Scholar Publishing Group
International Journal of Multimedia Computing
https://doi.org/10.38007/IJMC.2021.020302
ISSN 2789-7168 Vol. 2, Issue 3: 9-18

Design and Implementation of Animation Virtual Battlefield Middleware Based on Artificial Intelligence

Changwan Liu and Fuying Zou

Jiangxi Vocational Technical College of Industry & Trade, Nanchang 330038, China

1332656030@qq.com

Keywords: Artificial Intelligence, Animation Virtual, Virtual Battlefield, Middleware

Abstract: The main research object of this paper is the design and implementation of artificial intelligence in the animated virtual battlefield, including three-dimensional stereo model, virtual reality technology and three-dimensional simulation. By analyzing traditional middleware technology and artificial intelligence algorithm, different templates of system functions are introduced. Finally, the simulation task was completed combined with the VR system, and the corresponding animation virtual battlefield middleware library was established under the Windows operating system to realize the comparison and verification results of the final model and the real picture.

1. Introduction

The competition in animation development industry is becoming more and more fierce. Animation software development requires higher efficiency and reliable software quality assurance at the same time. With the continuous growth of animation software scale, the concept of middleware has been introduced into the animation software development industry [1-2]. Animation software middleware is more and more widely covered in animation development, and the demand of middleware in the field of animation artificial intelligence has also emerged. Animation artificial intelligence is a method and technology to solve the simulation of intelligent object in animation by using artificial intelligence and other technologies in the process of animation software development.

Research have on middleware. In the past two years, these middleware developers have launched some publicity in China, and some domestic game development projects have also begun to seek cooperation with these middleware. For example, in the China Joy exhibition in July 2010, kaitment and Tencent Games announced that they would cooperate in the development of MMORPG online games. However, the independent research and development project in the field of internal game AI middleware is still a blank. Relevant R & D technology and support services are still in relatively backward stages [3-4]. With the growing scale of the industry, it believes that the domestic game development team will gradually realize the advantages and value of adopting the middle. And the domestic independent-developed game AI middleware will gradually play an important role in the game development industry [5-6].

Based on the common virtual battlefield environment in various types of animation, this paper designs and implements a set of practical animation artificial intelligence middleware system according to the requirements and problems in the field of animation artificial intelligence. The terrain and character objects and the common problems that need to be dealt with in the process of interactive attacks are classified and sorted out [7-8].

2. Overview of Animation Virtual Battlefield Middleware Based on Artificial Intelligence

2.1. Traditional Middleware Technology

Middleware is a kind of software technology designed to manage the inherent complexity and heterogeneity of distributed systems. It is defined as a general service between operating system and application. This service has standard program interface and protocol, which can be implemented on different hardware platforms and operating systems, so that program developers only need to face a simple development environment with unified standards and focus on their own business, regardless of the complexity of the underlying hardware structure and the migration of programs to different system software. This greatly reduces the technical burden and procedural complexity [9-10].

Middleware can play a great role in product application: improving the development efficiency of application software, shortening the development cycle, reducing the development cost, improving the development quality, protecting the original hardware, network and other resources, convenient system integration is convenient for system maintenance, upgrade, expansion and transplantation, can prolong the life cycle of application software, reduce the cost of operation and maintenance[11-12]. Middleware covers a wide range, and a variety of middleware products with their own characteristics have emerged according to different application requirements. According to the role of middleware in the system and the technology used, it can be divided into the following three categories:

2.1.1. Database Middleware

The database middleware is the earliest produced and most widely used middleware. It is between the client and the database, and is responsible for receiving the data request of the client, doing the corresponding processing (such as SQL format conversion, determining the data source, etc.), then passing the request to the back-end database server for the final data processing, and then returning the results to the client by the database middleware. Database middleware can not only process the data connected to the database, but also migrate the application processing conducted by the client in the traditional client / server environment to the middle layer, thus simplifying the client, reduce the burden of the client, and also enhance the concurrent processing performance and security performance of the system.

2.1.2. Message Middleware

Message middleware uses reliable messaging mechanism to communicate data between different platforms, which is mainly used in distributed environment where reliable data needs to be transmitted. It provides two models for data communication: the messaging model and the message queuing model. Different objects trigger each other' s events by passing messages to complete the desired operation. Message middleware provides a connection between the client and the server to randomly transmit or store and forward messages.

2.1.3. Distributed Object Middleware

Object-oriented technology presents objects with the characteristics of encapsulation and inheritance, which provides the foundation for the reusability of the software, and the external transparency of the object also coincides with the requirements of the middleware. Distributed object middleware is to provide a unified interface between various objects, so that calls and information sharing between objects are not limited by language, operating system and location, and have interoperability in the application. In recent years, with object-oriented and distributed technologies, many middleware models that combine the two, such as RMI of DCOM, Sun, CORBA, Microsoft, OMG organization, continue to advance object middleware technology.

2.2. Overview of Middleware for Animation Virtual Battlefield

In the field of animation virtual battlefield research and development, the Animation Virtual Battlefield Engine (Gaume Engine) is a software system specially designed for Song to create and develop the electronic animation virtual battlefield. Animation virtual battlefield middleware (Game Middleware) is often used to refer to subsystems of specific functions in the animated virtual battlefield engine. Even some animated virtual battlefield middleware is only responsible for a specific task, and is often compelling efficient compared to other general animation virtual battlefield middleware. For rendering simulated trees and vegetation Animation Virtual Battlefield middleware provides the required core functionality in all aspects of animation virtual battlefield development, is a flexible and reusable software platform, and reduces development costs, development complexity, and development cycle.

Like other middleware solutions, the Animation Virtual Battlefield Engine provides an abstraction of the animated virtual battlefield running platform, allowing the same animated virtual battlefield to run on different animated virtual battlefield platforms. Animation Virtual Battlefield engines are also often designed to be component-based architecture, allowing some parts of animated virtual Battlefield engine systems to be replaced or extended to more dedicated animated virtual Battlefield middleware components.

Animation virtual battlefield artificial intelligence middleware is an integrated component of animation virtual battlefield software system, which is specially used to deal with and solve the artificial intelligence simulation process in animation virtual battlefield. It generally includes animation virtual battlefield artificial intelligence entity and interaction model, and defines the protocol and interface between the system and the external communication, and provides a series of data editing and processing tools in the field of animation virtual battlefield artificial intelligence.

2.3. Artificial Intelligence Technology Used in Animation Virtual Battlefield

There are some classic solutions to intelligent problems in the field of artificial intelligence. These methods look at intelligent problems from a unique point of view, and develop specific algorithms that come from one. The classical methods in the field of artificial intelligence include rule-based expert system, artificial neural network and evolutionary computing. They all have different degrees of application in animation development activities. The following is a brief introduction to their concepts and the process of solving intelligent problems.

2.3.1. Rules-Based Expert Systems

The rule-based expert system simulates the way of human thinking by describing the domain knowledge as a formatted rule and carries on the logical reasoning on the basis of the rule, so as to solve the intelligent problem.

The rules are generally in the format Production Rule, or a judgment rule: IF <Condition> THEN <Action>. It is divided into two parts: Part IF and Part THEN. The IF part is called the former term, and the THEN part is called the latter term. The previous term carries a logical expression that, when the value is true, performs the operation carried by the latter item, called motivating the rule. To calculate the value of a logical expression requires first determining the values of all the variables appearing in the expression that are taken in the expert system are called fact . The operation carried by the latter item can add to the facts or update the existing facts.

2.3.2. Artificial Neural Networks

Artificial neural network is based on the simulation of biological neural network reasoning and calculation methods. In animation, artificial neural network can be used to simplify the development of complex classification and decision-making problems, and the appropriate weight allocation parameters of neural network can be found through the early data training. It can also make use of the learning characteristics of artificial neural network to train dynamically in the process of animation operation in order to adjust the decision-making behavior of artificial intelligence roles and improve the animation experience of designers.

Artificial neural networks are composed of several neurons in certain structures. Each neuron has a thousand inputs (An) and an output (B) and α threshold. Each input corresponds to a Rn, and the weighted sum of all the inputs is A:

$$A = \sum_{n=2}^{n} a_n r_n \tag{1}$$

When the weighted sum A is no less than threshold, neurons are activated at value 4. The value of the output B is a function of the A-4:

$$B = K(A - \alpha) \tag{2}$$

This function K is called an activation function, which generally prefers a symbolic function, a step function, a S-shape function, a linear function, etc.

The output of neurons can be connected to the input of other neurons, and multiple neurons are connected to form a network. The neuronal networks for inference calculation are multi-layer structure, the most front layer is called the input layer, the input terminal connects the input signal, the last layer is called the output layer, their output signal as the output of the entire neuron network, and the middle layer is called the implicit layer. Each output of the previous layer in the two adjacent layers is connected to all the inputs of the latter layer.

3. System Functional Design of Animation Virtual Battlefield Middleware Based on Artificial Intelligence

3.1. Terrain Analysis and Query

The terrain of animation virtual battlefield will generally be divided into some areas with different functions, and there may also be the relationship of inclusion and intersection between regions. According to the different functions, the animation will generally have: ordinary area, line of sight blocking area, action blocking area and so on.

The open terrain is the ordinary area, the smoke covers the line of sight barrier area, the wall, the steep hillside is the movement barrier area at the same time, also is the line of sight barrier area, the semi-person high wall or the transparent glass curtain wall is only the movement barrier area. The functional properties of a zone may also be related to the ability to query objects. Rivers and lakes, for example, are blocking areas for terrestrial organisms, but not for amphibians.

In the process of animation running, the upper animation logic often needs to initiate a query about the location, distance and existence of the terrain area. Some of the common queries are compared to those shown in Table 1:

Table1. Common terrain-related query requirements

Query type	Examples
Determine whether the specified type exists within the range	Some areas can be marked in an animation with a resource where you can collect the resources. The role needs to be able to issue this query to discover the location of the resource
Look for the specified type area within the paradigm usage closest to the specified location	Some areas of the animation have hidden features that are suitable for avoiding enemy hazards. The character needs to find a hidden position in danger
Find the best location for an action	To place the blasting explosive pack, consider the target distance of the attack, the concealment of the resettlement process and the convenient evacuation after the placement
	To snipe a sniper, it is necessary to consider the selected position to the target direction with a good view to observe the target, to avoid being approached by the target and their own position is not easy to detect, it is easy to detect other approaching enemies
Regional connectivity judgments for specific conditions	To snipe a sniper, it is necessary to consider the selected position to the target direction with a good view to observe the target, to avoid being approached by the target and their own position is not easy to detect, it is easy to detect other approaching enemies
Query the connected path	To reach the designated location, the animation needs to arrange a path that you can pass through
Specifies whether the area is a control paradigm for a camp	The isolated role determines whether the current area is safe. Whether you need to consider a retreat or concealment
	In which areas the defense facilities need to be deployed or enhanced
Statistics the scope controlled by each camp	Whether the situation is for the camp, whether a comprehensive attack can be launched
Verify whether your own path is still valid, and adjust the fire effect path	The enemy barricaded at the intersection, and the terrain mountain has a new Yang barrier area

3.2. Role Behavior and Behavior Management

The common group behavior requirements in the animation virtual battlefield are shown in Table 2. The behavior of a character in animation refers to the actions performed by the character, the way in which he acts, and the response to the stimulation of external events. The behavior of the character can show the short-term activity goal of the character at that time, in the animation, the designer needs to experience the goal and character of the character in the animation by observing a number of consecutive behaviors of a character. Therefore, it is necessary to manage multiple related character behaviors for each character in animation, and combine them organically to shape the character characteristics and mission goals that conform to the background of animation story and animation rules.

Table 2. Common group behavior

Behavior requirements classification	Treatment flow sheet
Keep the formation and align the relative position during movement	1.Obtains the current position of several characters, calculating the core of the team and the desired position in the formation by the predetermined formation mode obituary 2. Return the move instruction according to the actual and desired position of each role
Cover, fire support, consider whether the target is a threat to a specific covered target	Obtains the position of the covered object and the surrounding hostile forces 2. Query the position of the covered object near the hostile potential month 3. Schedule the attack target behavior. Stop the hostile forces from approaching these positions

Table 3. Common individual behavioral needs

Behavior requirements classification	Treatment flow sheet
Near a specified target (fixed location or other removable role), escape one or type of specified target and avoid approaching	1.Get the location of the body and the target 2.Prediction and judge the movement trends of targets 3.Select the required direction of motion (close or away) 4.Issue a query to the map module in the direction of motion and get a short motion track
Avoid collisions with other objects and suddenly have high priority with a distance within a certain range	1.Gets the distance closest to the object within a certain threshold range around the white body position of the associated role 2.When the distance is less than a critical value, it returns a moving instruction carrying a high priority departure from the nearest object 3.Attack a specific target 4.Determine the current weapon attributes (remote attack or near-range attack, attack effective model chamber, etc.) and choose the appropriate attack position by reference to itself and the target position 5.Query the path to the attack location 6.Go a status, move 7.When reaching the attack position, the attack
Find, collect resources, allowing direct detection of resource distribution	1.Launch a resource location query request to the map module to obtain the location distribution 2.Query the line reaching the resource location 3.Enter the micro status and call the mobile instruction 4.Perform the acquisition command after reaching the resource location, and then query the recovery point path 5.Continue returning to the move instruction 6.Upon arriving at the recovery point, return the resource recovery instruction

	7.Repeat the entire process
Finding and collecting resources does not allow direct detection of resource distribution	1.Create targets randomly in unexplored areas to test forward 2.Query the path to the destination 3.Enter the excitation state and return to the move instruction 4.Scan the current visual foot resources, add to the discovered resource distribution data, and add the explored area to the known area collection 5.Determines whether to collect at a known resource location
Track, close to the target while avoiding access to the target vision and the scope of other detection	1.Select auspicious close to the target outside the target vision area and have a more hidden position 2.Select auspicious close to the target outside the target vision area and have a more hidden position 3.View the target vision area as impassable and query the path to the location 4.Enter the excitation state and return to the move instruction 5.Repeat the current unfinished path and repeat the process
Personality behavior, in the same environment and its own state condition, simulates and represents character character characteristics by prioritizing certain instructions in optional instructions	Cowardly: a encounter of the power, priority to escape rather than fire 2.Recklessness: encounter hostile forces, give priority to rub and fire 3.Caution: encounter hostile forces, such as their own life attributes in good state, choose to fire, otherwise choose to escape 4. Sacrifice: weigh the survival value of other members of the team when making behavioral decisions, and help other members survive when they encounter hostile forces

The common individual behavior requirements are shown in Table 3. The character behavior in animation can be divided into individual behavior and group behavior according to the number of related characters. The calculation of individual behavior depends on the state of the actor role and the state of its surrounding environment. Group behavior depends on several roles and the environment around them at the same time. Roles can not only have different behaviors at different times, but may also have multiple behaviors at the same time. These behaviors can handle state information about the external environment independently or interactively, and then make recommendations to the role for a variety of different action instructions. In the manager of role behavior, all instructions obtained by a role need to be filtered, combined, and merged according to certain rules. Therefore, there is a need for a way to describe the collaborative relationship between instructions.

4. Testing of Animation Virtual Battlefield Middleware Based on Artificial Intelligence

4.1. System Flow Simulation of Animation Virtual Battlefield Middleware

The working flow of the concentrator software integrated virtual device is as follows: within a fixed time interval, the data collection module calls the data communication module to the terminal device through the Zig bee, Ethernet or their hybrid mode, and returns the data to the collection module, which is stored as memory in the data collection module for other modules until the next time interval is covered by the new collection data at the time interval. When the upper computer needs to obtain the virtual device data, first send the command to the upper communication module, which is responsible for transmitting the communication data of the upper computer to the intelligent monitoring module. After receiving the command, the intelligent monitoring module will make corresponding prompts on the display and distribute tasks according to the command type. If the command is a virtual device related command, the command is distributed to the virtual device module. The module requires some additional files to locate the virtual device to the relevant terminal device. First, the emergence of artificial intelligence enables the user to dynamically modify the read and write operation of the terminal device without recompiling the application software to control the read and write data of the virtual device to the upper level, thus solving the problem of flexible operation of the device. Secondly, the configuration file. Because the virtual equipment finally needs to be positioned to the real terminal equipment, the configuration file of the virtual device cannot be fabricated from empty space, and it needs to be generated according to the existing equipment. Prior to this, the device management module on the concentrator generated a basic device profile while adding to the bound device to record the association with the real device. According to this profile, we can define a virtual device configuration file to establish a mapping relationship from the virtual device to the underlying terminal by obtaining and calling the configuration of the basic device. Since we need a user custom profile, the file description is somewhat more biased towards the natural language, and to convert it into a better recognized language, we add a language transformation module for the software. The self-detection configuration module detects whether new configuration files every once. If present, a final virtual device configuration file will be generated according to the basic device profile and those manually generated by the user without affecting the normal operation of the concentrator. Finally, the concentrator maps the virtual device to the corresponding terminal device through this file and AI technology to achieve data acquisition and device control. The detailed process is shown in Figure1.

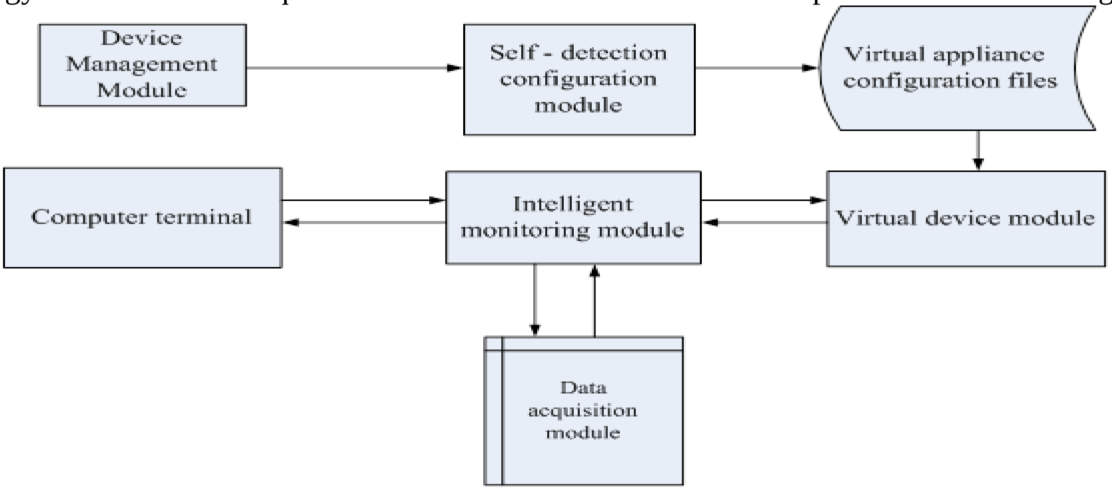
Device Management Module
Self - detection configuration module
Virtual appliance configuration files
Computer terminal
Intelligent monitoring module
Virtual device module
Data acquisition module

Figure 1. Flow chart of virtual device module and related module

4.2. Simulation Environment Simulation of Animation Virtual Battlefield Middleware

4.2.1. Management of Troop Strength Entities

This part includes force creation, force management, force 3D scene model drawing and elimination, and also includes the interaction between force and command post. This function is the core module of virtual battlefield environment system, and its management efficiency directly affects the system performance and user experience. In the virtual battlefield environment, the force not only needs to receive the position, operation and other behavior instructions from the command post, but also constantly updates its internal state and the impact on the surrounding environment according to the system clock.

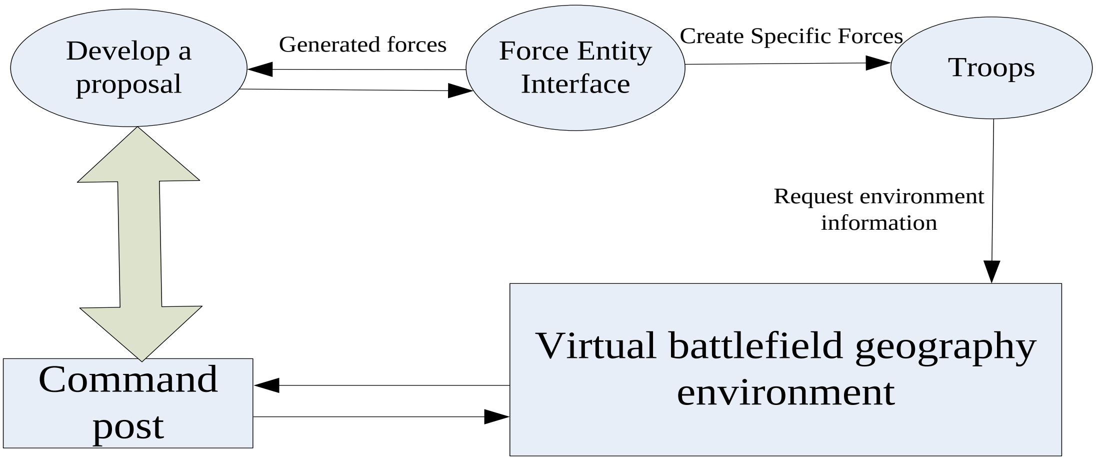

Figure 2. Virtual battlefield environment system forces

As shown in Figure 2, the management of force entity mainly involves the plan, force, command post, geographical environment, and other objects, in which the force object will select the specific implementation interface according to the operational scenario or command post order.

4.2.2. Simulation of Special Effects

Due to natural phenomena such as rain, snow, fog and explosive fire, light, smoke, debris, flame, missile tail flame and other special effects have very irregular appearance shape, not smooth surface, complex structure, change with time, the traditional modeling method is difficult to achieve, but its fidelity directly affects the effect of the whole battlefield simulation. Particle system mainly solves the problem of many small elements of objects on the computer, using the basic simple particle to build the object model, using the most basic geometric elements to construct complex entities, replacing the previous use of polygons and even cash to represent the object, from the perspective of the test situation, the processing effect is very good.

5. Conclusion

In the research work, this paper collects various specific problems in the field of animation AI, and actually designs and realizes the electronic animation artificial intelligence system of middleware facing the virtual battlefield. In the design and implementation process, the parts that may expand and change in the requirements are analyzed, and a special design for these extension requirements is carried out to improve the scalability of the middleware. At the same time, this paper detects the internal state of the module, which facilitates the module testing and the final animation development process, and potentially improves the software correctness and reliability.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] *Ch En K, Zu Y, Cui Y. Design and implementation of bilingual digital reader based on artificial intelligence and big data technology. Journal of Computational Methods in Sciences and Engineering, 202020 (2): 1-19.*

[2] *Chen K, Zu Y, Wang D. Design and implementation of intelligent creation platform based on artificial intelligence technology. Journal of Computational Methods in Sciences and Engineering, 2020, 20 (2): 1-18.*

[3] *Tekerek A, Bay O F. Design and Implementation of an Artificial Intelligence-Based Web Application Firewall Model. Neural Network World, 2019, 29 (4): 189-206. https://doi.org/10.14311/NNW.2019.29.013*

[4] *Ha T, Lee H. Implementation of Application for Smart Healthcare Exercise Management Based on Artificial Intelligence. Journal of the Institute of Electronics and Information Engineers, 2020,57 (6): 44-51. https://doi.org/10.5573/ieie.2020.57.6.44*

[5] *Dilling T J. Artificial Intelligence (AI) for Clinical Analysis: Design and Implementation of a Suitable Database Resource. International Journal of Radiation OncologyBiologyPhysics, 2019, 105 (1): E131 ≤ E132. https://doi.org/10.1016/j.ijrobp.2019.06.2263*

[6] *Fanea L. Disease Management Strategy for Direct and Immediate Implementation of Artificial Intelligence-Based MRI in Radiology. Journal of Biosciences and Medicines, 2019, 07 (9): 38-50. https://doi.org/10.4236/jbm.2019.79005*

[7] *Bhargava A, Bester M, Bolton L. Employees' Perceptions of the Implementation of Robotics, Artificial Intelligence, and Automation (RAIA) on Job Satisfaction, Job Security, and Employability. Journal of Technology in Behavioral Science, 2020 (2): 1-8. https://doi.org/10.1007/s41347-020-00153-8*

[8] *Wang Hua, Li gentry, He Xiaoyu, et al. Design and Implementation of Interactive Mask Erasing Animation Based on Personalized Customization. Journal of Graphicology, 2019,040 (003): 473-480.*

[9] *Li S, He C, Liu M, et al. Design and Implementation of Aerial Communication Using Directional Antennas: Learning Control In Unknown Communication Environments. Control Theory & Applications, IET, 2019, 13 (17): 2906-2916. https://doi.org/10.1049/iet-cta.2018.6252*

[10] *Hyengjun, Jo, Jinwon, et al. Design and Implementation of Artificial arm based on SPI communication. Proceedings of Symposium of the Korean Institute of Communications and Information Sciences, 2017 778-778.*

[11] *Jian Sheng Liu. Design and Implementation of Intelligent Reading Robot Based on STM32F4. Artificial Intelligence and Robotics Research, 2020, 09 (3): 170-181. https://doi.org/10.12677/AIRR.2020.93020*

[12] *Yin L. Design and Implementation of Digital Ink Animation Based on Decorative Style. Paper Asia, 2019, 2 (2): 6-12.*