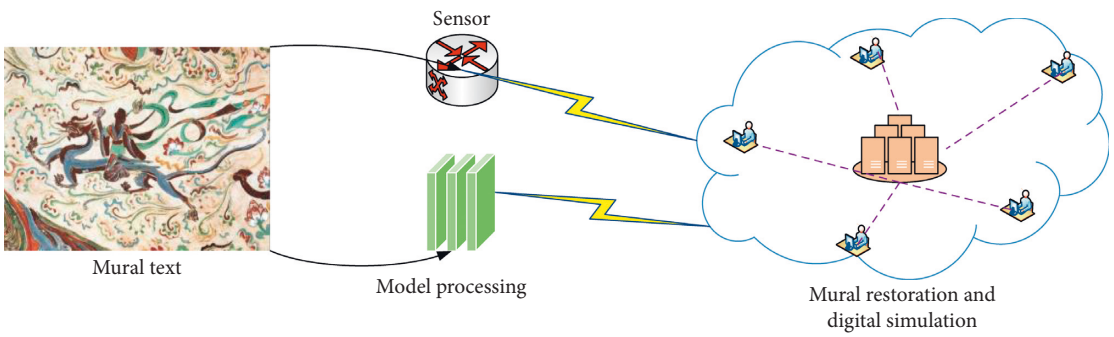


Figure 9: Mural text analysis and model establishment.

and elements that once existed only in the human brain or imagination one by one. Moreover, it allows people to travel as in the real world while maintaining all the perceptual states of the real world so that the audience can truly experience the scenes previously seen in sci-fi blockbusters. This high coincidence between reality and virtual scene enables people to enjoy and feel the cave mural culture at any time. No matter where they are, they can enter the virtual cave mural scene for appreciation. These precious historical and cultural heritages should be historical materials that people can read anytime and anywhere so that people can widely spread the historical and cultural spirit and focus on protecting the national treasure cultural relics.

The difference between VR technology and modern smartphones or other smart devices lies in whether it can present a perfect virtual world for people and whether people can get a new experience of infinite resources and space in this virtual world that cannot be felt and touched in real life. Maybe one day in the future, people can work in the virtual world and complete more work more relaxed and efficient. Moreover, they can even use completely different ways and tools from the current stage to maximize the liberation and improvement of productivity. The equipment used here is a handheld three-dimensional laser scanner, and the spatial measurement accuracy is 0.02 mm + 0.04 mm/m. The point and surface resolutions are 0.025 mm and 0.100 mm, respectively. The data transmission interface is universal serial bus 3.0, and the scanning range is 310 × 350 mm.

4. Results

4.1. Analysis of Colour Recognition Effect of Grotto Murals by the Digital Analysis Model. Many mineral pigments are adopted in the grotto murals. Moreover, with the years of sand erosion and man-made destruction, the color of many murals has significantly changed compared with the original color. Thereby, in the whole process of simulated restoration with VR technology, identifying the original color of grotto murals has become a basic work. In this section, the digital analysis model established above will be employed to identify some colors of grotto murals, to judge whether the model can accurately identify and verify colors and to prove the model's color recognition performance. There are two painting methods: stone carving and relief. Red, yellow, green, and black colors are selected for recognition and analysis. Figure 10 shows the results.

Figure 10 suggests that, in the system analysis results for red, the average recognition rate of stone carvings is between 80% and 90%, while the color recognition rate of relief carvings is in a steady rising state, with the highest point

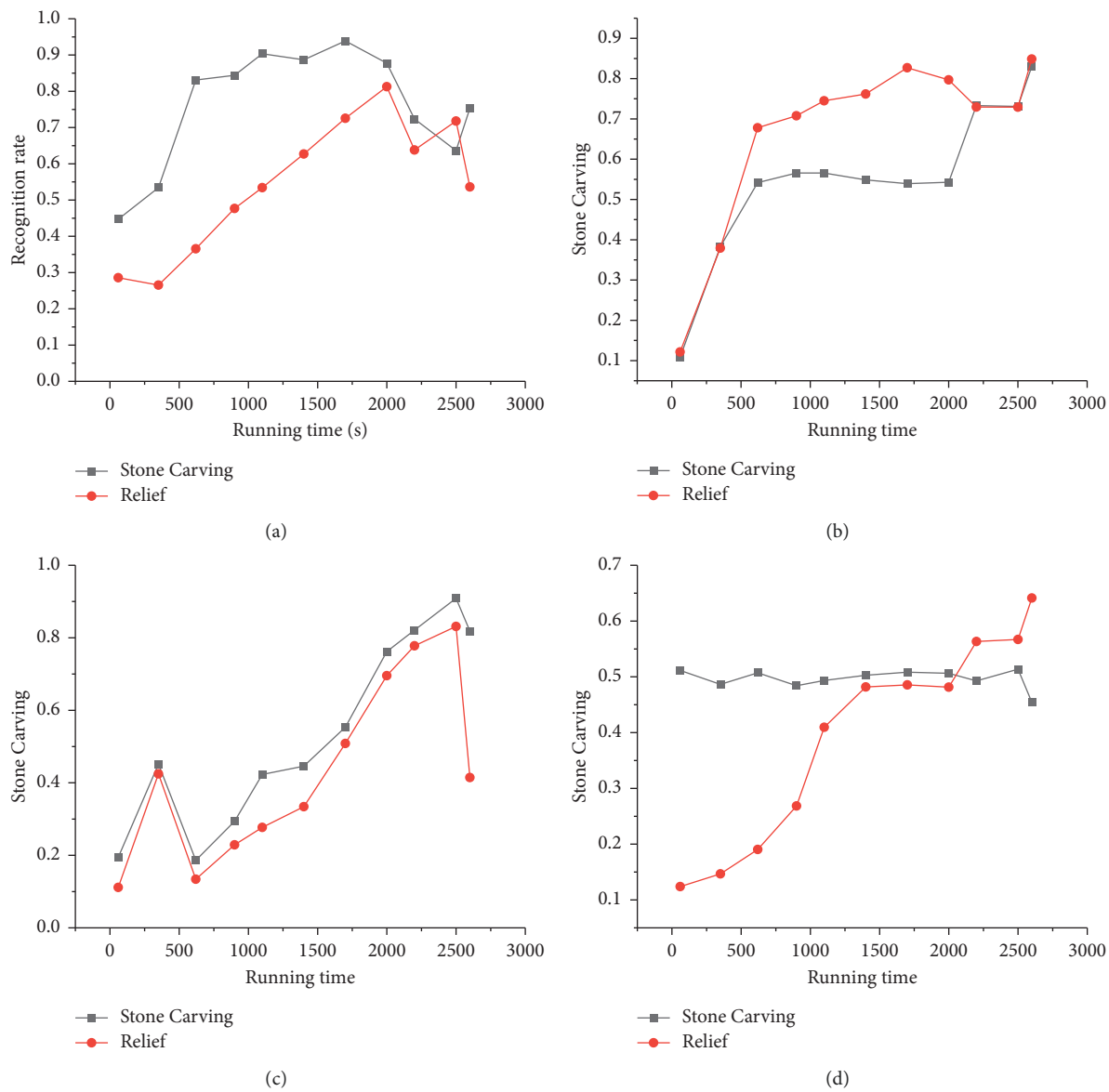


Figure 10: Color recognition results: (a) red, (b) yellow, (c) green, and (d) black.

reaching 80%. After running for more than 2250 s, the recognition rate begins to decline. In the recognition results of yellow, the recognition rates of stone carvings and relief carvings are rising at the beginning, and the best recognition rates of relief carvings and stone carvings are 85% and 90%, respectively. After 1600 seconds, the recognition rate of relief decreases, the overall trend is up, and the recognition results are satisfactory. In the green recognition results, the overall trend of the recognition rate is on the rise, and the highest recognition rates of relief and stone carvings are 80% and 90%, respectively. After 2500 s, the recognition rate of both begins to decline. In the first 500 s, the recognition rate data fluctuate in both cases. Finally, in black recognition, the recognition rate of stone carving is stable throughout the whole process, maintains at about 50%, and shows a downward trend after 2500 seconds. The recognition rate of relief is low at first, only 10%. With the system's running, the recognition rate of relief rises and reaches 65% at most. It reveals that the system is currently in an unstable state. The recognition rate of color is sometimes high and sometimes low, but the recognition rate of red, yellow, and green can reach more than 80%. The recognition rate can be accepted and used for actual detection. However, the recognition rate of black is not high. The possible reason is that all colors will change into various darker colors after being eroded and damaged, and the recognition interference of black is more obvious. Hence, optimizing the recognition rate of dark colors will be the main work in the follow-up research.

4.2. The Type of Color Material Recognized according to the Light of Color. Finally, the color of the grotto murals is verified and recognized using the digital recognition system. The results show that, except for black, the recognition rates

Table 2: Drawing material analysis results of specific colors.

Items	Recognition results						
Recognized color	Ocher	Stone green	Stone cyan	Gypsum	Graphite	Golden	Silver gold
Material type	Hematite	Malachite	Lapis lazuli	Calcium sulfate	Black lead powder	Phlogopite	Vermiculite

of red, yellow, and green can meet daily work needs. The low recognition rate of black may be due to the lack of rich colors, so black is not often used in murals. This section will use the proposed system to analyze color drawing materials. Specific materials need to be preserved in a specific way, so analyzing mural material categories is also crucial in preserving grotto murals. Tan et al. [38] analyzed the specific colors of model painting materials. Table 2 presents the results.

The results in Table 2 prove that the digital recognition model can clearly identify each unique color. Currently, the above seven colors are identified and analyzed. The identification material of the system for ochre is hematite, malachite for stone green, lapis lazuli for stone cyan, black lead powder for graphite color, phlogopite for golden, and vermiculite for silver gold. It reveals that the recognition accuracy of the color model is relatively accurate. It can accurately identify substances with a single color, which ensures the overall analysis of one-to-one recognition results, and the recognition accuracy of the model is high. However, given the low recognition rate of black in the previous section, the model needs to be further optimized to ensure that the recognition accuracy of other colors is also significantly improved.

5. Conclusion

Grotto murals are an art treasure of great value in China. They record bits and pieces of China's long history. Their historical value rises naturally since China is a large country with a long history. However, due to many years of wind and sand erosion of Gobi and desert and man-made unintentional or even intentional damage, the murals' patterns are incomplete and colors have faded. Thereby, a measure to repair the damage to grotto murals or preserve their complete appearance is urgently needed. However, the repair process is extremely complex and difficult, which will consume a lot of resources, and it is not optimistic after repair, so a method is needed to preserve it. According to the current advantages of DL technology and VR technology, this paper puts forward a way to preserve the murals by simulating and restoring the grotto murals based on VR technology. First, the grottoes' related concepts, materials, and colors are introduced. Then, the cultural relics information acquisition technology in the field of DL is put forward and combined with VR technology to build a digital recognition model for grotto murals. The results of color recognition of grottoes show that the model can accurately identify the colors of red, yellow, and green except black. Besides, it can identify the drawing material of each specific color one-to-one. It can be concluded that the digital recognition system can accurately record and simulate the restoration of grotto murals, which is of great significance for the preservation simulation of grotto murals using VR technology. However, the model's discrimination rate for black is currently low. Some reasons are summarized, and this problem will be mainly solved in the follow-up work.

Data Availability

The raw data supporting the conclusions of this article will be made available by the author without undue reservation.

Consent

Informed consent was obtained from all individual participants included in the study.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

The author acknowledges the help from the university colleagues.

References

[1] E. Chen, B. Zhang, and F. Zhao, "Comprehensive analysis of polychrome grotto relics: a case study of the paint layers from anyue, sichuan, China," *Analytical Letters*, vol. 53, no. 9, pp. 1455–1471, 2020.

[2] W. Ma, F. Wu, T. Tian et al., "Fungal diversity and its contribution to the biodeterioration of mural paintings in two 1700-year-old tombs of China," *International Biodeterioration & Biodegradation*, vol. 152, Article ID 104972, 2020.

[3] D. Stratford, K. Braun, and P. Morrissey, "Cave and rock shelter sediments of southern Africa: a review of the chronostratigraphic and palaeoenvironmental record from Marine Isotope Stage 6 to 1," *South African Journal of Geology*, vol. 124, no. 4, pp. 879–914, 2021.

[4] R. A. Armitage, R. Arrazcaeta, and S. Torres, "Chemical characterization and radiocarbon dating of the rock art of Las Charcas caves, Cuba," *Archaeometry*, vol. 63, no. 4, pp. 878–892, 2021.

[5] Y. Yin, Z. Yu, D. Sun et al., "A potential method to determine pigment particle size on ancient murals using laser induced breakdown spectroscopy and chemometric analysis," *Analytical Methods*, vol. 13, no. 11, pp. 1381–1391, 2021.

[6] X. He, J. Li, L. Tao, B. Zhang, Z. Fan, and B. Su, "Mechanisms of preservation damage: restoration materials affecting salt distribution and soil expansion in wall paintings of Dunhuang," *Archaeological and Anthropological Sciences*, vol. 11, no. 10, pp. 5171–5179, 2019.

[7] L. Liang, W. Zhang, L. Tan, and S. Chen, "Dust emission from Gobi under different dust content conditions: a wind tunnel study atop the Mogao grottoes," *Atmosphere*, vol. 12, no. 11, p. 1498, 2021.

[8] M. Bao, "Discussion on the painting in cave 254 of Mogao grottoes," *Journal of Contemporary Educational Research*, vol. 5, no. 10, pp. 40–46, 2021.

[9] M. Wu, A. Payshanbiev, Q. Zhao, and W. Yang, "Nonlinear optimization generating the tomb mural blocks by GANS," *Applied Mathematics and Nonlinear Sciences*, vol. 6, no. 1, pp. 43–56, 2021.

[10] L. Zhao, Z. Weng, and Y. Hu, "The exploration on digital restoration of Kizil Grottoes murals lost abroad," *Journal of Zhejiang University (Science Edition)*, vol. 47, no. 6, pp. 651–659, 2020.

[11] W. Bi, Z. Yan, and Z. Zhang, "Modeling and numerical simulation of heat and mass transfer in the cave wall of the Mogao Grottoes in China," *Building and Environment*, vol. 201, Article ID 108003, 2021.

[12] L. Zhou, C. Zhang, F. Liu, Z. Qiu, and Y. He, "Application of deep learning in food: a review," *Comprehensive Reviews in Food Science and Food Safety*, vol. 18, no. 6, pp. 1793–1811, 2019.

[13] J. van der Laak, G. Litjens, and F. Ciompi, "Deep learning in histopathology: the path to the clinic," *Nature Medicine*, vol. 27, no. 5, pp. 775–784, 2021.

[14] I. J. Jacob and P. E. Darney, "Design of deep learning algorithm for IoT application by image based recognition," *September 2021*, vol. 3, no. 3, pp. 276–290, 2021.

[15] M. Samadbeik, D. Yaaghobi, P. Bastani, S. Abhari, R. Rezaee, and A. Garavand, "The applications of virtual reality technology in medical groups teaching," *Journal of advances in medical education & professionalism*, vol. 6, no. 3, pp. 123–129, 2018.

[16] A. K. Bashabsheh, H. H. Alzoubi, and M. Z. Ali, "The application of virtual reality technology in architectural pedagogy for building constructions," *Alexandria Engineering Journal*, vol. 58, no. 2, pp. 713–723, 2019.

[17] K. Wang, H. Wang, M. Liu, X. Xing, and T. Han, "Survey on person re-identification based on deep learning," *CAAI Transactions on Intelligence Technology*, vol. 3, no. 4, pp. 219–227, 2018.

[18] S. Laxton, "Enchanted ground: andré Breton, modernism and the surrealist appraisal of fin-de-siècle painting by gavin Parkinson," *Modernism/Modernity*, vol. 26, no. 4, pp. 900–902, 2019.

[19] K. Li, A. Jain, A. Malovannaya, B. Wen, and B. Zhang, "DeepRescore: leveraging deep learning to improve peptide identification in immunopeptidomics," *Proteomics*, vol. 20, no. 21-22, Article ID 1900334, 2020.

[20] F. Farazmand, "The role of collective unconscious in the murals of altamira cave in Spain," *KURMANJ; The Journal of Culture, Humanities and Social Science*, vol. 1, no. 1, pp. 1–7, 2019.

[21] Y. Duan, F. Wu, D. He et al., "Diversity and spatial–temporal distribution of airborne fungi at the world culture heritage site Maijishan Grottoes in China," *Aerobiologia*, vol. 37, no. 4, pp. 681–694, 2021.

[22] X. Wang, Y. Wang, Q. Guo, Q. Pei, and G. Zhao, "The history of rescuing reinforcement and the preliminary study of preventive protection system for the cliff of Mogao Grottoes in Dunhuang, China," *Heritage Science*, vol. 9, no. 1, pp. 58–18, 2021.

[23] M. R. Singha and B. R. Manib, "Characterization of pigments and binders in mural painting fragments from bezeklik, China," *Indian Journal of History of Science*, vol. 54, pp. 348–360, 2019.

[24] M. Nishanthi, "PlAster and colour technology of ancient sri lAnKAn MurAls," *Archaeology*, vol. 1, no. 1, pp. 91–100, 2021.

[25] Z. Qin and Y. Song, "The sacred power of beauty: examining the perceptual effect of Buddhist symbols on happiness and life satisfaction in China," *International Journal of Environmental Research and Public Health*, vol. 17, no. 7, p. 2551, 2020.

[26] D. Han, L. Ma, S. Ma, and J. Zhang, "The digital restoration of painted patterns on the No. 2 Qin bronze chariot based on hyperspectral imaging," *Archaeometry*, vol. 62, no. 1, pp. 200–212, 2020.

[27] J. Xu, T. Zhang, Y. Jiang et al., "Preparation of self-healing acrylic copolymer composite coatings for application in protection of paper cultural relics," *Polymer Engineering & Science*, vol. 60, no. 2, pp. 288–296, 2020.

[28] Y. Chang and G. P. Wang, "A review on image-based rendering," *Virtual Reality & Intelligent Hardware*, vol. 1, no. 1, pp. 39–54, 2019.

[29] Y. Guan, X. Sang, S. Xing, Y. Li, and B. Yan, "Real-time rendering method of depth-image-based multiple reference views for integral imaging display," *IEEE Access*, vol. 7, Article ID 170545, 2019.

[30] S. Li, X. Han, and Y. Chang, "Adaptive cyclopean image-based stereoscopic image-quality assessment using ensemble learning," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2616–2624, 2019.

[31] T. Bashir, I. Usman, A. Albesher, K. A. Almejalli, and S. S. Naqvi, "GP based smart reversible watermarking of depth image-based rendering for stereoscopic images," *Multimedia Tools and Applications*, vol. 78, no. 15, Article ID 21943, 2019.

[32] H. Shin, D. Rim, H. Kim, S. Park, and S. Shon, "Educational characteristics of virtual simulation in nursing: an integrative review," *Clinical Simulation in Nursing*, vol. 37, pp. 18–28, 2019.

[33] A. J. Lungu, W. Swinkels, L. Claesen, P. Tu, J. Egger, and X. Chen, "A review on the applications of virtual reality, augmented reality and mixed reality in surgical simulation: an extension to different kinds of surgery," *Expert Review of Medical Devices*, vol. 18, no. 1, pp. 47–62, 2021.

[34] J. Liang, P. Wang, L. Zhu, and L. V. Wang, "Single-shot stereo-polarimetric compressed ultrafast photography for light-speed observation of high-dimensional optical transients with picosecond resolution," *Nature Communications*, vol. 11, no. 1, pp. 5252–5310, 2020.

[35] F. Lv, Y. Li, and F. Lu, "Attention guided low-light image enhancement with a large scale low-light simulation dataset," *International Journal of Computer Vision*, vol. 129, no. 7, pp. 2175–2193, 2021.

[36] M. Paakki, M. Sandell, and A. Hopia, "Visual attractiveness depends on colorfulness and color contrasts in mixed salads," *Food Quality and Preference*, vol. 76, pp. 81–90, 2019.

[37] Y. Shi, J. Qin, Y. Tao, G. Jie, and J. Wang, "Natural weathering severity of typical coastal environment on polystyrene: experiment and modeling," *Polymer Testing*, vol. 76, pp. 138–145, 2019.

[38] Y. Tan, M. Guo, Y. Hao, C. Zhang, and W. Song, "Structural parameter optimization for large spacing sublevel caving in chengchao iron mine," *Metals*, vol. 11, no. 10, p. 1619, 2021.

Hindawi
Scientific Programming
Volume 2022, Article ID 8543443, 6 pages
https://doi.org/10.1155/2022/8543443

Research Article

Theater Music Data Acquisition and Genre Recognition Using Edge Computing and Deep Brief Network

Xiaohua Wang,[1] **Lei Cheng**,[2] **Ding Cheng**,[3] **and Qinlin Zhou**[4]

[1]*College of Music, Qinghai Normal University, Xining 810016, China*
[2]*School of Art, Ludong University, Yantai, China*
[3]*School of Music and Dance, Qiqihar University, Qiqihar 161000, China*
[4]*School of Arts, Hunan University of Information Technology, Changsha 410000, China*

Correspondence should be addressed to Qinlin Zhou; 2019210426@mail.chzu.edu.cn

Received 15 March 2022; Revised 18 July 2022; Accepted 9 August 2022; Published 25 August 2022

Academic Editor: C. H. Wu

Artificial intelligence (AI) and the Internet of Things (IoT) make it urgent to push the frontier of AI to the network edge and release the potential of edge big data. The model's accuracy in data acquisition and music genre classification (MGC) is further improved based on theater music data acquisition. First, machine learning and AI algorithms are used to collect data on various devices and automatically identify music genres. The data collected by edge devices are safe and private, which shortens the time delay of data processing and response. In addition, the deep belief network (DBN)-based MGC algorithm has better overall recognition and classification effect on music genres. The MGC accuracy of the proposed improved DBN algorithm is nearly 80%, compared to 30%–40% of the traditional algorithms. The DBN algorithm is more accurate than the traditional classical algorithm in MGC. The research has an important reference value for developing Internet technology and establishing a music recognition model.

1. Introduction

Human-computer interaction (HCI) is increasingly frequent in the light of the Internet of Things (IoT). Combining high bandwidth, reliability, and data security gives birth to the intelligent edge computing (IEC) technology in the IoT field [1, 2]. IoT promotes geological space and information space integration. More people, machines, and things are connected to the information space, generating massive amounts of data, which has higher requirements for bandwidth and timely transmission [3, 4]. In particular, edge computing (EC) performs the computing tasks close to the data source to shorten the processing time. It speeds up data transmission, improves system reliability, and protects data security and privacy [5]. Meanwhile, IoT devices' demands for data collection are growing exponentially [6, 7]. Data transmission procedures and means of intelligent devices vary dramatically. Thus, the data acquisition of each kind of device needs targeted adaptation development, and the acquisition devices cannot be reused, resulting in the complexity of secondary development. The growth of data acquisition volume also increases transmission bandwidth, adding pressure on the cloud. Therefore, an adaptive data acquisition IoT gateway based on EC is proposed by enabling the edge IoT gateway to carry out data acquisition, analysis, and conversion on devices. As a result, the adaptability and universality of IoT gateway acquisition are enhanced [8].

In recent years, music information retrieval (MIR) and the music genre classification (MGC) have been concerned with the development of the Internet and digital audio technology. By now, MIR and MGC systems mainly extract music features manually and then train the classifier to establish the model to recognize and classify the test music samples. However, there is trouble in extracting music features manually. Different recognition and classification tasks require distinct music features; sometimes, the required music features cannot be named. As a new feature extraction technology, the DBN (deep belief network) makes

great achievements in image processing, natural language understanding (NLU), and other fields and has become increasingly mature.

The popularity of multimedia technology generates more online music works than ever. Classifying and managing them becomes a challenge. Many music users are only interested in specific music genres. MGC systems can divide music into different types according to style for users to choose, retrieve, and manage their favorite music. Indeed, MGC plays a crucial role in MIR. The main contribution of this work is to explore a new MGC approach through the ensemble classification method: the DBN-integrated edge data collection algorithm. The innovation is that the MGC effect of the DBN is compared and analyzed through experiments. In addition, studying MGC can promote the accuracy of theater music data acquisition, which is very necessary for modelling. This work will provide a reference for applying MGC algorithms in the future.

2. Recent Related Works

Some relevant methods of EC and heterogeneous data integration are cited. Salama et al. [9] studied the heterogeneous data integration method with active learning and evaluated the proposed model through the experiment. Five heterogeneous datasets from different fields were used: health reform dataset, Sander Frandsen dataset, financial phrase bank dataset, spam collection dataset, and textbook sales dataset. According to the results, the new data analysis method performed better than the traditional method. Anthony et al. [10] studied the digital development of virtual enterprises in the coronavirus disease 2019 (COVID-19). The results showed that during the COVID-19 pandemic, enterprises used virtual platforms to carry out business. Wewerka and Reichert [11] studied robot process automation and introduced the latest technology in robot process automation through system mapping research. Reine et al. [12] examined the return on investment (ROI) of software test automation (STA). The study provided a survey and analysis to understand the ROI of test automation by industry test professionals from product and service organizations. The results showed that the most commonly used method was the graphical user interface (GUI) test automation of functional scenarios to reduce manual testing and increase repeatability. To sum up, the accuracy of model recognition could be improved using a heterogeneous data integration method with active learning for music data acquisition and model training.

3. Methods

3.1. Data Acquisition of EC. EC is a hot term in recent years, data computing at the terminal close to the data source and completing the traditional cloud data processing task at the edge. EC and storage resource processing can alleviate network bandwidth overload and network delay [13, 14]. EC can be completed locally and can be operated on large, medium, and small equipment. EC devices are widely used, including computers, mobile phones, IoT intermediate nodes, smart homes, gateways, and even municipal terminals, such as automatic teller machine (ATM) and cameras [15]. EC renders local intelligent control services, intelligent data collection, data analysis, and intelligent industrial manufacturing that gradually lag. In the absence of cloud services, the corresponding research and design purposes of data collection based on EC and IoT can also be achieved [16, 17]. Traditional data processing is mainly carried out through cloud computing, and now, it is carried out at the edge, greatly reducing the pressure on the cloud. Of course, cooperative processing is also possible. Data collection or equipment monitoring can analyze data. Suppose the computing power of the device is close to the data source. In that case, it can be carried out at the edge to reduce the pressure on the cloud and draw conclusions quickly and efficiently. EC has become an important part of the current information infrastructure, and its future development is promising [18].

In the EC system, multiple data acquisition nodes complete edge devices' data acquisition. Each node is composed of an edge device and several external devices [19]. For physical data, data collection is done by external devices. Various devices collect data and transmit them to edge devices in real-time according to the predetermined format. As shown in Figure 1, edge devices provide various hardware interfaces, including universal asynchronous transceiver, universal serial bus, and wireless transmission interface. These functions increase the compatibility with more external devices. When multiple sensors send data to edge devices, each sensor uploads its data to edge devices through parallel transmission. The edge device must access the target web page through the network module, download the required data, and collect the network data [20]. The data exchange between edge devices and the cloud computing centre is completed through wireless transmission, with no need for other devices to collect data.

The traditional physical data acquisition module usually does not have the data analysis function but integrates and packages the data and leaves it to other related modules. Doing so increases the system computation and delay, affecting the timely response of data analysis results. Therefore, the reserved sensor interface can complete the sensor acquisition task by connecting the sensor with the edge device. The software is used to preprocess the collect data, forming the underlying data acquisition module. This can expand the application of the sensor, ensure the collected data are uploaded to the cloud in time, and shorten data processing delay [21]. The process is shown in Figure 2.

Generally, network data are acquired over web pages. The HyperText transfer protocol (HTTP)-based network application layer requires users to send requests to the Internet through the terminal modules. Here, the edge devices and the crawler technology are used together to connect to the Internet to collect the required data over the physical data collection method. Python is used to download the overall code of these web pages and complete the data collection. Network data collection is usually divided into two ways in light of their collecting order. For one case, the regular expression theory is used to extract the web address

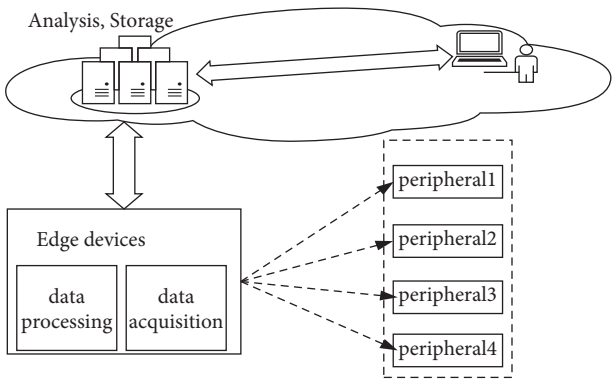


Figure 1: System design diagram.

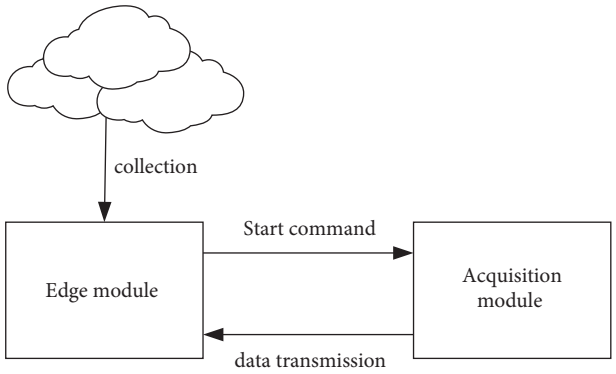


Figure 2: The physical data acquisition system.

related to the current query web page. The new web page is iteratively visited to extract network data until no hyperlink is available. The data extraction process is shown in Figure 3.

3.2. MGC Based on DBN. With the rapid development of Internet technology, online music services have gradually become the most convenient and main means for people to listen to music works. For massive music works, the performance of the MIR system is related to the quality of music services. Automatic MGC technology is an important part of content-based MIR, which has recently attracted much attention. Music signal has complex frequency composition and rich semantic information. The key to MGC lies in an effective music feature expression method. Here, EC technology is used to collect music-related information. The data are preprocessed and analyzed to get relevant information. Then, the music genre is identified and distinguished by DBN technology.

MGC is to identify the music information and genres for unknown songs or music. For example, George divides music into ten genres according to content: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. This classification plays a very important role in MIR [22]. Many music users are only interested in specific kinds of music, and the MGC system can classify music into different types according to style [23]. In this case, music can be recommended for users according to their interests, which is convenient for quick music information retrieval and

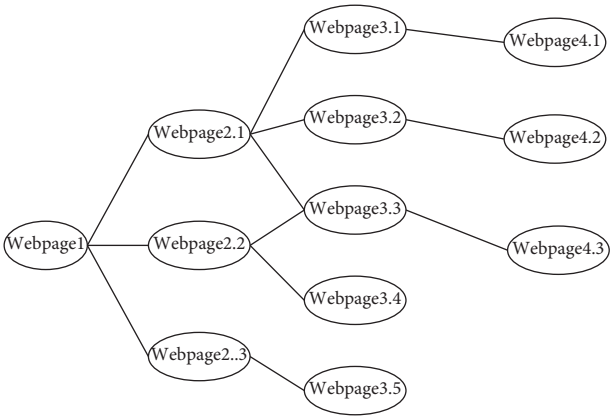


Figure 3: Web page-based network data extraction roadmap.

efficient management. Most music works are sung by people, accompanied by the performance of various musical instruments. In addition, the structural features of the music genre vary significantly. Even the same person can sing different sounds in different ranges when performing different music genres. These factors make it difficult for people to extract the features of music signals. Thus, improving the MGC accuracy is tricky. Recently, MGC has been widely concerned and developed rapidly. In particular, how to further improve the accuracy and efficiency of MGC has become the focus of the relevant research.

MGC often follows three steps: music signal preprocessing, music feature extraction, and music genre discrimination. Music signals must be preprocessed to facilitate feature extraction. Music features are another form of expression of the music signal with many redundancies being removed. Original music signals need a substantial amount of calculation [24]. Thus, the classifier must be trained to optimal the optimal model parameters to identify and classify music samples [25]. The traditional feature extraction method is very complex. Fortunately, DBN has autonomous learning characteristics and can be used to extract more abstract music features [26].

After preprocessing, the music samples' features must be extracted as many as possible according to the specific recognition and classification tasks. The extracted features will be used to train the classifier until the optimal model parameters are obtained to identify and classify the music samples with different genres or musical instruments. The classification flow is shown in Figure 4.

According to Figure 4, the key to MGC research is feature extraction and classifier design [27]. There is no unified standard for selecting feature quantity. To improve the accuracy of recognition and classification, some researchers start from the principle of signal generation to find new and effective features, while some fuse existing single feature quantities [28]. The above methods do improve the MGC accuracy in a sense. However, there is no universal MGC method and agreement on which music features to extract. Under the circumstance, manual feature extraction can hardly complete MGC tasks, and DL algorithms' advantage matters [29]. DBN can simulate the structure of the

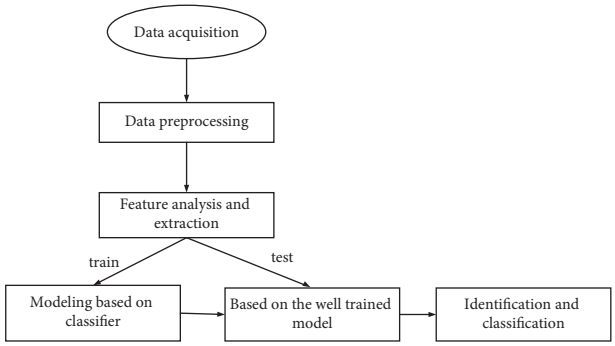


Figure 4: MGC flow.

human brain, store and process large amounts of information, and mine the internal correlation of data. That is, it extracts more essential data features to improve the performance of recognition and classification.

The specific steps of MGC are as follows: extract music features and label each music segment and use machine learning (ML) model to learn the relationship between music features and corresponding music genre labels [30]. Afterward, a classifier is generated through supervised training of music samples and applied to the MGC task. The classical classification methods include decision tree, K-nearest neighbor (KNN), support vector machine (SVM), and logistic regression [31].

3.3. Case Analysis and Testing. The traditional manual music feature extraction algorithm has encountered difficulties. The music features manually extracted using the above classical shallow classification method is directly recognized and classified. Its performance has not been improved. DL has made great progress as a new feature extraction technology in the MIR field. Therefore, DL is used to identify and classify western music genres. Here, DBN in DL is chosen further to learn the essential features of different music genres. SoftMax regression tests the music sample genres. This section takes genre music clips as samples, and the training set contains 1,600 samples of each genre, totaling 16,000 music genre samples. The validation and test sets contain 800 samples from each genre, and each genre has 8,000 music samples. First, all music genre samples are labeled, and the training model is trained on the sample set. In addition, the validation set is used for cross-validation. The trained model is used to predict music genre samples. Finally, the predicted music genre labels are compared with the actual music school labels to obtain the average MGC accuracy.

4. Results and Discussion

4.1. Comparison of Traditional Classification and DBN Algorithm. The classification algorithm based on DBN with SoftMax and the traditional classification method is used to train, verify, and test 10 music genres, respectively. The average accuracy is shown in Table 1.

Table 1 shows that different methods' MGC accuracy varies on the same music genre training set, the same validation set cross-validation, and the same test set. When the same feature model predictive control is input, the MGC accuracy of the traditional classification method is very low. The decision tree algorithm's accuracy is the lowest, only 33.5%. The highest accuracy goes to SVM, only 47.2%. By comparison, the MGC accuracy is significantly improved using the DBN algorithm without loss and momentum, as high as 70.9%. The MGC accuracy of the upgraded DBN algorithm is even 76.0% and over a 5% increase over the DBN algorithm. The first five groups of experimental results show that compared with traditional classification methods, DBN has better performance in MGC tasks. This is because DBN can independently learn and extract music features more suitable for MGC. The experimental results of the last two groups show that the upgraded DBN has a stronger feature extraction ability.

4.2. Recognition and Classification of Different Music Genres. The 480-dimensional MPC features of each music genre sample are input into various classification methods. The MGC accuracy of different algorithms on ten music genres is shown in Figure 5.

Figure 5 shows that the DBN algorithm has an excellent MGC accuracy on all music genres (over 50%) except for rock (just over 40%). These results are better than those of the DT classification. Meanwhile, the upgraded DBN algorithm is better than the traditional DBN algorithm for most music genres.

To sum up, after comparing the recognition matrix of the six classification methods, different classification methods have distinct recognition effects on the same music genre. Different classification methods' MGC accuracy on classical music is always the highest. Probably, there are obvious differences between the musical features of classical music and other types of music. The MGC accuracy for rock music is always very low. Possibly, there are similar features between rock music and other music genres, leading to the misjudgment of other types. Understanding music genres show that rock music overlaps with other music genres. This is because heavy metal and reggae music also belong to rock music. However, rock music belongs to pop music and is influenced by blues music, so it is easily recognized as other types of music. In addition, after using DBN for feature extraction, the MGC accuracy for each music genre has improved, further showing DBN's superiority in MGC. Ebel et al. [32] used data flow models and digital twins to design automation projects. They determined the degree of completion by considering the historical data using ML algorithms and currently uncompleted artifacts. The information was integrated from previous projects. The research helped realize the project progress measurement of automation engineering. Flechsig et al. [33] studied robot process automation in supply management. The results showed that robot process automation had attracted more attention in digital transformation. This cutting-edge technology automated robot behavior and had great potential. Therefore, the proposed MGC model can achieve a better music recognition effect.

Table 1: MGC accuracy of the proposed DBN with SoftMax and the traditional classification methods.

Method	DT (%)	KNN (%)	SVM (%)	SoftMax (%)	DBN combined with SoftMax (%)	Upgraded DBN combined with SoftMax (%)
Accuracy	33.5	44.3	47.2	46.8	70.9	76.0

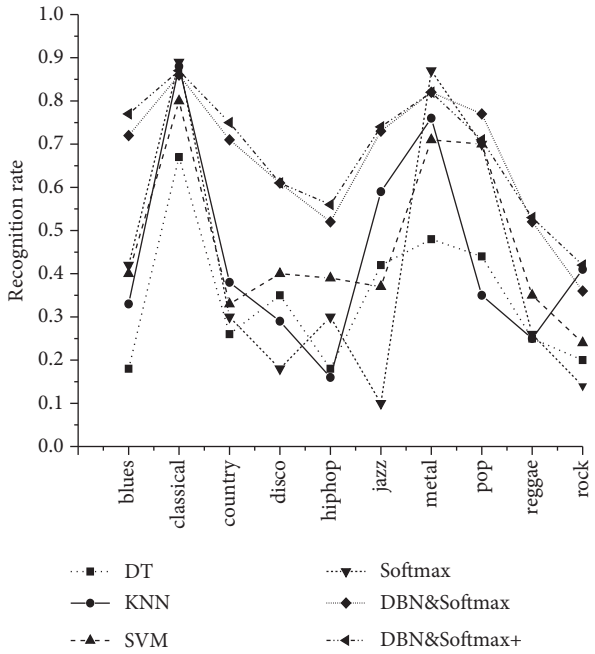


Figure 5: MGC accuracy for different music genres.

5. Conclusion

The originality statement is provided. According to the algorithm and its practical application, the data required by the system are collected by the edge device itself and related sensors. As such, the data can be used for subsequent analysis and processing. It has been successfully applied in many fields, such as music data acquisition. DBN is widely used in image processing but less in MIR. Compared with the classical algorithm, the proposed DBN with the SoftMax algorithm directly extracts music's acoustic features or musical features. The main contribution of this work is to train the classifier to get the classification results. While reducing the workload of manual extraction and recognition of classification features, the MGC accuracy of the proposed algorithm is also better than the classical algorithm. The proposed upgraded DBN algorithm improves the MGC accuracy and solves the problem of which features need to be extracted in manual classification and recognition. The experimental results also further verify the great value of DBN in music genre recognition and classification.

The shortcomings and prospects are stated as follows: compared with the classical recognition and classification methods, the DBN algorithm improves the average MGC accuracy by more than 20%. However, the overall accuracy is still low due to the music quality in the music database. The follow-up experiments can try to separate the song from its accompaniment and use pure song data for MGC. DBN builds the MGC system, involving many internal network parameters, such as the number of network layers, the number of neuron nodes, network optimization strategy, and the like. Subsequently, the parameters can be further optimized. Relatively, the experimental database is small regarding massive amounts of online music data. It is expected to expand the training samples and choose higher-performance hardware to handle MGC tasks.

Data Availability

The data used to support the conclusions of this study are available from the corresponding author upon request.

Consent

Informed consent was obtained from all individual participants included in the study.

Disclosure

This study was presented in the 2020 Chinese Control and Decision Conference (CCDC).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

References

[1] Z. Zhao, G. Min, W. Gao, Y. Wu, H. Duan, and Q. Ni, "Deploying edge computing nodes for large-scale IOT: a diversity aware approach," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3606–3614, 2018.

[2] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent edge computing for IOT-based energy management in smart cities," *IEEE Network*, vol. 33, no. 2, pp. 111–117, 2019.

[3] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-edge computing architecture: the role of MEC in the Internet of Things," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 84–91, 2016.

[4] D. Georgakopoulos, P. P. Jayaraman, M. Fazia, M. Villari, and R. Ranjan, "Internet of things and edge cloud computing roadmap for manufacturing," *IEEE Cloud Computing*, vol. 3, no. 4, pp. 66–73, 2016.

[5] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[6] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive transmission optimization in SDN-based industrial Internet of Things with edge computing," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1351–1360, 2018.

[7] Z. Sun, X. Zhang, T. Wang, and Z. Wang, "Edge computing in Internet of things: a novel sensing-data reconstruction algorithm under intelligent-migratoin stragegy," *IEEE Access*, vol. 8, Article ID 50696, 2020.

[8] C. H. Chen, M. Y. Lin, and C. C. Liu, "Edge computing gateway of the industrial Internet of things using multiple collaborative microcontrollers," *IEEE Network*, vol. 32, no. 1, pp. 24–32, 2018.

[9] M. Salama, H. Abdelkader, and A. Abdelwahab, "A novel ensemble approach for heterogeneous data with active learning," *International Journal of Engineering Business Management*, vol. 14, Article ID 184797902210826, 2022.

[10] B. Anthony and S. Abbas Petersen, "Examining the digitalisation of virtual enterprises amidst the COVID-19 pandemic: a systematic and meta-analysis," *Enterprise Information Systems*, vol. 15, no. 5, pp. 617–650, 2021.

[11] J. Wewerka and M. Reichert, "Robotic process automation-a systematic mapping study and classification framework," *Enterprise Information Systems*, pp. 1–38, 2021.

[12] D. Reine, S. Reanzi, and P. Ranjit Jeba Thangaiah, "A survey on software test automation return on investment, in organizations predominantly from Bengaluru, India," *International Journal of Engineering Business Management*, vol. 13, Article ID 18479790211062044, 2021.

[13] H. Li, K. Ota, and M. Dong, "Learning IOT in edge: deep learning for the Internet of Things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.

[14] S. U. Amin and M. S. Hossain, "Edge intelligence and Internet of things in healthcare: a survey," *IEEE Access*, vol. 9, pp. 45–59, 2021.

[15] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[16] S. Zhiyong, L. Weiping, Z. Yanghua, and H. Yanshan, "Research on intelligent power automation technology based on edge computing of power Internet of things," *E3S Web of Conferences*, vol. 204, no. 6, p. 2006, 2020.

[17] N. Ansari and X. Sun, "Mobile edge computing empowers Internet of things," *IEICE - Transactions on Communications*, vol. E101.B, no. 3, pp. 604–619, 2018.

[18] X. Zhang, Z. Cao, and W. Dong, "Overview of edge computing in the agricultural Internet of things: key technologies, applications, challenges," *IEEE Access*, vol. 8, Article ID 141748, 2020.

[19] P. Hu, W. Chen, C. He, Y. Li, and H. Ning, "Software-defined edge computing (SDEC): principle, open IOT system Architecture, applications, and challenges," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5934–5945, 2020.

[20] S. Agarwal, S. Yadav, and A. K. Yadav, "An efficient architecture and algorithm for resource provisioning in fog computing," *International Journal of Information Engineering and Electronic Business*, vol. 8, no. 1, pp. 48–61, 2016.

[21] W. Hu, Z. Wang, M. Ma, and L. F. Sun, "Edge video CDN: AWi-fi content hotspot solution," *Journal of Computer Science and Technology*, vol. 31, no. 6, pp. 1072–1086, 2016.

[22] S. Oramas, F. Barbieri, O. Nieto, and X. Serra, "Multimodal deep learning for music genre classification," *Transactions of the International Society for Music Information Retrieval*, vol. 1, no. 1, pp. 4–21, 2018.

[23] M. A. Ali and Z. A. Siddiqui, "Automatic music genres classification using machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 8, pp. 337–344, 2017.

[24] J. Lee, K. Yoon, D. Jang, S. J. Jang, S. Shin, and J. H. Kim, "Music recommendation system based on genre distance and user preference classification," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 5, pp. 1285–1292, 2018.

[25] W. Zou, "Design and application of incremental music recommendation system based on slope one algorithm," *Wireless Personal Communications*, vol. 102, no. 4, pp. 2785–2795, 2018.

[26] X. Geng, "Research on athlete's action recognition based on acceleration sensor and deep learning," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 2, pp. 2229–2240, 2021.

[27] Y. M. G. Costa, L. S. Oliveira, and C. N. Silla, "An evaluation of convolutional neural networks for music classification using spectrograms," *Applied Soft Computing*, vol. 52, pp. 28–38, 2017.

[28] D. Bisharad and R. H. Laskar, "Music genre recognition using convolutional recurrent neural network architecture," *Expert Systems*, vol. 36, no. 4, pp. 1–13, 2019.

[29] X. Li, W. Min, Q. Han, and R. Liu, "License plate location and recognition based on deep learning," *Journal of Computer Aided Design and Computer Graphics*, vol. 31, no. 6, pp. 979–987, 2019.

[30] J. Yoon, H. Lim, and D. W. Kim, "Music genre classification using feature subset search," *International Journal of Machine Learning and Computing*, vol. 6, no. 2, pp. 134–138, 2016.

[31] X. Shao and L. Yao, "Music classification based on SVM active learning," *Computer engineering and applications*, vol. 52, no. 6, pp. 127–133, 2016.

[32] H. Ebel, T. Riedelsheimer, and R. Stark, "Enabling automated engineering's project progress measurement by using data flow models and digital twins," *International Journal of Engineering Business Management*, vol. 13, Article ID 184797902110336, 2021.

[33] C. Flechsig, F. Anslinger, and R. Lasch, "Robotic Process Automation in purchasing and supply management: a multiple case study on potentials, barriers, and implementation," *Journal of Purchasing and Supply Management*, vol. 28, no. 1, Article ID 100718, 2022.

Hindawi
Scientific Programming
Volume 2022, Article ID 5984948, 12 pages
https://doi.org/10.1155/2022/5984948

Research Article

The Design of Urban Sculpture Space with User Behavior Based on Internet of Things and Edge Computing

Chili He

School of Architecture, South China University of Technology, Guangzhou 510641, China

Correspondence should be addressed to Chili He; arhcl@scut.edu.cn

Received 15 March 2022; Revised 18 May 2022; Accepted 31 May 2022; Published 8 July 2022

Academic Editor: Y. P. Tsang

To solve the application of Virtual Reality (VR) technology in the design of urban sculpture space, a VR network model is implemented based on Internet of Things (IoT) and Edge Computing. First, this work expounds and analyzes the theory of urban sculpture space and the interaction of urban sculpture space. Second, aiming at the problem that the development of IoT cannot meet the large amount of data and low delay of VR, a resource allocation algorithm, Joint Optimization of Resource Allocation and FOV (J-RAF) strategy is proposed based on the user interest and the resource allocation problem of (FOV), and simulation experiments are used to verify the effectiveness of the J-RAF algorithm. Finally, a VR network system is established based on cache cooperation and user interest, and a resource allocation algorithm Joint Cooperative Content Caching and FOV (J-CCF) is proposed, and its effectiveness is verified. Experiments show that compared with random strategy and random viewport strategy, algorithm J-CCF has 3% and 20% system gain. Compared with noncooperative cache scheme and random viewport strategy, algorithm J-CCF can improve the system gain by 15% and 25%, respectively. It is verified that the convergence of J-RAF algorithm and J-CCF algorithm and the effectiveness of improving the system gain. This work provides a technical basis for optimizing the design method of urban sculpture and contributes to the optimal design of urban space.

1. Introduction

With the development of the Internet of Things (IoT), a variety of new technologies emerge in endlessly. The traditional urban sculpture can no longer meet people's functional needs for urban sculpture space [1]. The combination of IoT technology and urban sculpture space has greatly improved the interactivity compared with traditional urban sculpture [2]. Urban sculpture, using IoT technology pays more attention to users' feelings and experience, improves the interactivity and interest of urban sculpture space and places and endows users with better spiritual experience [3].

Scholars have done a lot of research on the combination of IoT technology and edge computing, new technology, and urban sculpture space. Du [4] proposed that the development of a city should not only have the urban image with the characteristics of the times but also make full use of historical and cultural resources, retain strong local characteristics and historical context, and form a unique urban personality. Urban sculpture is the best form of expression and the "eye of the city," which shows the spiritual outlook of the city and becomes the business card of a city. A good urban sculpture embodies not only the beauty of form but also the guiding sign of a city's spiritual culture. Urban sculpture is also one of the important elements of urban landscape environment. Qiu et al. [5] proposed that with the development of science and technology, more and more science and technology played an important role in urban construction, and IoT technology, as the current mainstream science and technology, had been developed in various industries. Therefore, through IoT technology, urban sculpture is developed, urban design is planned, more reasonable landscape elements are established for the city, and it plays an important role in urban development. Wang et al. [6] proposed a new model to collect reliable data based on the edge computing of the IoT. The nodes were evaluated from multiple dimensions to obtain accurate trust values. By mapping the trust value of the node to the mobile data collector, the best mobile path with high trust was generated.

Jha et al. [7] proposed a novel simulator Internet of Things simulation-Edge (IoTSim-Edge), which captured the behavior of heterogeneous IoT and edge computing infrastructure, and allowed users to test their infrastructure and framework in a simple and configurable way. IoTsim edge extends CloudSim's capabilities to include different features of edge and IoT devices. The effectiveness of IoTsim edge is described using three test cases. To sum up, the current science and technology have made great progress in the IoT, edge computing, and the combination of the two, and the IoT technology has also played a great role in urban design. However, the concept of urban design based on the combination of the IoT and edge computing has not been developed, so more research is needed to integrate the IoT and edge computing. The development of urban sculpture space and place design is an innovative research, which will also make more contributions to urban development.

First, based on the IoT and edge computing combined with user behavior, this work puts forward the combination of Virtual Reality (VR) and urban sculpture space. Additionally, according to the application requirements of VR technology in urban sculpture space, a VR network system is established based on user interest and Field Of View (FOV). Finally, a VR network system is established based on joint cache cooperation and user interest, which proves the effectiveness of the two algorithms for VR technology. The innovation is to use IoT and edge computing technology to design the application model of VR technology in urban planning, comprehensively analyze its application effect, and innovate the technical methods of urban planning. The proposed algorithm has a good reference for the combination of VR technology and urban sculpture and improves the interactivity and experience of urban sculpture space based on user behavior.

2. Method

2.1. An Overview of Theory of Urban Sculpture Space. Urban sculpture is a sculpture work standing in urban public places. In a city with many tall buildings and vertical and horizontal roads, it can alleviate the congestion, rigidity, and singleness caused by the concentration of buildings and can increase the balance on the open space [8]. According to the theory of urban sculpture space, a certain space is designed according to urban sculpture, and enough sculpture is designed to beautify the city. This design space is called urban sculpture space [9]. There are obvious differences between the current urban sculpture and the traditional sculpture in terms of concept and function. Urban sculpture is mainly aimed at the public and has publicity. The design of sculpture needs to be considered from the following aspects, as shown in Figure 1.

Nowadays, with the development of the IoT, the combination of the IoT and urban sculpture is more in line with modern needs [10]. The combination of digital technology and urban sculpture breaks the form of traditional static sculpture. Through combining them, a variety of urban sculptures appear, and the presented forms are more colorful. The urban sculpture is also more life-like and popular,

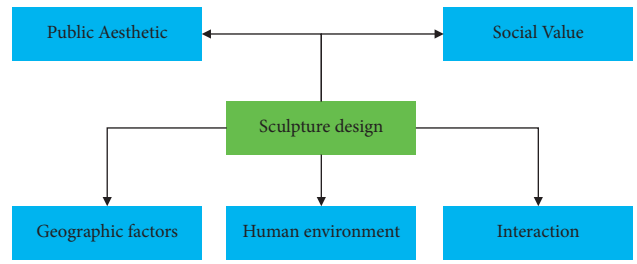


Figure 1: Framework of design considerations for urban sculpture.

and architecture and installation art can be displayed in the form of sculpture; any landscape and landmark also can be presented in the form of sculpture, and all small street sculptures to large architectural landscapes are all modern urban sculptures [11]. The places of urban sculpture space are studied, which focuses on the design of the places. These also emphasize the participation of users, serving users, and classifying them in terms of functions by the behavior of users and the characteristics of the space, as shown in Figure 2.

One of the important attributes of urban sculpture is its spatial attribute. The sculpture is not only a work of art but can also convey the corresponding information of the urban and space to the public. The artistic effect of urban sculpture and the interaction of the spatial environment complement each other. The sculpture not only needs to consider various specific environmental factors of the environment in its design but also needs to coordinate with the spatial environment. The sculpture itself also affects the environment in which it is located and improves the spatial environmental quality of the place [12]. The interactive sculpture in urban sculpture can improve the function of the space, increase the vitality of the space, and enhance the spirit of the space.

2.2. A New Technology of Interactive Urban Sculpture. With the development of IoT and digital technology, the combination of multimedia technology and art came into being, allowing people to participate in sculpture art and get a variety of different feelings. This open art mainly highlights the experience and feeling brought by the work itself, so it is also a challenge to the traditional sculpture art [13]. Traditional sculpture art combines digital technology and IoT technology, changing from static existence to dynamic existence. There are many ways to combine digital technology with urban sculpture. Usually, IoT technology is used to design software, and electronic devices are added to the urban sculpture itself. For example, using 3D modeling, augmented reality (AR) or VR and other IoT technologies, these technologies show the colorful and vibrant urban sculpture art [14]. As one of the main IoT technologies in the art reform, the current three-dimensional digital art has played an important role in urban sculpture. Its fundamental connotation is three-dimensional design, which is the basis of a new generation of digital, virtual, and intelligent design platform. It is a new design method based on plane and two-dimensional design, which makes the design goal more three-dimensional and visual.

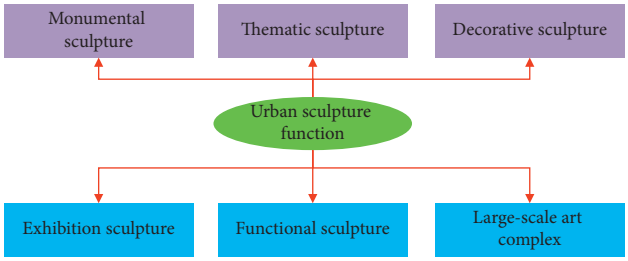


Figure 2: Classification of functions of urban sculpture space.

The traditional urban sculpture is to place the completed sculpture directly on the site and integrate the sculpture and space. Works are less designed from the perspective of user behavior. For users, the feeling given by urban sculpture is a single psychological feeling of looking up [15]. Users are just admirers of urban sculpture. The interaction between users and urban sculpture is poor, and lack interest and the single form of interaction, which cannot make users become real participants [16]. The combination of the IoT technology and urban sculpture greatly improves the interactivity compared with traditional urban sculpture. Urban sculpture using the IoT technology pays more attention to users' feelings and experience, narrowing the distance between users and sculpture art [17]. The combination of the IoT technology and urban sculpture is complementary, and the two cannot be replaced each other. The IoT technology can help urban sculpture better improve its interest and interactivity. Combining the characteristics of urban sculpture with new technologies can better provide users with spiritual experience [18].

2.3. VR Network System Model Based on User Interest and FOV. With the development of the IoT and the upgrading of related equipment, VR technology can be well combined with urban sculpture in urban sculpture spaces. It can create an interesting and interactive space for users [19]. In the urban sculpture space, if VR technology is to bring a good experience to users, the problem is that it needs to meet the requirements of low latency and consume a lot of network resources. These are also the problems that the current VR technology needs to be improved [20]. Mobile edge computing (MEC) can provide good help for the pressure of network resources brought by VR. MEC servers are deployed at the edge of the network, which can improve the computational capability and data caching capability of the network, reduce user latency, and reduce network congestion [21]. Song et al. [22] used mobile edge computing (MEC) to extract FOV videos from 360-degree videos to avoid transmission bandwidth occupation and backhaul link traffic between the base station and the core network. The cache placement and FOV selection of wireless VR service network are studied. Joint content caching and FOV selection are expressed as an optimization problem, which aims to maximize the utility function of the system under the existing network resources. For multivariable-coupled nonconvex problems, a joint optimization algorithm of resource allocation and FOV is proposed, and the algorithm is evaluated by Taylor expansion and successive convex optimization. The results show that the joint optimization algorithm of resource allocation and FOV can effectively meet the delay period and maximize the system revenue under the efficient use of resources. The specific network framework is shown in Figure 3.

In Figure 3, there are M cells, N users in each cell, and H types of videos in the system. According to the visual characteristics of the human eye, the FOV video is transmitted to the users. The viewport size when user n requests h in the M cell is $f_{m,n,h} \in \{\theta_1, \ldots, \theta_k\}$ in which $f_{m,n,h}$ is the viewport angle. When multiple different versions of the same type of FOV video are stored in the MEC, it will take up too much storage space. Therefore, the FOV video is stored in the MEC in a format of 360°, and the optimized strategy is used to process it as a transmission of FOV video [23].

When a user sends a service request to the base station, different revenue settings are performed by the system for videos with different viewport angles. If it is found that the video requested by the user's service is in the cache of the MEC, the MEC can directly send the video from the cache to the user, which can save the transmission bandwidth, and the resulting gain is shown in equations (1)–(3).

$$G_{m,n,h} = \varnothing * \frac{f_{m,n,h}}{360^{\circ}}, \tag{1}$$

$$G_{\text{reward}} = \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{h=1}^{H} \rho_{m,n,h} G_{m,n,h}, \tag{2}$$

$$G_{\text{cach}} = \alpha \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{h=1}^{H} \rho_{m,n,h} X_{m,h} r^{\text{mec}}. \tag{3}$$

$G_{m,n,h}$ user n brings benefits to video h in cell m. $\varnothing$ the weight coefficient of return. $\rho_{m,n,h}$ the popularity video. $X_{m,h}$ cache Set. α unity gain of saving bandwidth. r^{mec} transmission rate of sending video content. There is always a cache overhead when caching video in the MEC. Even if the user does not request it, it will occupy the cache space and consume the cache. The 360° video is stored in the system, which needs to be processed by the MEC, and the FOV video requested by the user is extracted from the 360° video and sent to the user. If the cache of the video is not requested in the MEC, and the video is fetched from the core network. There are two processing schemes. One is that the core network performs video processing, and the 360° video is processed as FOV video and sent. The other is that the 360° video is sent first, and then the cell performs video processing and processes it into FOV video. To save transmission resources and reduce latency, the first scheme is adopted. The specific equation is shown in (4)–(7):

$$C_{\text{cach}} = \delta S_{360} \sum_{m=1}^{M} \sum_{h=1}^{H} X_{m,h}, \tag{4}$$

$$C^{1}_{\text{compute}} = \beta \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{h=1}^{H} \rho_{m,n,h} z^{\text{mec}}_{m,n,h}, \tag{5}$$

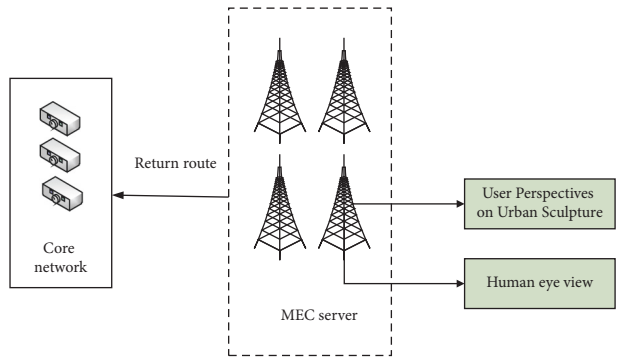


Figure 3: A model of VR network based on user interest and FOV.

$$C^2_{\text{compute}} = \beta \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{h=1}^{H} \rho_{m,n,h} z^{\text{core}}_{m,n,h}, \tag{6}$$

$$C_{\text{compute}} = \beta \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{h=1}^{H} \left(\rho_{m,n,h} X_{m,h} z^{\text{mec}}_{m,n,h} + \rho_{m,n,h}\left(1 - X_{m,h}\right) z^{\text{mec}}_{m,n,h}\right), \tag{7}$$

δ is the unity overhead of storage space. S_360 is the size of cache 360°video. $z^{mec}_{m,n,h}$ is the overhead of MEC calculation. β is the unit calculation cost. $z^{core}_{m,n,h}$ is the core network resources of consuming. $z^{core}_{m,n,h} = \eta * f_{m,n,h}/360^{\circ}$. When the video is transmitted to the user, the video cache is requested by the user in the MEC. The MEC directly provides the user with the content. When the requested video cache is not in the MEC, the content is sent from the core network to the base station, and the base station sends it to the user. The specific equation is shown in (8)–(11):

$$C_1 = \alpha \rho_{m,n,h} r^{\text{user}}_{m,n,h}, \tag{8}$$

$$r^{\text{user}}_{m,n,h} = \sum_{k=1}^{K} B^{\text{user}}_k * \log_2\left(1 + \frac{p_m g_{m,n,k} X_{m,n,k}}{\sum_{m^*=1, m^* \neq m}^{M} \sum_{n=1}^{N} p_m * g_{m^*,n,k} X_{m,n,k} + \sigma^2}\right), \tag{9}$$

$$C_2 = \alpha \rho_{m,n,h} r^{user}_{m,n,h} + \alpha \rho_{m,n,h} r^{mec}, \tag{10}$$

$$C_{\text{transmission}} = \alpha \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{h=1}^{H} \left(\rho_{m,n,h} z^{\text{user}}_{m,n,h} + \rho_{m,n,h}\left(1 - X_{m,h}\right) r^{\text{mec}}\right). \tag{11}$$

$r^{\text{user}}_{m,n,h}$ is the MEC user transmission rate in the cell. B^{user}_k is the channel bandwidth of user downlink. σ^2 is the additive white Gaussian noise. $X_{m,n,k}$: whether the channel is occupied. If the requested content is not in the MEC, the core network needs to process video, and the processing time is shown in equations (12)–(14):

$$T^{up}_{m,n,h} = A + \left(1 - X_{m,h}\right) B, \tag{12}$$

$$T_3 = \frac{S_{m,n,h}}{R_m}, \tag{13}$$

$$T_4 = \frac{S_{m,n,h}}{R_c}. \tag{14}$$

A is the requested time of uplink. B is the requested time of base station uplink to the core network. $S_{m,n,h}$ is the size of the requested video. $1/R_m$ is the MEC processing time of 1 bit. $1/R_c$ is the core network processing time of 1 bit. According to the above equation, when the user sends the request, the processing time is calculated. The time when the MEC is sent to the user, the time when the core network is sent to the base station, and the time of complete transmission delay are shown in equations (15)–(19):

$$T^{\text{process}}_{m,n,h} = X_{m,h} \frac{S_{m,n,h}}{R_m} + \left(1 - X_{m,h}\right) \frac{S_{m,n,h}}{R_c}, \tag{15}$$

$$T_5 = \frac{S_{m,n,h}}{r^{\text{user}}_{m,n,h}}, \tag{16}$$

$$T_6 = \frac{S_{m,n,h}}{r^{mec}}, \tag{17}$$

$$T^{\text{transmission}}_{m,n,h} = \frac{S_{m,n,h}}{r^{user}_{m,n,h}} + \left(1 - X_{m,h}\right) \frac{S_{m,n,h}}{r^{mec}}, \tag{18}$$

$$\begin{aligned} T_{m,n,h}^{\text{total}} &= T_{m,n,h}^{up} + T_{m,n,h}^{\text{process}} + T_{m,n,h}^{\text{transmission}} \\ &= +\left(1 - X_{m,h}\right)B + X_{m,h}\frac{S_{m,n,h}}{R_m} + \frac{S_{m,n,h}}{r_{m,n,h}^{\text{user}}} \\ &+\left(1 - X_{m,h}\right)\frac{S_{m,n,h}}{R_c} + \left(1 - X_{m,h}\right)\frac{S_{m,n,h}}{r^{mec}} \end{aligned} . \quad (19)$$

By optimizing the cache scheme of MEC and the cell channel allocation of the user's viewport strategy, the situation of optimally allocating resources is shown in equation (20):

$$\begin{aligned} &\max_{\{f_{m,n,h}\}\{X_{m,h}\}\{X_{m,n,k}\}} U, \\ &s.t. D1\colon A + \left(1 - X_{m,h}\right)B + X_{m,h}\frac{S_{m,n,h}}{R_m} + \frac{S_{m,n,h}}{r_{m,n,h}^{\text{user}}} \\ &\quad + \left(1 - X_{m,h}\right)\frac{S_{m,n,h}}{R_c} + \left(1 - X_{m,h}\right) \\ &\quad \cdot \frac{S_{m,n,h}}{r^{mec}} \le T_D, \quad \forall m \in M, \quad \forall n \in N, \\ &D2\colon X_{m,h} \in \{0,1\}, \quad \forall m, h, \\ &D3\colon \sum_{h=1}^{H} X_{m,h} S_{360} \le S_m, \quad \forall m \in M, \\ &D4\colon X_{m,n,k} \in \{0,1\}, \quad \forall m, n, k, \\ &D5\colon \sum_{n=1}^{N}\sum_{k=1}^{K} X_{m,n,k} = K, \quad \forall m, \\ &D6\colon \sum_{n=1}^{N} X_{m,n,k} = 1, \quad \forall m, k \\ &D7\colon f_{m,n,h} \in \{\theta_1, \theta_2 \ldots, \theta_k\}, \quad \forall m, n, h. \end{aligned} \quad (20)$$

In the equation, $D1$ is the user's maximum tolerable delay. $D2$ is the cache constraint. $D3$ is the maximum cache capacity limit of MEC. $D4$ is the channel constraint. $D5$ and $D6$ are the guarantees that the channel is only occupied once, $D7$ is the viewport constraint of FOV. $D2$, $D4$, and $D7$ are integer constraints. $D1$ is a nonconvex constraint, which is a problem of an nonconvex optimization.

2.4. The Solving Algorithm of Problems of VR Network System Model Based on User Interest and FOV. An efficient iterative algorithm is proposed by using variable relaxation, Lagrangian decomposition and other methods. The three subproblems are alternately iterated to obtain the optimal scheme of resource allocation. First, the placement method of the MEC cache is optimized, as shown in equation (21):

$$\begin{aligned} p1\colon &\max_{\{X_{m,h}\}} \alpha \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{h=1}^{H} \rho_{m,n,h} X_{m,h} r^{\text{mec}} \\ &- \alpha \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{h=1}^{H}\left(\rho_{m,n,h}\left(1 - X_{m,h}\right)r^{\text{mec}}\right) \\ s.t. &- \delta S_{360}\sum_{m=1}^{M}\sum_{h=1}^{H} X_{m,h} - \beta \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{h=1}^{H} \\ &\cdot \left(\rho_{m,n,h} X_{m,h} z_{m,n,h}^{\text{mec}} + \rho_{m,n,h}\left(1 - X_{m,h}\right) z_{m,n,h}^{\text{core}}\right) D1 - D3, \end{aligned} \quad (21)$$

$p1$ is the problem of standard linear programming. $D7$are the constraints of convex multiple discrete-value. Finally, the method of cell channel allocation is optimized. The use function of system efficiency and the objective function, etc., are shown in equations (22)–(23):

$$U = \alpha \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{h=1}^{H} \rho_{m,n,h} \sum_{k=1}^{K} B_k^{\text{user}} * \log_2\left(\frac{\sum_{m'=1,m' \ne m}^{m}\sum_{n=1}^{N} p_m \cdot g_{m',n,k} X_{m,n,k} + \sigma^2}{\sum_{m'=1,m' \ne m}^{M}\sum_{n=1}^{N} p_m \cdot g_{m',n,k} X_{m,n,k} + \sigma^2 + p_m g_{m,n,k} X_{m,n,k}}\right), \quad (22)$$

$$\begin{aligned} U = \alpha \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{h=1}^{H} \rho_{m,n,h} \sum_{k=1}^{K} B_k^{\text{user}} &\left[\log_2\left(\sum_{m'=1,m' \ne m}^{M}\sum_{n=1}^{N} p_{m'} g_{m',n,k} X_{m,n,k} + \sigma^2\right)\right. \\ &\left. - \log_2\left(\sum_{m'=1,m' \ne m}^{M}\sum_{n=1}^{N} p_m' g_{m',n,k} X_{m,n,k} + \sigma^2 + p_m g_{m,n,k} X_{m,n,k}\right)\right]. \end{aligned} \quad (23)$$

U is the objective function. $X_{m,n,k}$ is the variable. t is number of iterations; $p3$ is the problem of convex programming. The joint placement cache and joint optimization of resource allocation and FOV strategy (J-RAF) are proposed. The computational complexity is $\mathcal{O}(\tau(MH)^2 + (MH) + (MNH))$.

2.5. VR Network System by the Joint Cache Cooperation and User Interest. In the VR wireless network, the panoramic video service has the characteristics of high data, low latency, and many viewing angles [24]. The main research is to use the MEC-assisted VR wireless network to maximize the efficiency of each part of the system, and a resource allocation optimization algorithm is proposed to solve it. In the specific system model, the number of cells is set to M, the number of users in a single cell is set to N, the video category is set to H in the system, and the user viewport is set to $f_{m,n,h}, f_{m,n,h} \in \{\theta_1, \ldots, \theta_k\}$. The specific system model is shown in Figure 4.

The user first sends a request to the base station, and the system is proportional to the income according to the size of the FOV video. The income of the cell and the total income of the system are shown in equations (24)–(27). The cache can save the bandwidth of the backhaul link, and this part of the bandwidth is called caching gain. There is also caching gain in the transmission of user-requested video and adjacent cooperative cells. However, if the video in the cache is not the video requested by the user, these videos will also occupy resources of system, and the total overhead of system is as shown in equations (24)–(27):

$$G_{m,n,h} = \varnothing^* \frac{f_{m,n,h}}{360^\circ}, \tag{24}$$

$$G_{\text{reward}} = \sum_{n=1}^{N} \sum_{h=1}^{H} p_m(h) G_{m,n,h}, \tag{25}$$

$$G_{\text{cach}} = \alpha \sum_{n=1}^{N} \sum_{h=1}^{H} P_m(h) X_{\text{m},h} r^{\text{mec}}, \tag{26}$$

$$G_{\text{cach}} = \delta S_{360} \sum_{m=1}^{N} \sum_{h=1}^{H} X_{\text{m},h}. \tag{27}$$

$\varnothing$ is the gain coefficient, $p_m(h)$ is the distribution of content popularity, α is the unity gain, $X_{\text{m},h}$ is the cache set, r^{mec} is the transmission rate is sent by the core network to the cell. δ is the unit cache overhead, S_{360} is the size of 360° video. The videos of all the users finally receive FOV, and the system needs to process and calculate 360° videos, and the consumption of system resources is proportional to the size of the processed FOV videos. If mmc1 are the adjacent cells that cannot cooperate, or the user requests that the content is not cached, the first method is that the core network can only process the video as FOV video, or the core network directly sends the video to the cell for processing by the cell itself. The second method can save some transmission bandwidth and reduce latency. The processed content is transmitted to the user, and there are two transmission overheads corresponding to the two scenarios, and there are a cache of requested videos in the MEC and no requested video in the MEC of the cell where it is located. However, there is a cache in the adjacent cooperative cells, and the adjacent cells can transfer cooperatively. The specific transmission overhead is shown in equation (28):

$$\begin{aligned}
C_1 &= \alpha P_{m,h} r^{\text{user}}_{m,n,h} \\
C_2 &= \alpha P_{m,h} r^{\text{user}}_{m,n,h} + \alpha P_m(h) \\
&\quad \cdot \left(1 - X_{m,h}\right) \sum_{m^* \neq m} X_{m^*,h} Y(h) r^{\text{cell}} \\
C_3 &= \alpha P_{m,h} r^{\text{user}}_{m,n,h} + \alpha P_m(h) \\
&\quad \cdot \left(1 - X_{m,h}\right)\left(1 - \sum_{m^* \neq m} X_{m^*,h}\right) r^{\text{core}} \\
&\quad + \alpha P_m(h)\left(1 - X_{m,h}\right) \\
&\quad \cdot \sum_{m^* \neq m} X_{m^*,h} (1 - Y(h)) r^{\text{core}} \\
C_{\text{transmission}} &= \sum_N \sum_H \alpha P_m(h) r^{\text{user}} + \sum_N \sum_H \alpha P_m(h) \\
&\quad \cdot \left(1 - X_{m,h}\right) \sum_{m^* \neq m} X_{m^*,h} Y(h) r^{\text{cell}} \\
&\quad + \sum_N \sum_H \alpha P_m(h)\left(1 - X_{m,h}\right) \\
&\quad \cdot \sum_{m^* \neq m} X_{m^*} (1 - Y(h)) r^{\text{core}} \\
&\quad + \sum_N \sum_H \alpha P_m(h)\left(1 - X_{m,h}\right)(1 - Y(h)) r^{\text{core}}.
\end{aligned} \tag{28}$$

β is unit processing cost. $z^{mec}_{m,n,h}$ is MEC computing processing overhead. $z^{core}_{m,n,h}$ is the computing processing overhead of the core network. $r^{\text{user}}_{m,n,h}$ is the transmission rate of user. r^{cell} is the cell transmission rate. r^{core} is the transmission rate from core network to cell. When users use VR, the network needs to meet the requirements of low latency, and it is necessary to calculate the delay of each link. When the content requested by the user is not in the MEC, the video is processed through the core network. The processing delay and the transmission delay from the MEC to the user are shown in equation (29):

$$\begin{aligned}
T_4 &= \frac{S_{m,n,h}}{R_c} T^{\text{process}}_{m,n,h} \\
&= X_{m,h} \frac{S_{m,n,h}}{R_m} + \left(1 - X_{m,h}\right) \frac{S_{m,n,h}}{R_m} \sum_{m^* \neq m} X_{m^*,h} Y(h) \\
&\quad + \left(1 - X_{m,h}\right) \frac{S_{m,n,h}}{R_c} \sum_{m^* \neq m} X_{m^*,h} (1 - Y(h)) \\
&\quad + \left(1 - X_{m,h}\right) \frac{S_{m,n,h}}{R_c} \left(1 - \sum_{m^* \neq m} X_{m^*,h}\right) T_5 \\
&= \frac{S_{m,n,h}}{r^{\text{user}}_{m,n,h}}.
\end{aligned} \tag{29}$$

$S_{m,n,h}$ is the size of the requested video. $1/R_m$ is MEC processing time of 1 bit。 $1/R_c$ is the core network processing time of 1 bit. $T^{process}_{m,n,h}$ is the processing delay of service calculation. If the content is in adjacent cells, the cells can transfer

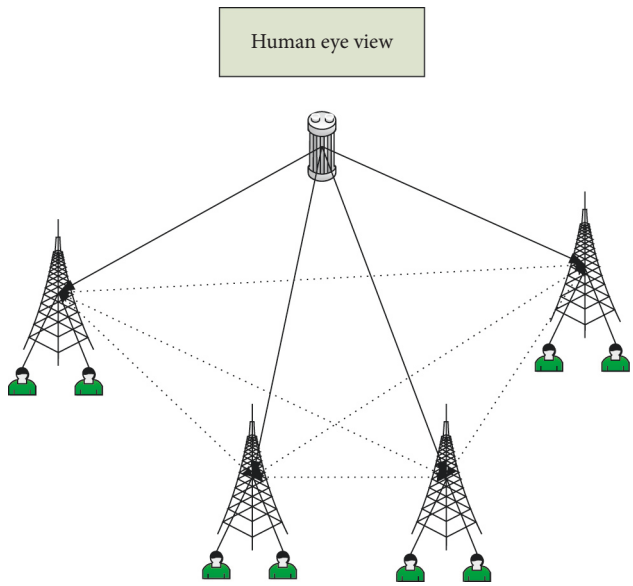


Figure 4: The diagram of VR Network System by the joint cache cooperation and user interest.

cooperatively. If the adjacent cells cannot transfer cooperatively, they need to be transmitted from the core network. The two transmission delays are shown in equation (30):

$$\begin{aligned} T_6 &= \frac{S_{m,n,h}}{r^{\text{cell}}} + T_{\text{queue}} T_7 \\ &= \frac{S_{m,n,h}}{r^{\text{core}}} T_{m,n,h}^{\text{process}} \\ &= \frac{S_{m,n,h}}{r_{m,n,h}^{\text{user}}} + \left(1 - X_{m,h}\right)\left(1 - \sum_{m^* \neq m} X_{m^*,h}\right)\frac{S_{m,n,h}}{r^{\text{core}}} \\ &\quad + \left(1 - X_{m,h}\right)\left(\frac{S_{m,n,h}}{r^{\text{cell}}} + T_{\text{queue}}\right)\sum_{m^* \neq m} X_{m^*,h} Y \\ &\quad + \left(1 - X_{m,h}\right)\frac{S_{m,n,h}}{r^{\text{core}}}\sum_{m^* \neq m} X_{m^*,h}(1 - Y) T_{m,n,h}^{\text{total}} \\ &= T_{m,n,h}^{\text{up}} + T_{m,n,h}^{\text{process}} + T_{m,n,h}^{\text{transmission}}. \end{aligned} \tag{30}$$

The system utility function of the VR network system and of joint cache cooperation and user interest is shown in equation (31) and the problem of optimization is shown in equation (32). $\{X_{m,h}\}$ is MEC cache placement, $\{f_{m,n,k}\}$ is the viewport allocation of user. $\{Y_h\}$is the strategy of cooperative cache.

$$U = G_{\text{reward}} + G_{\text{cach}} - C_{\text{cach}} - C_{\text{transmission}} - C_{\text{compute}}, \tag{31}$$

$$\begin{aligned} &\max_{\{f_{m,n,h}\}\{X_{m,h}\}\{X_{m,n,k}\}} U \\ &D1\colon T_{m,n,h}^{\text{total}} \leq T_D, \quad \forall m \in M, \quad \forall n \in N, \\ &D2\colon X_{m,h} \in \{0,1\}, \quad \forall m, h, \\ &D3\colon \sum_{h=1}^{H} X_{m,h} S_{360} \leq S_m, \quad \forall m \in M, \\ &D4\colon Y_h \in \{0,1\}, \quad \forall h, \\ &D5\colon f_{m,n,h} \in \{\theta_1, \theta_2 \ldots, \theta_k\}, \quad \forall m, n, h, \\ &D6\colon \sum_{m^* \neq m} X_{m^*,h} \in \{0,1\}. \end{aligned} \tag{32}$$

$D1$ is the maximum delay of gratification of the user, which needs to ensure a good experience for users. $D2$ is the strategy of MEC cache prevention. $D3$ is the limit of MEC maximum cache capacity. $D4$ is the cooperative transmission strategy of adjacent cells. $D6$ is to select one of the cells of caching multiple requested content for cooperative transmission. $D1$ is a nonconvex constraint, which is a problem of nonconvex optimization.

2.6. The Solving Algorithm of Problems of VR Network System Based on Joint Cache Cooperation and User Interest. The problem of nonconvex optimization is divided into three methods to solve. The first method is the optimization of the MEC cache placement, as shown in equation (33). Through the transformation and deformation, the equation (34) can be obtained. Finally, the optimization method of the cache is obtained by solving, as the input of the next question.

$$\begin{aligned} p2\colon \max_{\{X_{m,h}\}} \; & \alpha \sum_{n=1}^{N}\sum_{h=1}^{H} P_m(h) X_{m,h} r^{\text{mec}} - \delta S_{360} \sum_{h=1}^{H} X_{m,h} \\ & - \left[\begin{aligned} & \beta \sum_{n=1}^{N}\sum_{h=1}^{H} P_m(h) z_{m,n,h}^{\text{mec}} + \beta \sum_{n=1}^{N}\sum_{h=1}^{H} P_m(h)\left(1 - X_{m,h}\right) z_{m,n,h}^{\text{mec}} \sum_{m^* \neq m} X_{m^*,h} Y(h) + \beta \sum_{n=1}^{N}\sum_{h=1}^{H} P_m(h)\left(1 - X_{m,h}\right) z_{m,n,h}^{\text{core}} \sum_{m^* \neq m} X_{m^*,h} Y(h)\left(1 - \sum_{m^* \neq m} X_{m^*,h} Y(h)\right) \\ & + \alpha \sum_{n=1}^{N}\sum_{h=1}^{H} P_m(h)\left(1 - X_{m,h}\right) \sum_{m^* \neq m} X_{m^*,h} Y(h) r^{\text{cell}} + \sum_{n=1}^{N}\sum_{h=1}^{H} P_m(h)\left(1 - X_{m,h}\right)\left(1 - \sum_{m^* \neq m} X_{m^*,h} Y(h)\right) r^{\text{core}} \end{aligned} \right] \end{aligned} \tag{33}$$

$$f_1(x)f_2(x) = \frac{(f_1(x)+f_2(x))^2}{2} - \frac{f_1(x)^2+f_2(x)^2}{2}(f_1(x)+f_2(x))^2 \geq (f_1^r(x)+f_2^r(x))^2 + 2(f_1^r(x)+f_2^r(x))(f_1(x)-f_1^r(x)) + 2(f_1^r(x)+f_2^r(x))(f_2(x)-f_2^r(x)). \tag{34}$$

The second is to fix the MEC cache settings and cooperative transfer method to transform the original optimization problem $p2$ into a tractable problem $p3$, as shown in equation (35). D5 is the convex multidiscrete value constraint. The third is to give the cache settings viewport allocation method of users, which can convert $p3$ to $p4$, as shown in equation (36):

$$\begin{aligned} &\max Q(t)W_{\text{coo}} \sum_{h=1}^{H}\sum_{m}^{M} p_m(h)\left(1-X_{m,h}\right)S_{360}f_{m,h}Y_{m,h}\sum_{m^*\neq m}X_{m^*,h} \\ &+ Q(t)W_{\text{bh}}\sum_{h=1}^{H}\sum_{m=1}^{M} p_m(h)S_{360}f_{m,h}\left(1-X_{m,h}\right)\left(1-Y_{m,h}\sum_{m^{\circ}\neq m}X_{m^*,h}\right) s.t. D4, \end{aligned} \tag{35}$$

$$\begin{aligned} &\min Q(t)W_{\text{coo}} \sum_{h=1}^{H}\sum_{m}^{M} p_m(h)\left(1-X_{m,h}\right)S_{360}f_{m,h}Y_{m,h}\sum_{m^*\neq m}X_{m^*,h} \\ &- Q(t)W_{\text{bh}}\sum_{h=1}^{H}\sum_{m=1}^{M} p_m(h)S_{360}f_{m,h}Y_{m,h}\sum_{m^*\neq m}X_{m^*,h} s.t. D3. \end{aligned} \tag{36}$$

$D3$ is binary variable constraints. According to the above three methods, the MEC cache setting, FOV allocation of user, and cooperative caching strategy are optimized and solved. The Binary Code Decimal (BCD) algorithm is used to propose an algorithm of resource allocation, Joint Cooperative Content Caching and FOV (J-CCF), that combines MEC cache setting, FOV allocation of user, and cooperative transmission optimization. Cooperative Content Caching and FOV, J-CCF, and the computational complexity of J-CCF algorithm are $\mathcal{O}(\tau(MH)^2 + (MH) + (MH))$.

3. Result

3.1. An Analysis of Experimental Results of VR Network System Using User Interest and FOV. The relationship between the VR network system of user interest and FOV and the number of iterations, and between the efficiency function and the number of users are shown in Figure 5.

Figure 5(a) shows the relationship between the system efficiency of the VR network and FOV and the number of iterations. It is obvious that the J-RAF algorithm has the best performance among the three comparisons between the J-RAF algorithm, the random cache strategy, and the random viewport strategy. Moreover, the J-RAF algorithm becomes convergent after the seventh time, the speed of convergence is fast, and the performance is good. The proposed J-RAF algorithm also has the best performance. Figure 5(b) indicates the relationship between system return and the number of users. As the number of users increases, the efficiency function of the system also increases gradually, and the increase becomes more and more gentle. The relationship between system benefit and the cache space, and between the different cache spaces and user delay times of user are shown in Figure 6.

Figure 6(a) shows the relationship between the size of the cache space and the system benefit. Comparing the J-RAF algorithm, the random cache strategy and the random viewport strategy, the three methods are different. Compared with the other two schemes, the J-CCF algorithm has different performance; the gains are 3% and 20% respectively. However, when the cache space continues to increase, the benefit growth of the system gradually slows down. In Figure 6(b), it is found that the user delay time of the J-RAF scheme and the random viewport scheme increases with the increase of the cache space, mainly because the increase of the cache space can provide users with more services and resources, resulting increase in the delay time. The user delay time of the random caching scheme decreases as the cache space increases, and the delay time decreases because the video does not need to be sent from the core network.

3.2. A Comparative Analysis of VR Network System Results Based on Joint Cache Collaboration and User Interest. The efficiency of the network system varies according to the number of iterations, and the comparative relationship between the efficiency function and the number of users is shown in Figure 7.

Figure 7(a) expresses the relationship between the efficiency of the VR network system and the number of iterations of joint cache cooperation and user interest. It means that the J-CCF algorithm has the best performance among the three comparisons between the cache strategy without cooperation and the random viewport strategy. Moreover, the J-CCF algorithm becomes convergent after the sixth time, the convergence speed is fast, and the convergence performance is good. The proposed J-CCF algorithm has the best performance. Figure 7(b) is a graph showing the

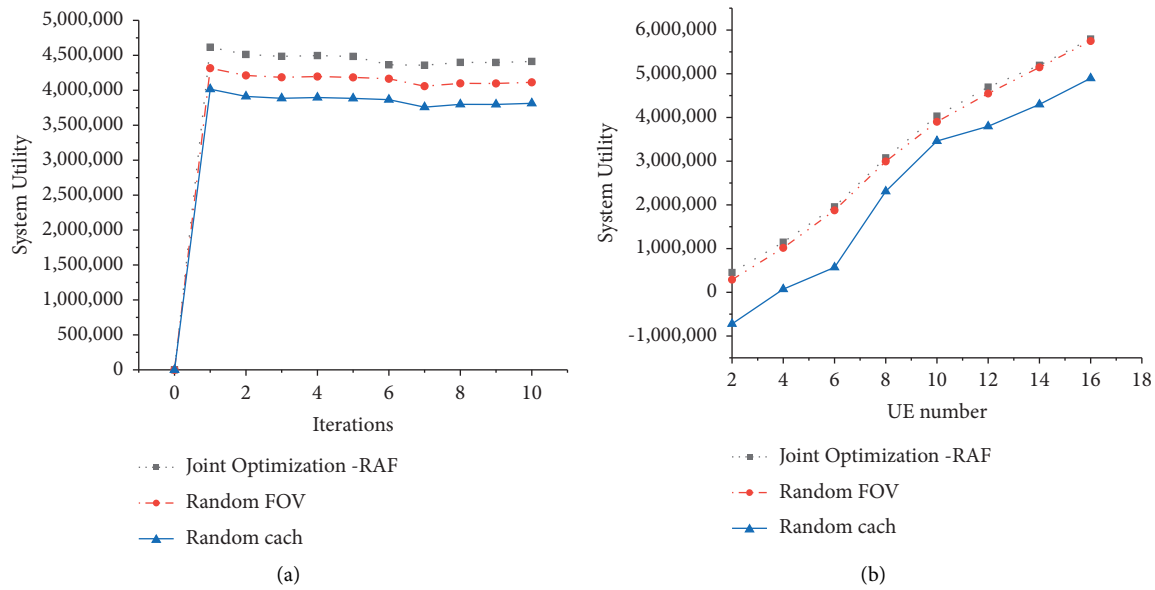


Figure 5: The relationship between the number of users and the system efficiency function according to different optimization schemes: (a) The system efficiency function of different optimization schemes, (b) The system efficiency function of different numbers of users.

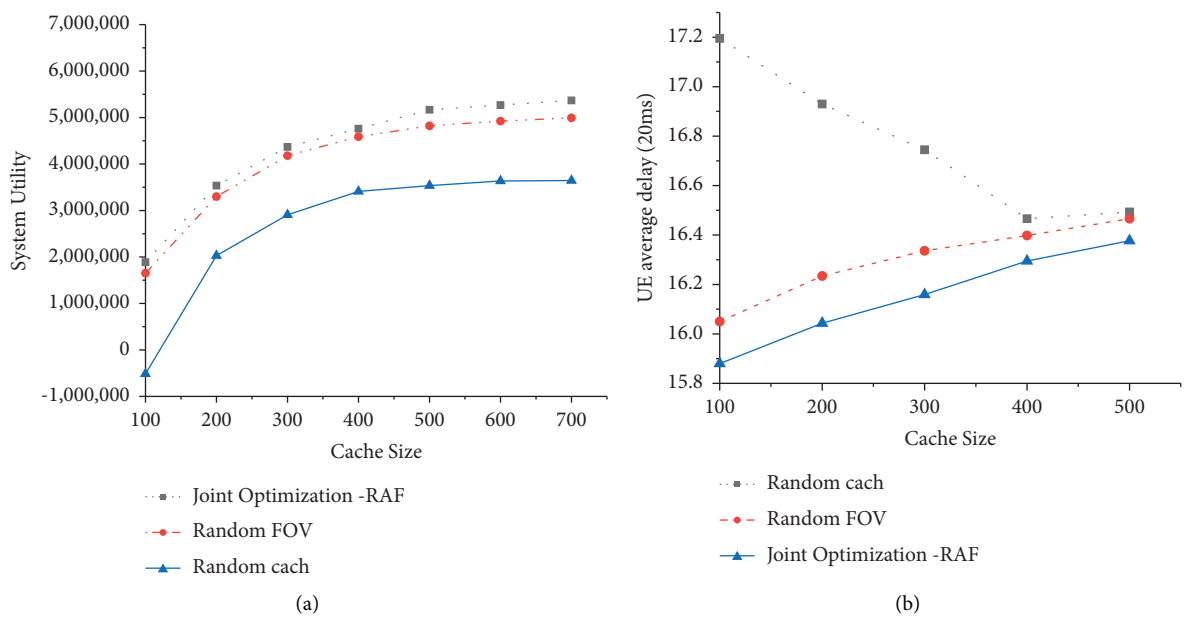


Figure 6: The comparison of the size of caching space, the efficiency function of the system, and the delay time: (a) The efficiency function of the system for different cache spaces, (b) The delay time of user under different cache sizes.

relationship between the system benefit and the number of users. As the number of users increases, the efficiency function of the system also increases gradually, and the increase becomes more and more gentle. And when the number of users is more, it will slowly reach the highest performance of the system. Figure 8 shows the relationship between system benefit and cache space, and the change in cache space and user delay time.

Figure 8(a) represents the relationship between the size of the cache space and the system benefit. Through comparing the J-CCF algorithm, the cache strategy without cooperation, and the random viewport strategy, the J-CCF algorithm has different boosts in terms of performance compared with the other two schemes. The gains are 15% and 25% respectively. However, when the cache space continues to increase, the benefit growth rate of the system gradually slows down. In Figure 8(b), it shows that with the increase of cache space, the average delay time of users increases, and the increase of cache space will provide more services, and the delay time of user will increase slightly. The service rate with and without the strategy is shown in Figure 9.

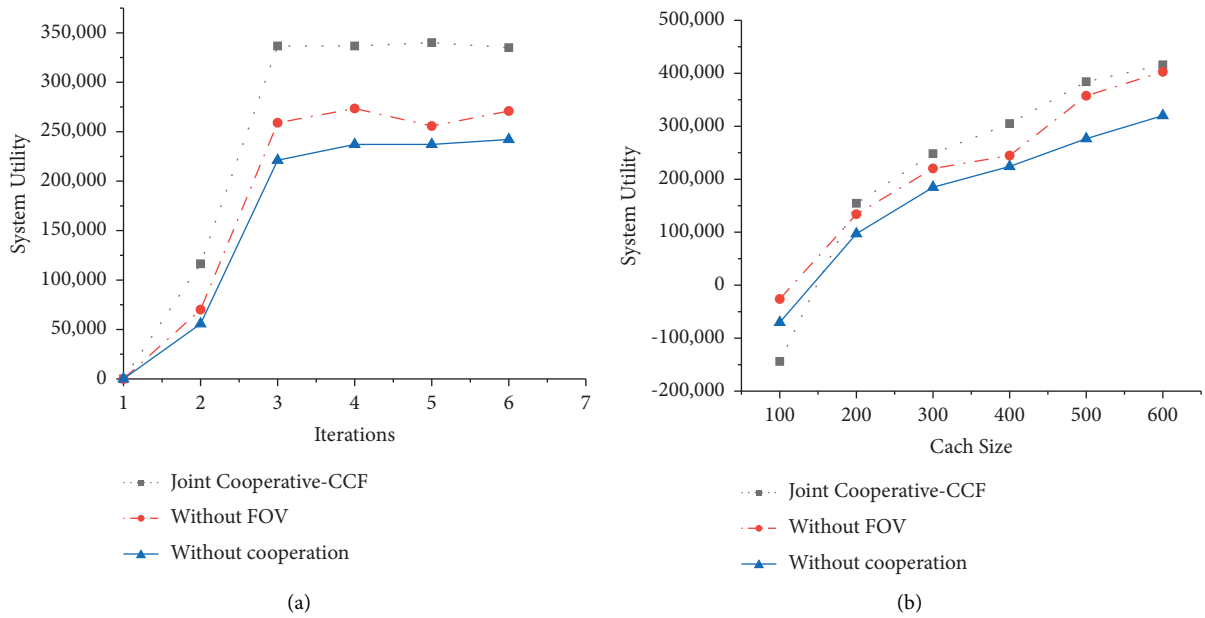


Figure 7: The comparison of the performance of J-CFF algorithm and the average system efficiency function under different numbers of users: (a) The performance of J-CFF algorithm, (b) The result of the average system efficiency function for different numbers of users.

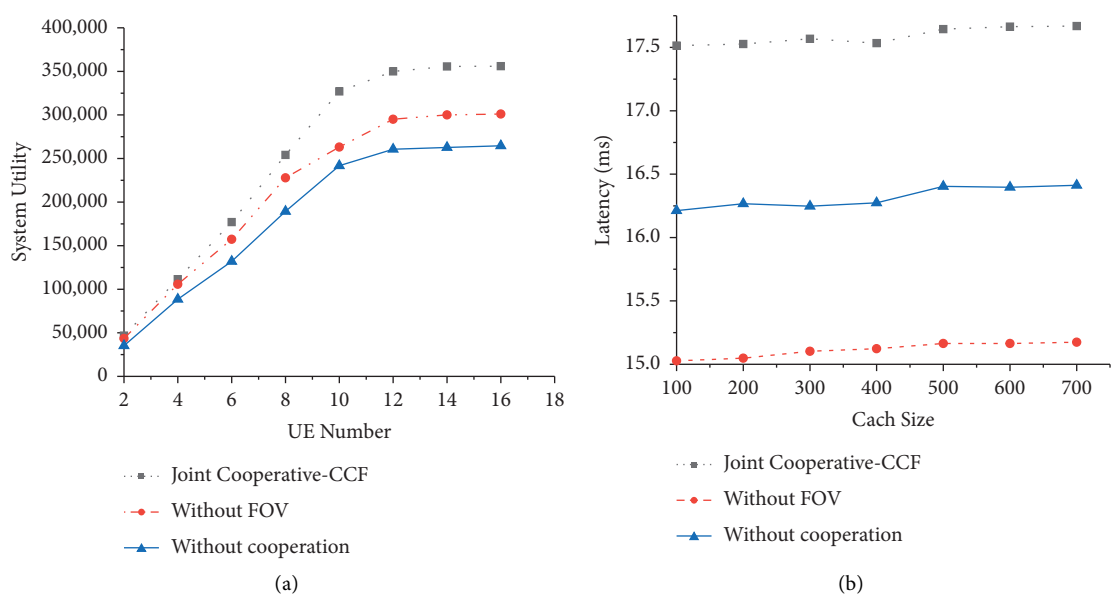


Figure 8: The comparison of the size of caching space, the efficiency function of the system, and the delay time: (a) The efficiency function of the system for different cache spaces, (b) The delay time of user under different cache sizes.

Figure 9 indicates that when the cooperation scheme of J-CCF algorithm is adopted, the overall load situation of the service area is more stable and balanced than that of the service area that does not use the J-CCF algorithm. The load situation can be doubled. The variances of the two schemes are 0.00236 and 0.00076, respectively. Using the cooperation scheme of J-CCF algorithm can balance the load and avoid the situation of business concentration, which verifies the effectiveness of the cooperation scheme of J-CCF algorithm.

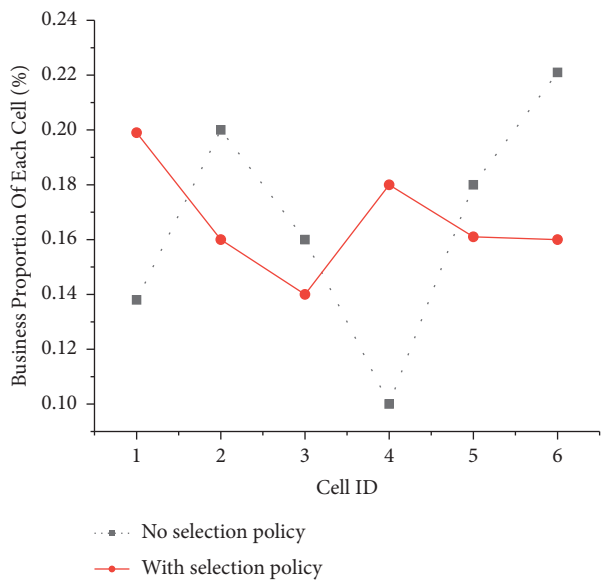


Figure 9: The service rate of each cell before and after the adoption of the scheme.

4. Conclusion

Based on the IoT and edge computing technology, this work analyzes the existing problems of urban sculpture, introduces VR technology into the design of urban sculpture space, and improves the interactivity and experience of urban sculpture space. First, aiming at the introduction of VR technology, to meet the large amount of data and low delay of the IoT required by VR technology, a resource allocation algorithm J-RAF is proposed based on the resource allocation problem of user interest and FOV,, and the effectiveness of J-RAF algorithm is verified by simulation experiments. Second, a VR network system is established based on cache cooperation and user interest, and a resource allocation algorithm J-CCF is proposed. Simulation experiments are used to verify the effectiveness of J-CCF algorithm. Experiments show that J-RAF algorithm can converge faster and has better convergence performance, and has 3% and 20% system gain, respectively. Compared with no cooperative cache scheme and random viewport strategy, J-CCF algorithm can achieve convergence after six iterations, with better convergence performance and 15% and 25% gain in system efficiency. It is proved that the two algorithms are effective for VR technology. The proposed algorithm has a good reference for combining new technology and urban sculpture and improves the interactivity and experience of urban sculpture space based on user behavior. In this work, the use of VR and the setting of user preferences are fixed values, so there is no algorithm in the analysis of user interest and consequently a lack of dynamic calculation of user interest in the calculation process. Hence, the research on resource allocation will be carried out in the future research to strengthen the contribution of dynamic calculation of user interest to the design of urban sculpture space.

Data Availability

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Consent

Informed consent was obtained from all individual participants included in the study.

Conflicts of Interest

All authors declare that they have no conflicts of interest.

Authors' Contributions

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Acknowledgments

The authors acknowledge the help from the university colleagues.

References

[1] F. Yin, H. J. Wen, and X. H. Guo, "Thematic planning of urban sculpture systems: a case study of red sculpture system planning in yan'an," *Journal of Landscape Research*, vol. 12, no. 5, pp. 97–102, 2020.

[2] B. Bachler, "Slowness, streams, and networks in the more-than-human world: prototyping an Internet of things for water," *Journal of Science and Technology of the Arts*, vol. 12, no. 3, pp. 25–44, 2020.

[3] T. Jiang, "Urban public art and interaction design strategy based on digital technology," *Cluster Computing*, vol. 22, no. S2, pp. 3471–3478, 2019.

[4] J. Du, "Research on landscape sculpture design of modern zen and tea theme block," *OALib*, vol. 07, no. 07, pp. 1–8, 2020.

[5] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O Wu, "Edge computing in industrial Internet of things: architecture, advances and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.

[6] T. Wang, L. Qiu, A. K. Sangaiah, A. Liu, M. Z. A. Bhuiyan, and Y Ma, "Edge-computing-based trustworthy data collection model in the Internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4218–4227, 2020.

[7] D. N. Jha, K. Alwasel, A. Alshoshan et al., "IoTSim-Edge: a simulation framework for modeling the behavior of Internet of Things and edge computing environments," *Software: Practice and Experience*, vol. 50, no. 6, pp. 844–867, 2020.

[8] A. Hochman, "RACSO art gallery presents " "ContraFuerte": conceiving an urban sculpture"," *Sculpture Review*, vol. 68, no. 1, pp. 26–31, 2019.

[9] E. Naseri and A. Nadalian, "The emergence of the horseman sculpture in tehran's urban arts," *Journal of History Culture and Art Research*, vol. 8, no. 2, p. 225, 2019.

[10] W. Cudny and H. Appelblad, "Monuments and their functions in urban public space," *Norsk Geografisk Tidsskrift - Norwegian Journal of Geography*, vol. 73, no. 5, pp. 273–289, 2019.

[11] W. Kuang and Y. Dou, "Investigating the patterns and dynamics of urban green space in China's 70 major cities using satellite remote sensing," *Remote Sensing*, vol. 12, no. 12, p. 1929, 2020.

[12] Y. Chen, J. Zhang, Q. Chen et al., "Three-dimensional printing technology for localised thoracoscopic segmental resection for lung cancer: a quasi-randomised clinical trial," *World Journal of Surgical Oncology*, vol. 18, no. 1, p. 223, 2020.

[13] N. A. A. Gonzalez, F. S. Warden, H. N. Q. Milian, and S. Hosseini, "Interactive design and architecture by using virtual reality, augmented reality and 3D printing," *International Journal of Simulation and Process Modelling*, vol. 15, no. 6, p. 535, 2020.

[14] O. Jean-Baptiste, "Augmented and virtual reality art: a new Frontier of legal protection," *Interactive Entertainment Law Review*, vol. 4, no. 2, pp. 102–111, 2021.

[15] A. Akande, "Manifestations of orí (head) in traditional yorùbá architecture," *IAFOR Journal of Cultural Studies*, vol. 5, no. 2, pp. 5–19, 2020.

[16] C. You and J. Li, "Application of landscape sculpture in interior design-taking wood carving as an example," *OALib*, vol. 07, no. 08, pp. 1–5, 2020.

[17] A. Gor, "Reimagining the iconic in new media art: mobile digital screens and chôra as interactive space," *Theory, Culture & Society*, vol. 36, no. 7-8, pp. 109–133, 2019.

[18] M. Li, T.-C. Hsiao, and C.-C. Chen, "Exploring the factors of cooperation between artists and technologists in creating new media art works: based on AHP," *Sustainability*, vol. 12, no. 19, p. 8049, 2020.

[19] X. I. A. Fan and M. Lü, "New space for city communication: a study on culture transmission by nanjing metro[J]," *Cross-Cultural Communication*, vol. 17, no. 1, pp. 30–34, 2021.

[20] H. Zhao, Q. H. Zhao, and B. Ślusarczyk, "Sustainability and digitalization of corporate management based on augmented/virtual reality tools usage: China and other world IT companies' experience," *Sustainability*, vol. 11, no. 17, p. 4717, 2019.

[21] Y. Wu, J. Pan, Y. Lu, J. Chao, H. Yu, and S. Wan, "Psychotherapy for advanced cancer patients: a meta-analysis of the quality of life and survival assessments," *Palliative & Supportive Care*, vol. 14, pp. 1–7, 2022.

[22] W. Song, Y. Wang, M. Liu, and Z. Fei, "Joint optimization of resource allocation and FOV for VR services in mobile edge networks," in *Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pp. 535–541, Chengdu, China, December 2020.

[23] C. Chen, J. Jiang, Y. Zhou, N. Lv, X. Liang, and S. Wan, "An edge intelligence empowered flooding process prediction using Internet of things in smart city," *Journal of Parallel and Distributed Computing*, vol. 165, no. 3, pp. 66–78, 2022.

[24] I. J. Akpan, D. Soopramanien, and D. H. A. Kwak, "Cutting-edge technologies for small business and innovation in the era of COVID-19 global health pandemic," *Journal of Small Business and Entrepreneurship*, vol. 33, no. 6, pp. 607–617, 2021.