Hindawi
Complexity
Volume 2021, Article ID 6685861, 11 pages
https://doi.org/10.1155/2021/6685861

Research Article

Fast Adaptive Character Animation Synthesis Based on Greedy Algorithm

Yanqiu Zhu,[1] Qixing Chen,[2] Jing Liu,[1] and Xiaoying Tian[1]

[1]*College of Culture and Art, Chengdu University of Information Engineering, Chengdu 610225, China*
[2]*College of Communication, Chengdu University of Information Engineering, Chengdu 610225, China*

Correspondence should be addressed to Xiaoying Tian; txy@cuit.edu.cn

Received 15 December 2020; Revised 9 January 2021; Accepted 19 January 2021; Published 8 February 2021

Academic Editor: Wei Wang

On the premise of ensuring the animation effect and real-time performance, it is of great significance and value for large-scale group character animation synthesis how to reduce the disaster coincidence degree among various models of fast adaptive character animation synthesis. The realization method of object-oriented finite state machine is studied in detail. Finite state machine (FSM) is an efficient behavior modeling method, which can describe the behavioral decisions of fast adaptive character animation synthesis in a complex virtual environment. Based on the implementation defects of the finite state machine in the traditional structure, using object-oriented thinking, combined with the state design mode, we further studied a finite state machine implementation method based on object-oriented technology. This achieves code reuse and simple program maintenance. The effect is extensible and effectively overcomes the shortcomings of traditional character animation synthesis. Secondly, the multipath matching tracking algorithm of the greedy algorithm is studied to generate multiple candidate sets through multiple paths, and finally, the candidate set with the minimum residual error is selected as the estimated support set, so as to improve the reconstruction performance. Further, based on the idea of multipath, using the regularization method of the ROMP algorithm, the regularized multipath matching tracking RMSP algorithm is proposed. It uses the regularized subset method to generate multiple paths and chooses the path with the fastest residual reduction as the support set of this iteration. The simulation results show that the RMSP algorithm has better reconstruction performance than the SP algorithm.

1. Introduction

Group character animation synthesis has always been a technical problem in computer character animation synthesis. The artificial life method of computer animation overcomes the defects of traditional character animation synthesis technology and improves the efficiency of group character animation synthesis. However, as the character animation synthesis model-making technology used in this method is becoming more and more complex, the disaster coincidence degree among the models of the animation system is also increasing, which makes the character animation synthesis more and more difficult. In particular, when the number of intelligent characters increases, the computational complexity will increase rapidly, which greatly affects the real-time performance of character animation synthesis and limits the wide application of this method. The above problems are studied in depth. Without reducing animation and real-time capabilities, we need to reduce the degree of damage between animation models, reduce computer calculations, and improve the efficiency of character animation synthesis. Based on the two-layer decomposition of independent character animation synthesis and animation, a comprehensive model of group character animation is designed, and the specific realization technology of the model has been studied and realized in depth.

On the basis of the "artificial fish" animation framework, computer vision, machine learning [1, 2], cognitive model [3, 4], and other aspects have been widely carried out, and it has taken the lead in the world. Among them, computer vision research mainly uses computer graphics knowledge to establish a "comprehensive visual model" for character animation synthesis to improve the perceptual performance

of character animation synthesis when the number of characters in group animation increases and/or when the living environment of character animation synthesis is complex and changeable [5]. The research on machine learning and cognitive model is mainly used to improve the intelligence of character animation synthesis, especially the intelligence of higher organisms, so as to realize the simulation of high-level character animation synthesis [6]. At present, the research on the artificial life method of computer animation combined with the multirole animation synthesis system (MAS) and complex adaptive system (CAS) has become the feature and hotspot of computer animation research and creation. Classic greedy reconstruction algorithms include MP algorithm, OMP algorithm, piecewise orthogonal matching tracking algorithm [7], ROMP algorithm [8], compressed sampling matching tracking algorithm [9], subspace tracking algorithm [10], and sparsity adaptive matching tracking algorithm [11]. In addition to the greedy algorithm, there are also a lot of improved algorithms. In literature [12], a multipath subspace tracking algorithm is proposed, which improves the SP algorithm by using the idea of the MMP algorithm. The MSP algorithm has two different threshold choices in the atomic selection stage. The multipath strategy is used to improve the reconstruction of the algorithm. In literature [13], an improved adaptive matching tracking algorithm is proposed. By using the excess criterion of sparsity estimation, the range of sparsity is estimated first, and then, the estimated sparsity is corrected, which speeds up the execution speed of the algorithm and is an improvement of the SAMP algorithm. Literature [14] proposed an improved algorithm, which improved the reconstruction performance of the algorithm by modifying the number of iterations. Literature [15] proposed a regularized adaptive matching tracking algorithm, which combines the characteristics of the SAMP algorithm and ROMP algorithm to improve the reconstruction performance of the algorithm. In literature [16], the SP algorithm is embedded in the OMP algorithm. In each iteration of the OMP algorithm, SP algorithm is used to optimize the estimated support set of the OMP algorithm, which improves the reconstruction accuracy of the algorithm. In literature [17], a forward orthogonal matching pursuit algorithm is proposed, which is the improvement of the OMP algorithm; in each iteration, the LAOMP algorithm selects L atoms and then puts all the atoms into the estimated support set; we choose the smallest residual as the cycle of the currently selected atoms, and the LAOMP algorithm can greatly improve the accuracy of the support set. Literature [18] combined the ROMP algorithm and genetic algorithm to improve the image reconstruction performance of the algorithm. In literature studies [19, 20], Monte Carlo matching tracking algorithm was proposed. The recursive Bayesian process was used to obtain the denoising estimation of signals, and the sparse degree of weakly matched constrained signals was used to reconstruct signals under the condition of unknown sparsity and noise pollution. The core of the automatic generation scheme of bipedal character animation is to create a real-time data-driven character animation synthesis generation model, which enables the character animation synthesis to control the generation of character animation through natural, fluent, and diversified character animation synthesis after making behavioral decisions autonomously [21]. Gaussian process implicit variable model [22] is applied to calculate the low-dimensional potential space of data to help solve the redundancy problem of reverse dynamics [1]. The model based on the kernel method [23] is affected by the high memory cost of processing the covariance matrix. A local GP method to limit the number of interpolating samples is proposed to overcome this problem, but the K-nearest neighbor search algorithm [24] is needed. This algorithm requires high memory requirements in both predictive computation and runtime when processing high-dimensional data such as human character animation data. In order to generate the character animation synthesis that adapts to the environment geometry in real time in the character animation, a phase function neural network [25, 26] is proposed, which can map the control information of the game controller as input to the character animation synthesis. The model uses a phase function to generate variables representing the period of the character animation as the weight of the regression network for each frame [27]. Once generated, these weights will be used in the neural network to generate the role posture of the current frame matching the control parameters in real time [28–30]. In addition, the PFNN also solves the drift problem by performing IK posttreatment on the feet.

The fast adaptive character animation synthesis of the multipath greedy reconstruction algorithm is studied. Firstly, the MMP algorithm is introduced in detail, then the MMP algorithm is analyzed, and the RIP condition of the MMP algorithm is given for the correct reconstruction of the signal support set under noisy and noiseless conditions. The reconstruction performance of the MMP algorithm under noisy and noiseless conditions is simulated and analyzed. On this basis, the regularized multipath subspace tracking algorithm is proposed, and the SP algorithm is improved by using the idea of MMP algorithm and ROMP algorithm regularization. This paper proposes a greedy algorithm-based automatic character animation generation solution for the character animation synthesis and describes the network model structure in this solution and how to use this model to generate the character animation synthesis rapidly adaptive to the character animation synthesis, including the processing of training samples and the training of the model. This paper deals with the defects of the finite state machine (FSM) based on the structural method in implementing the role behavior decision model. When describing the behavioral decisions of swarm intelligence roles with changeable states, complex external stimuli, and frequent state transitions, there are great defects in the implementation, which increase the difficulty of implementation. Studying and solving the above defects are beneficial to reduce the difficulty in animation implementation, realize code reuse, and improve the efficiency of implementation.

2. Automatic Generation Scheme for Fast Adaptive Character Animation Synthesis

Aiming at the defects of the character animation synthesis model similar to "artificial fish," such as difficulty in establishment and control, complex constraint relation, huge computation, and limited real-time performance of animation, a new group character animation synthesis model was designed, which is a two-layer decomposition group character animation synthesis model based on autonomous character animation synthesis. From another point of view, the character animation of an intelligent character can be decomposed into two relatively independent character animations in the world coordinate system and in the local coordinate system, in which the intelligent character is regarded as a particle in the world coordinate system, and the speed and direction of the character animation are specified for this particle. The character animation in the local coordinate system specifically analyzes the character animation of different parts of the body in the local coordinate system of the intelligent character. At any moment, the character animation of the intelligent character is the synthesis of the character animation of the intelligent character in the world coordinate system and the character animation of the intelligent character in the local coordinate system. According to this idea, a two-layer decomposition group character animation synthesis model based on autonomous character animation synthesis was designed (see Figure 1).

It can be seen from Figure 1 that the two-layer decomposition group character rapid adaptive animation synthesis model based on autonomous character animation synthesis can be divided into three models: perception model, behavior decision model, and character animation model, which correspond to the perception system, behavior control system, and character animation system that realize the individual role of animation, respectively. Among them, the behavior decision model is divided into two submodels: motivation model and behavioral decision model. The body character animation model also includes two submodels: the whole character animation model and the animation model.

First, the original animation uses a data segmentation algorithm, and the key frame extraction algorithm uses character animation data editing technology to preprocess the data, and then, according to the semantic characteristics of semantic annotation, the same semantic character animation quickly adapts to the character animation data structure, and the relationship between the database according to the national construction character animation and the design of the database structure is optimized with character animation, and it can provide interface for the expression of user demand not only according to the needs of users in the database to perform role animation query, role animation connection, role animation smooth, and other role animation editing operations but also according to the needs of users to meet the requirements of the role animation sequence.

In the process of character animation synthesis, the corresponding character animation segment is selected from the character animation library according to the semantics. If the end frame of a certain state is similar to the start frame parameter of another state, a connection relationship can be established between the animation states of these two characters, or an artificial connection relationship can be established. The design of the finite behavior state machine requires the character animation database with semantic annotation, which is constructed with randomness, and the diversity of the generated character animation behavior also depends on the design automaton. By building a finite behavior state machine, the character animation of a virtual person in a virtual environment can be generated under the target drive and can be applied to different roles and a variety of complex environments. When a new type of character animation captures data, manually and semantically annotate the data for the new type of character animation as a new state node and then add a new state node for the automaton to be applied.

Given a fast adaptive character animation path curve $t(u)$ and a velocity function $m(t)$, the task of velocity interpolation is to calculate the fast adaptive root joint interpolation position in each frame period. Let n be the arc length of the curve. According to the definition, the derivative of arc length with respect to parameter U can be obtained:

$$\frac{d_n}{d_u} = \sqrt{\left(\frac{d_x}{d_u}\right) + \left(\frac{d_y}{d_u}\right)^2 + \left(\frac{d_z}{d_u}\right)^2}. \tag{1}$$

All the joints used when capturing animation data are Euler angles to represent the joint rotation, and due to the direct interpolation algorithm, the use of Euler angles will appear unrealistic. Therefore, we first convert the quaternion of Euler angles, then proceed linear interpolation, and finally convert Euler angle:

$$s(t) = \frac{\sin\theta}{\sin t} s_1 + \frac{\sin(1-\theta)}{\sin(1-t)} s_2. \tag{2}$$

Based on the finite behavior state machine, the search tree is established by the global search algorithm, and the search path is generated in real time according to the two-dimensional grid map. The construction process of the tree is the search and search process of the character animation synthesis state, which is subject to the constantly changing environment constraints. The global search algorithm adopts the algorithm to establish two related data structures: search tree and state cost queue. In the global search algorithm, the steps to select the best advantage include the lowest cost node in the queue, the end node, and all the state nodes that can be connected after the Sbest node, as shown in Figure 2. The character animation synthesis uses motion capture data to create a directed graph, which is composed of the original motion segments and the transitions between motion segments and can represent the relationship between motion segments. After the character animation composition is established, the new movement can be generated by traversing the graph. The traversing process is the process of combining the motion fragments.

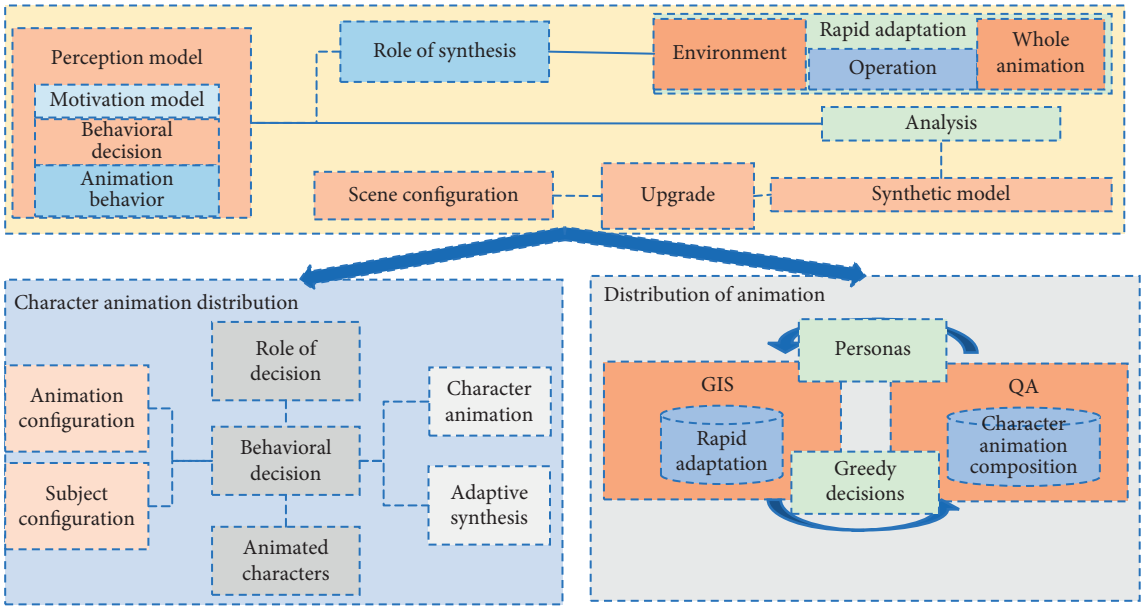

Figure 1: Flowchart of the autonomous character animation synthesis system.

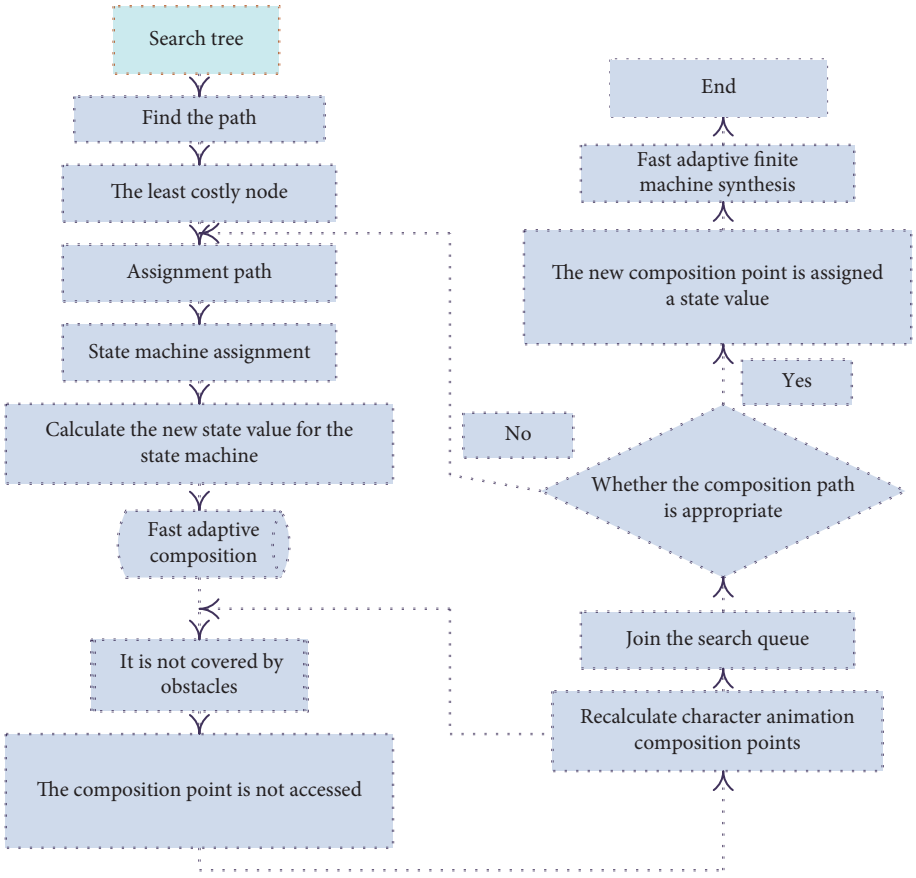

Figure 2: Fast adaptive character animation synthesis process for the finite behavior state machine.

3. Research on Fast Adaptive Character Animation Synthesis Based on the Greedy Algorithm

In the practical process of solving the problem, the multipath greedy algorithm has several key points as follows: the first is to set the initial conditions. Usually, the initial conditions are generated according to the problem to be solved, but the optimal initial conditions can be made in conjunction with the local optimal solution method to the extent allowed. The second is how to solve the problem of local optimization. The optimization problem is generally solved by proposing an optimization function. The quality of the optimization function directly determines the quality of the whole greedy algorithm, which is the most critical link of the greedy algorithm. The third point is to judge the end conditions. The so-called end condition includes two parts. One is the end condition of the global greedy algorithm, which should be determined according to the accuracy requirement of the problem. The other is the constraint condition for optimal selection at each local stage, which determines whether the optimal solution has been obtained for the optimization function. Therefore, it determines the overall complexity of the algorithm to some extent. To sum up, the basic idea of the greedy algorithm to solve problems can be summarized as follows:

(1) Determine the end condition of the algorithm according to the problem target

(2) Analyze the constraint conditions for obtaining the local optimal solution of the problem

(3) According to the constraints of the optimal solution and the understanding of the overall problem, determine the optimization function

(4) According to the algorithm end conditions determined by the three steps above, the constraint conditions of the optimal solution, and the optimization function, formulate the greed criterion so that each link can be connected to determine the optimal solution to the greatest extent

In the actual problem-solving process, the specific solving steps of the multipath greedy algorithm can be summarized as follows:

Step 1: set up the initial solution to the problem

Step 2: follow the greedy strategy loop t and search for the next target until the exit loop is satisfied

Step 3: calculate a solution element s in the feasible solution

Step 4: recombine all the resulting solution elements to become a final feasible solution to the problem

In the process of the multipath greedy algorithm, the smallest element is added to the existing merge scheme. In order to ensure the operation speed of the algorithm, after adding a new minimum unit each time, only K merging schemes with the maximum current value are kept. When the maximum value is in parallel, the remaining schemes are randomly selected. The algorithm runs until the minimum unit merge is complete. Taking $K = 2$ as an example, this paper shows how the greedy algorithm adds the smallest unit to the existing merge scheme. The algorithm is shown in Algorithm 1, and the algorithm flowchart is shown in Figure 3.

The core of the biped character animation automatic generation scheme is to create real-time character animation synthesis based on a data-driven model. This enables character animation synthesis to control the formation of character animation through natural, smooth, and diverse character animation synthesis. The difference lies in that the character animation synthesis of the fast adaptive character is more complex, so the fast adaptive animation is more difficult. In general, animators must be specially trained to create complex character animation compositions for fast adaptive characters.

The MANN-based character animation synthesis and generation scheme mainly includes two parts:

(1) Model training: using a large number of character animation synthesis capture data for model training, the character animation synthesis generation model was obtained

(2) Composition and generation of character animations: using the trained character animation synthesis model, the character animation trend control synthesis obtains the behavior control module behavior modeling character animation synthesis (the user controls when replacing the training model), and the character animation synthesis is generated to obtain the character animation synthesis generation model input

The MANN framework is divided into three stages: pretreatment stage, training stage, and operation stage. In the preprocessing stage, training data are first prepared, including the gait category label of the character animation synthesis data. In the training phase, the MANN is trained with these data in order to generate character animation synthesis in each frame with given control parameters. During the operational phase, MANN's input parameters are collected from the character animation synthesis and environment and input into the system to determine the character animation synthesis.

MANN's input/output and corresponding features to be extracted are described. Input for each frame includes the following: the position, direction, and speed of the trajectory, the character's gait category, and the position, speed, and rotation angle of the previous state node. Similarly, the output includes the predicted trajectory position, direction, and velocity, the position angle and velocity of the joints in the current state, the transformation of the root node, and the velocity of the previous frame. In order to avoid the quaternion interpolation problem of the neural network during training, the rotation of the joint is represented by the rotation of the relative forward and upward vectors. These

Input: minimum cell T and maximum number of reserved schemes L
Output: region R after integration
(1) $t_i \longrightarrow T$
(2) for each t_i do
(3) for each $s_i \longrightarrow S$ do
(4) $s_i \cup t_i \longrightarrow S'$
(5) for each s
(6) $s_i + t_i$
(7) $S' \cup s_i \cup t_i$
end for
(8) for each t_i do
(9) $S' \cup s_i \cup t_i \longrightarrow$ fit
end for
The biggest K solutions of fit
end for
The biggest solution of fit
return R

Algorithm 1: Reconstruction of the multipath greedy algorithm.

are taken from the original quaternions in the character animation synthesis capture and converted back at run time to generate the character animation synthesis.

The input is divided into two parts:

(1) The synthesis state of the role animation in the previous frame is represented by the position, rotation, and velocity of the joint node. After obtaining the global features, it is projected to the ground, and the rotation is aligned with the forward vector of the role.

(2) In the control part, frame I is used as the center of the time window to check every t frame around. For example, $t = 12$ is used to sample the surrounding frames, covering the character animation synthesis of 1 second before and 1 second after. A series of features were extracted from each sampled frame fragment data, including label variables for the position, orientation, speed, and gait of the character in frame I.

The output is divided into two parts:

(1) The synthesis state of the role animation of the current frame is represented by the position of the joint node relative to the root node, the speed and rotation, the displacement of the root node, and the angular velocity projected by the root node to the ground

(2) Prediction of the position, direction, and speed of the synthetic trajectory of character animation in the next frame

The description of network hyperparameters is shown in Table 1.

After the training, the parameters of the gating network and character animation synthesis prediction network were saved as the final model parameters. During the operation, neural network input X must be provided for each frame, and variables related to joint position and speed are used as input for the next frame by means of autoregression with the calculated results of the previous frame. Input X also includes past/future trajectories.

The future trajectory is predicted. When preparing MANN's operation input X, the character animation synthesis gait label is obtained through the character animation synthesis control parameters of the character animation synthesis, enabling the required character animation synthesis, including sitting, standing, idle, lying flat, jumping, and moving, and obtaining the speed and direction of the character. The target velocity and direction are interpolated smoothly and are ultimately used to predict future trajectories. The trajectory curve is inferred using an exponential weighted bias value, which defines the maximum length of the future trajectory to provide the required velocity of the role in m/s and to produce a smooth trajectory. This constitutes all of MANN's input variables, at which point the output Y can be calculated.

The character animation synthesis obtains the terrain height information through the perception system and passes it as a parameter to the character animation synthesis generation system. The obtained character animation synthesis can better adapt to the terrain, make varied character animation synthesis, and increase the diversity and sense of nature of the character animation synthesis.

4. Example Verification

The reconstruction performance of the MMP algorithm is analyzed by MATLAB simulation and compared with the OMP algorithm and SP algorithm. In the case of no noise, the probability of successful reconstruction of these algorithms is compared, and in the case of noise, their reconstruction errors are compared.

Set the signal length $N = 256$, the size of the measurement matrix is 128×256, the sparsity K selects different values between 10 and 70, and the step length is 5. For each fixed sparsity, 500 independent experiments were carried out with the reconstruction probability varying with the sparsity. The change

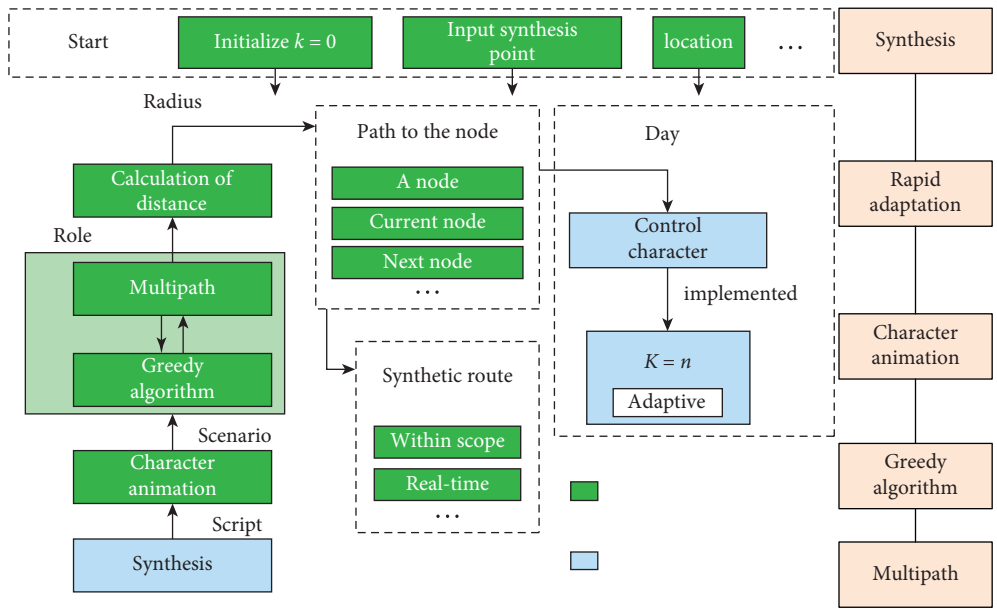

Figure 3: Multipath greedy algorithm reconstruction of the character animation synthesis flowchart.

Table 1: MANN network hyperparameters.

Parameter name	Parameter selection
Learning rate	0.0002
Batch size	34
Epoch	155
Dropout	0.9
Weight decay rate	0.0028
Number of expert networks K	3.5

of K with successful reconstruction is shown in Figure 4, where Figure 4(a) is the reconstruction result of Gaussian signal and Figure 4(b) is the reconstruction result of 0-1 signal. As can be seen from Figure 4, for the reconstruction of Gaussian signal and 0-1 signal, the reconstruction performance of the MMP algorithm is between the OMP algorithm and SP algorithm.

Set the signal length $N = 256$ and the signal sparsity $K = 30$, and select different measurement number M between 60 and 160 with a step length of 5. Conduct 500 independent experiments for each fixed measurement number, and the reconstruction results are shown in Figure 5, where Figure 5(a) is the reconstruction of the Gaussian signal and Figure 5(b) is the reconstruction of 0-1 signal.

As can be seen from Figure 5, the OMP algorithm has the worst reconstruction effect for Gaussian signal reconstruction, the MMP algorithm and SP algorithm have better reconstruction effect, and when the number of measurements is small, the MMP algorithm is better than the SP algorithm; when the number of measurements is large, the SP algorithm is better than the MMP algorithm. For 0-1 signal reconstruction, the reconstruction performance of the MMP algorithm is between the OMP algorithm and SP algorithm.

As can be seen from Figure 6, in the case of noise, the MMP algorithm has the best reconstruction effect for Gaussian signal reconstruction, while for 0-1 signal reconstruction, the MMP algorithm has the middle reconstruction effect.

MATLAB was used to simulate the greedy algorithm's fast adaptive character animation synthesis algorithm. Figure 7 shows the simulation results of placing the greedy algorithm under unlimited and limited conditions. It can be observed from Figure 7 that the bar chart represents the energy consumption under the unlimited condition and the limited condition. It can be seen from the figure that the energy consumption under the restricted condition is not greater than that under the unlimited condition.

As can be seen from Figure 8, when the maximum number of schemes retained by the greedy algorithm is fixed, with the increase of the minimum number of units 1, the running time of the minimum unit combination can be approximated to polynomial time. When $K = 7$, the operation time is around 3 s.

It can be seen from the score that the method in this paper has higher score than the regular deformation method in terms of exaggeration and artistry, which shows that the greedy algorithm is effective in discovering character animation, and the exaggeration of features is also in line with people's observation purpose of character animation synthesis. The comparison results are shown in Table 2.

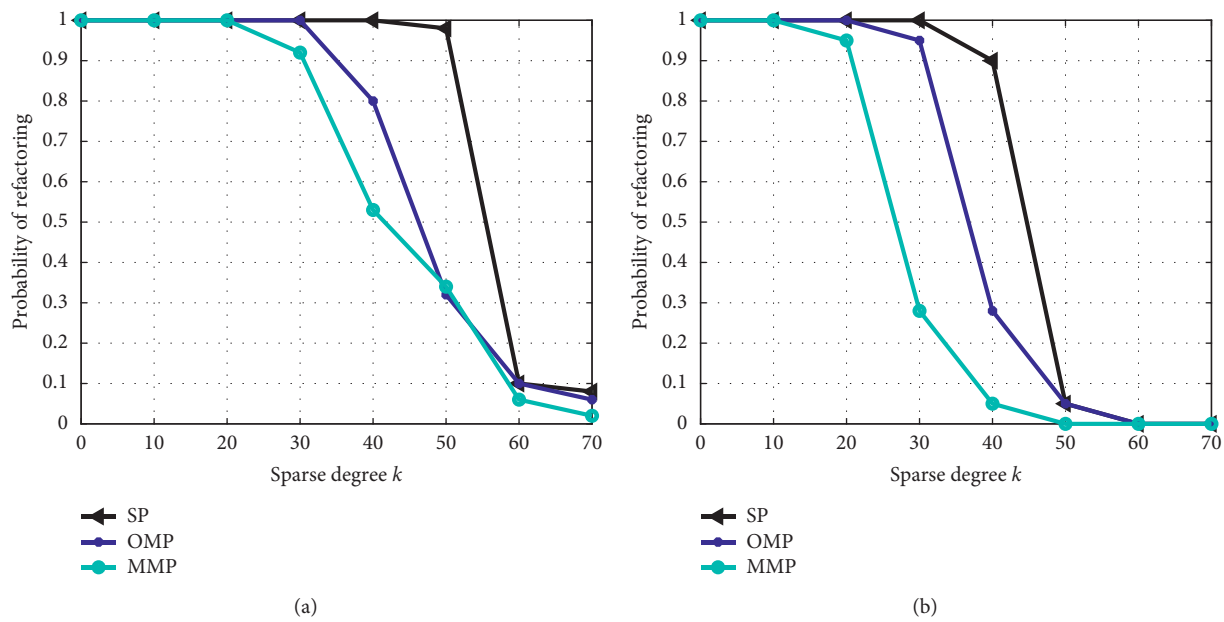

Figure 4: .Variation of reconstruction probability of different greedy reconstruction algorithms with sparsity K. (a) Gaussian signal. (b) 0-1 signal.

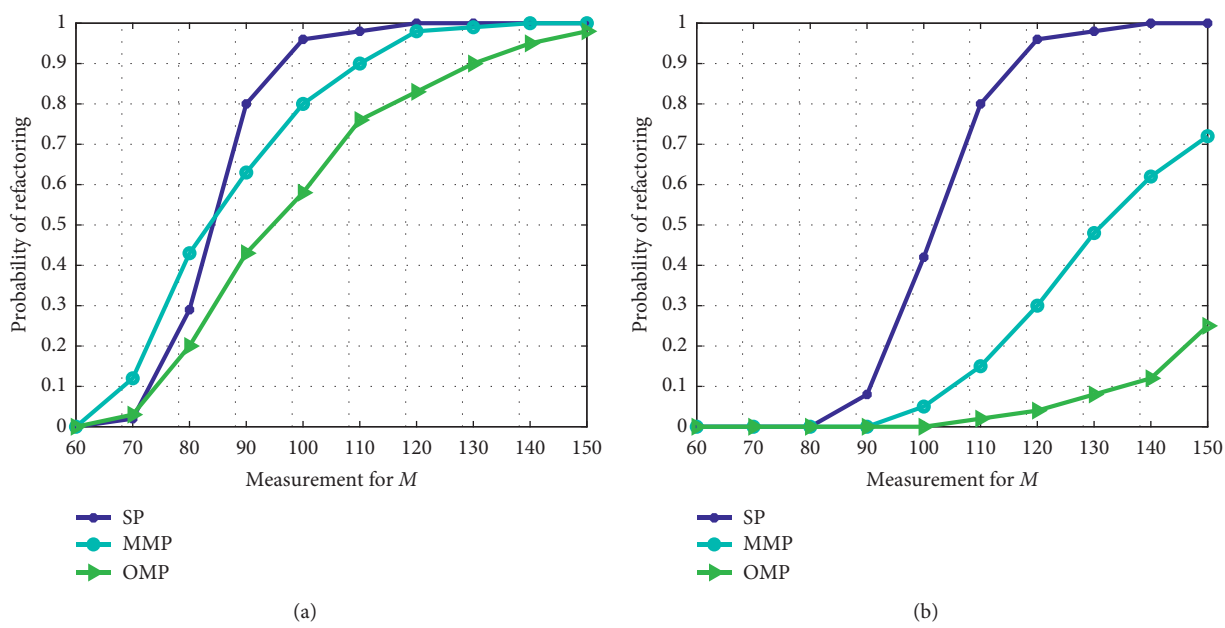

Figure 5: Variation of reconstruction probability of different greedy reconstruction algorithms with measured number M. (a) Gaussian signal. (b) 0-1 signal.

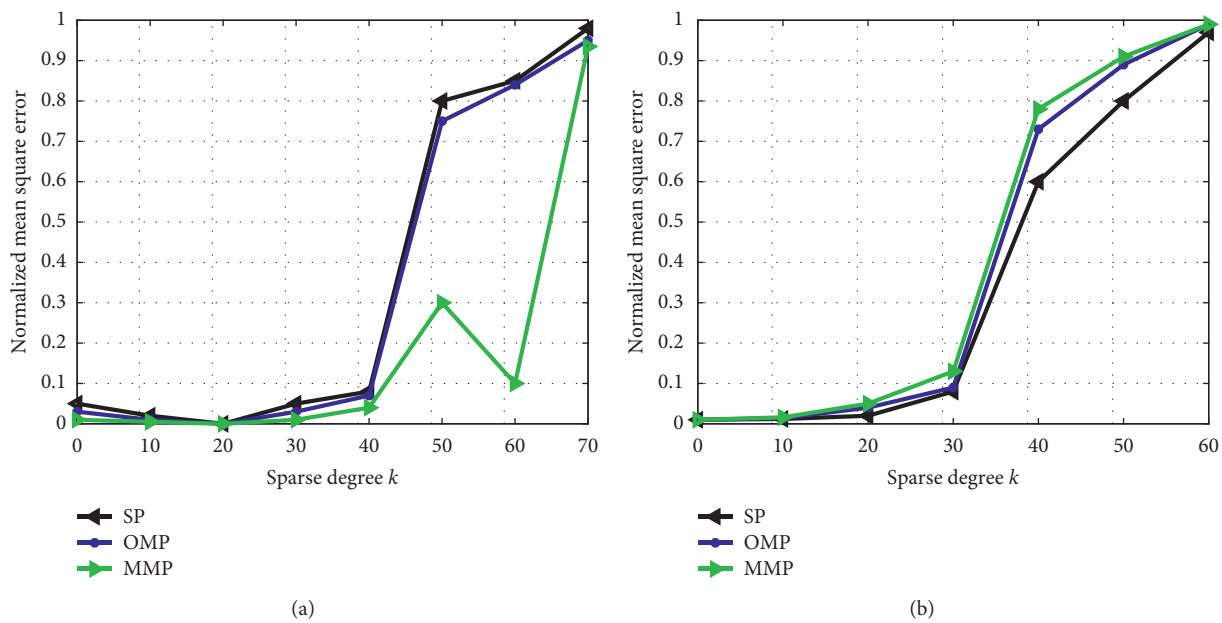

Figure 6: .Variation of reconstruction errors of different greedy reconstruction algorithms with sparsity K. (a) Gaussian signal. (b) 0-1 signal.

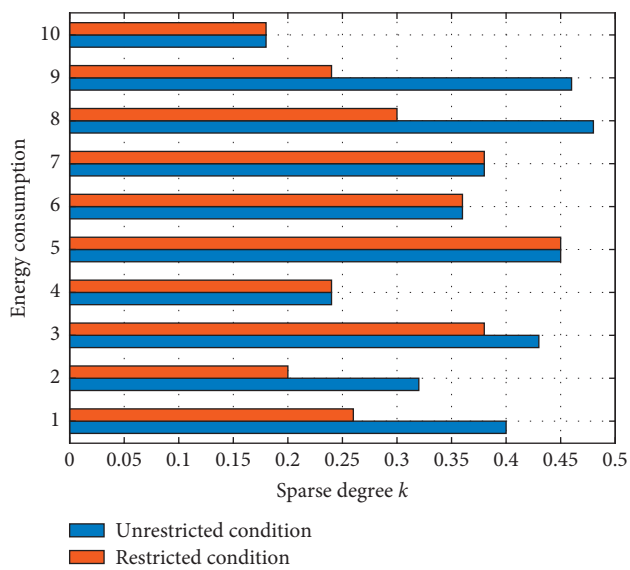

Figure 7: Energy consumption of greedy algorithm's fast adaptive character animation synthesis.

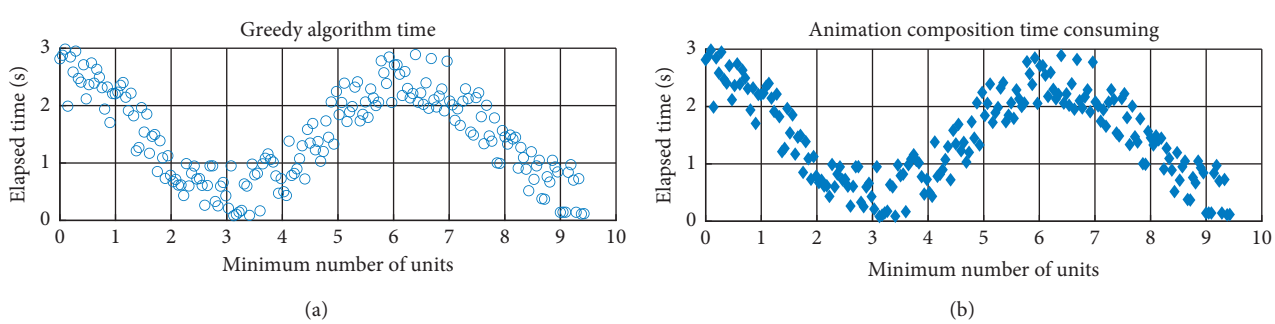

Figure 8: Greedy algorithm's fast adaptive character animation synthesis time. (a) Greedy algorithm time. (b) Animation composition time consumed.

Table 2: Comparison table of result evaluation scores.

Method	Role 1		Role 2		Role 3	
	Greedy algorithm	Rules of law	Greedy algorithm	Rules of law	Greedy algorithm	Rules of law
Similarity	3.54	3.77	3.35	4.0	3.23	3.65
Exaggeration	3.85	3.62	3.77	3.69	4.12	3.46
Artistry	3.54	3.38	4.08	3.5	3.73	3.54
Evenly split	3.64	3.59	3.73	3.73	3.69	3.66

5. Conclusion

The multipath greedy algorithm is studied, and the performance of the MMP algorithm in noisy and noise-free character animation synthesis and reconstruction is theoretically analyzed. The RIP condition obtained is better than the existing algorithm. The RMSP algorithm is further improved, and the RMSP algorithm is proposed. This algorithm uses the feature of the MMP algorithm multipath and ROMP algorithm regularization for reference and selects the path with the minimum residual error to estimate the support set. The simulation results show that the RMSP algorithm has a good performance of rapid adaptive character animation synthesis and reconstruction. An object-oriented finite state machine implementation method is studied in detail. Finite state machine (FSM) is an efficient behavior modeling method, which can effectively depict the behavioral decisions of organisms in a complex virtual living environment. For large fast adaptive character animation synthesis, traditional finite state machine implementation based on the structured method of multifarious, code reuse, hard maintenance, and expandability of issue, using object-oriented technology, combining with the state design pattern, further study of a kind of method is realized by using the finite state machine of object-oriented technology, effectively solving the defects of traditional fast adaptive character animation synthesis.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Sichuan Education Department Humanities and Social Sciences Key Research Base Animation Research Center Key Project "Sichuan Taoism Cultural Resources Animation Gamification Research" (DM201802) and the project of "Sichuan Network Culture Research Center" of Sichuan Social Science Key Research Base: "Exploration on the Road of Network Animation Industry Based on "Child Safety" (WLWH17-33).

References

[1] B. Wang and K. Xie, "Design of open parking lot based on grid-segmentation and greedy-algorithm," *World Entific Research Journal*, vol. 5, no. 9, pp. 268–277, 2019.

[2] U. Bayazit, "A greedy region growing algorithm for anisotropic stretch adaptive triangulation of geometry images," *Graphical Models*, vol. 106, pp. 101045–101057, 2019.

[3] C. Chesseboeuf, "A greedy algorithm for brain MRI's registration," *Acta Biotheoretica*, vol. 64, no. 4, pp. 1–16, 2016.

[4] X. Xiao, C. Yang, and X. Yang, "Adaptive learning-based projection method for smoke simulation: adaptive Projection Method based on Machine Learning," *Computer Animation and Virtual Worlds*, vol. 29, no. 3-4, pp. e1837–e1848, 2018.

[5] H. Shi, Y. Song, and L. Shi, "A high-influence greedy maximization algorithm based on community structure," *Journal of Computational Information Systems*, vol. 11, no. 2, pp. 449–456, 2015.

[6] N. G. Nair and A. S. Spanias, "Fast adaptive algorithms using eigenspace projections," *Signal Processing*, vol. 83, no. 9, pp. 1929–1935, 2015.

[7] Y. Wang, Y. Y. Tang, and C. Zou, "Modal regression based greedy algorithm for robust sparse signal recovery, clustering and classification," *Neurocomputing*, vol. 2019, pp. 372–389, 2019.

[8] X. Wang, Y. Zhang, and W. Wang, "Image segmentation using fast generalized fuzzy C-means clustering based on adaptive filtering," *Moshi Shibie Yu Rengong Zhineng/Pattern Recognition and Artificial Intelligence*, vol. 31, no. 11, pp. 1040–1046, 2018.

[9] M. M. Selim, R. P. Koomullil, and A. S. Shehata, "Incremental approach for radial basis functions mesh deformation with greedy algorithm," *Journal of Computational Physics*, vol. 340, pp. 556–574, 2017.

[10] G. Yu, M. Gao, and C. Jia, "A fast filtering method based on adaptive impulsive wavelet for the gear fault diagnosis," *ARCHIVE Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science*, vol. 203, no. 210, pp. 9544–9568, 2020.

[11] K. W. Cheung, Y. T. Siu, and T. W. Shen, "Fast adaptive bases algorithm for non-rigid image registration," *Journal of Imaging Ence and Technology*, vol. 63, no. 1, pp. 010505.1–010505.8, 2018.

[12] W. Qi, "Research on *K* Maximum dominant skyline and E-GA algorithm based on data stream environment," *Computer Systems Science and Engineering*, vol. 33, no. 5, pp. 369–378, 2018.

[13] Y. Song, G. Peng, D. Sun, and X. Xie, "Active contours driven by Gaussian function and adaptive-scale local correntropy-based *K*-means clustering for fast image segmentation," *Signal Processing*, vol. 174, pp. 107625–107639, 2020.

[14] S. Khadirnaikar, A. Chatterjee, and P. Kumar, "A greedy algorithm-based stem cell LncRNA signature identifies a

novel subgroup of lung adenocarcinoma patients with poor prognosis," *Frontiers in Oncology*, vol. 10, pp. 786–793, 2020.

[15] L. Yu and J. Zhao, "Hybrid optimization method of sparse FIR DFEs based on reweighted 1-norm minimization and greedy algorithms," *Electronics Letters*, vol. 55, no. 8, pp. 45–76, 2019.

[16] G. S. Malyshkin, "Experimental testing of the efficiency of fast projective adaptive algorithms," *Acoustical Physics*, vol. 65, no. 6, pp. 749–764, 2019.

[17] J. A. Velazquez-Iturbide, "GreedEx: a visualization tool for experimentation and discovery learning of Greedy algorithms," *Transactions on Learning Technologies*, vol. 6, no. 2, pp. 130–143, 2013.

[18] C. Bai, S. Xu, J. Duan, B. Jing, M. Yang, and M. Wan, "Pulse-inversion subharmonic ultrafast active cavitation imaging in tissue using fast eigenspace-based adaptive beamforming and cavitation deconvolution," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 64, no. 8, pp. 1175–1193, 2017.

[19] H. Fahmi, M. Zarlis, E. B. Nababan, and P. Sihombing, "Implementation of the Greedy Algorithm to determine the nearest route Search in distributing food production," *IOP Conference Series: Materials Science and Engineering*, vol. 769, pp. 012005–012023, 2020.

[20] H. Fahmi, "An adaptive sparse kernel technique in greedy algorithm framework to simulate an anomalous solute transport model," *Engineering Analysis with Boundary Elements*, vol. 121, pp. 243–254, 2020.

[21] R. Ranjan, N. Agrawal, and S. Joshi, "Interference mitigation and capacity enhancement of cognitive radio networks using modified greedy algorithm/channel assignment and power allocation techniques," *IET Communications*, vol. 14, no. 9, pp. 1502–1509, 2020.

[22] I. C. Olivos-Castillo, R. Menchaca-Mendez, R. Menchaca-Mendez, M. M. Carvalho, and M. E. Rivero-Angeles, "An optimal greedy algorithm for the single access contention resolution problem," *IEEE Access*, vol. 7, pp. 28452–28463, 2019.

[23] D. A. Rosenblueth, "A greedy algorithm for decomposing large clothing orders into lays," *ACM SIGAPP Applied Computing Review*, vol. 19, no. 3, pp. 45–51, 2019.

[24] A. P. P. Kumar, "Greedy algorithm based deep learning strategy for user behavior prediction and decision making support," *Journal of Computer & Communications*, vol. 06, no. 6, pp. 45–53, 2018.

[25] G. A. Widyadana, A. T. Widjaja, and K. J. Wang, "A hybrid greedy algorithm and simulated annealing for single container loading problem: a case study," *Jurnal Teknik Industri*, vol. 20, no. 2, pp. 89–102, 2019.

[26] A. A. B. Homaid, A. Alsweari, and K. Z. Zamli, "Adapting the elitism on the greedy algorithm for variable strength combinatorial test cases generation," *IET Software*, vol. 13, no. 4, pp. 286–294, 2019.

[27] F. L. . Yuan, "Combined 3D thinning and greedy algorithm to approximate realistic particles with corrected mechanical properties," *Granular Matter*, vol. 21, no. 2, pp. 19.1–19.18, 2019.

[28] Y. Zhang, K. Li, and X. Zhang, "Optimising data access latencies of virtual machine placement based on greedy algorithm in datacentre," *International Journal of Computational Science and Engineering*, vol. 18, no. 2, pp. 186–197, 2019.

[29] S. F. Interlenghi, J. L. de Medeiros, and O. D. Q. F. Araújo, "On small-scale liquefaction of natural gas with supersonic separator: energy and second law analyses," *Energy Conversion and Management*, vol. 221, pp. 113117–113124, 2020.

[30] E. Alqudah and A. Jarrah, "Parallel implementation of genetic algorithm on FPGA using Vivado high level synthesis," *International Journal of Bio-Inspired Computation*, vol. 15, no. 2, pp. 90–97, 2020.