Research Article

An Intelligent Animation Interaction Design Algorithm Based on Example and Parameterization

Yingquan Wang, Peng Han, and Bo Yang

1 School of Arts, Qingdao Huanghai University, Qingdao, Shandong 266427, China
2 Qingdao Huanghai University, Qingdao, Shandong 266427, China
3 Department of Secrecy Archive and Screen Culture, Logistics University of PAPF, Tianjin 300300, China

Correspondence should be addressed to Peng Han; hanp@qdhhc.edu.cn

Received 4 April 2022; Revised 9 June 2022; Accepted 23 June 2022; Published 21 July 2022

Copyright © 2022 Yingquan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Intelligent design for the animation interaction is very challenging and needs to be resolved on priority basis. If this design is based on smart learning based mechanism, then it will be more appropriate and effective. Therefore, based on the idea of process modeling, the concrete method is based on example modeling and parameterization, and the VC programming environment is the tool. Animated interactions are not counted. The animation parameterization in the sample animation software and its attribute information are calculated, and the principle and realization process of the new animation interaction design are given, as well as the calculation method and steps of each area of animation elements. The same interaction design pattern can be filled in different areas by using parametric methods. In practice, we have increased the diversity of animation interaction design. The method proposed in this paper can quickly and automatically generate a virtual animation design network according to the user’s design ideas and can be controlled by the user in real time interactively, providing a good interface and a new way for the user to design animation environment modeling.

1. Introduction

Initially, the current advancement in the technology has led to the development and realization of the different systems for different application domains. These systems must be smart enough to make timely decisions which are assumed to be accurate and precise. These decisions would be more accurate and precise if artificial intelligence or computational based techniques have been adopted in the developmental process of the modeling and simulation as well. Parametric animation modeling has been increasingly popular in numerous sectors, including video watching and game production, since the 21st century. The most essential thing to remember when modeling parametric animation is to model parametric animation [1], since the mode of parametric animation has an impact on the visual effect of the animation. Traditional modeling software for large-scale parametric animation modeling is time-consuming and takes a lot of personnel and material resources. The parametric animation process modeling approach is a viable option. The idea of process modeling can be well applied in research, education, and entertainment industries [2, 3]. The main purpose of this paper is to model the idea in terms of a similar process. Using sample-based modeling methods and parametric technology, a set of parametric animation interaction design systems are established, and large-scale parametric animation styles are generated in real time.

This paper proposes an example-based animation interaction design algorithm. Through the extraction of animation interaction point attribute information [4, 5], the original animation style characteristics are maintained to a certain extent, and the average parameterization method is used to map the generated animation interaction to different animations in the area.

For animation interaction and animation modeling, the process modeling approach is employed. The L-system is
used in the majority of existing approaches, and a fixed system is used to drive the model’s progressive creation. It is frequently difficult to add new rules to manage the authenticity of the model due to the strictness and complexity of the rule system. The tensor field technique has recently been offered as a way to assist the design of animation interaction algorithms, although this method has flaws in algorithm transition. With the use of some existing data, example-based modeling technologies may swiftly develop more complicated models. Texture synthesis is where these approaches are most commonly applied, as evidenced by literature [6–8]. Reference [9] suggested a method for generating urban landscapes using sample GIS data, which obtained good results and realized regional transition and white natural connections.

The parametric method of mapping from one surface to another known area was first used for texture mapping. The parameterization process of this paper is based on the idea of the literature [10]. The coordinate mean method is used to express the coordinates of each two-dimensional point as a convex combination of its neighbor points, which is a method of harmonic mapping, which has both length and angle maintained to a certain extent [11, 12].

This article is dedicated to the development of an effective modeling technique or algorithm which is specifically designed for the interaction of the animations. This mechanism could be more impressive if artificial intelligence enabled designed and decision support systems have been developed. Actually, the proposed system is based on the example which is realized through the animation extraction of the point of interaction. Moreover, the proposed model or design has the built-in capacity to maintain various characteristics of the different styles.

The remainder of the paper is organized as follows. The rationale for parameterized implementation with examples is reported in the second section of the paper along with various parameters which are described in the paper. Parametric animation interaction design algorithm and its various steps along with detailed discussion are presented in the third section of the paper. Parametric animation interaction design is discussed in detail in the fourth section of the paper which is followed by the conclusion section.

2. Rationale for Parameterized Implementation with Examples

A typical structure-hinge four-bar mechanism is taken as an example to illustrate the design method of parametric animation.

(1) Determining the parameters that need to be changed: As shown in Figure 1, showing the four-bar mechanism ABCD, the parameters that need to be changed are the length L of the rod AB, the length L2 of the BC, the length L3 of the CD, and the length L4 of the AD [13]. At the same time, from the perspective of control, the speed of animation playback is also an adjustable parameter, and its factor is set to L7.

(2) Determining the overall framework: control of four variable parameters and data display, animation playback, playback control and speed control, the conversion relationship between the size of the animation area and the length of the four rods, and so forth [14]. The overall frame of the four-bar mechanism is shown in Figure 2. The conversion relationship between the size of the animation area and the length of the four rods can be based on the screen resolution.

(3) Programming, including the following:

(1) The dynamic generation of the mechanism;
(2) The control of parameter dragging;
(3) Other judgment knowledge related to the mechanism (whether it is composed of a mechanism, the basic type of the mechanism) and so on. The basic method is to find the coordinates of the four points A, B, C, and D and then connect the four points with a line.

(4) As shown in Figure 1, the four points A, B, C, D are put as symbols into the scene as Move-Clip, named AA, BB, CC, and DD, respectively. Among them, AA is fixed and used as the positioning reference, and the corresponding layer is established for easy control. The key lies in the action layer, as shown in Figure 3.

(5) Initialization preparation and related knowledge preparation. For the four-bar structure, the key lies in whether the four-bar structure is formed at any length and the basic type of the four-bar structure. Frame 1 of the action layer mainly does this work.

The main code on frame 1 of the action layer is as follows:

(i) Initial data preparation:
(ii) Define variables: initial length of four rods _
global.11 = 50; _global.12 = 90; _global.13 = 180; _
global.14 = 180;

(iii) animation initial speed variable _global.17 = 1;

(iv) Four-bar length dynamic display variable 111 = _
global.11; 112 = _global.12; 113 = _global.13; 114 = _
global.14, the four values are realized with dy-
namic text boxes combined with variables.

(v) The initial position of the graphic base reference
point position: root.AA._x = 200;

(vi) root.AA..y = 250;

(vii) scrollbar initial position global. Bar_ _x = 538;

(viii) //Adjust the basic reference point X position of the
data drag slider;

(ix) root.

(x) bar._x = _global.bar._x + int(_global.11–40
- 18)/1.4);

i is the angle between AB and CD; j is the angle
between CD and AD _global.i = 30; as shown in Figure 1.

Function preparation:

1. global. Anguec can find the angle of j according to
the angle of i, _global. Arguec = function(q) {.....

2. global arguei The angle of i can be found according to
the angle of j, _global. Arguei = function(q) .....

3. global. Okj = function( ..... ) judgment

4. global.arbitate = function(a, b, c, d) {///This function
judges the length of the four-bar and returns the
corresponding value
gmin = Math.min (Math.min(a, b), Math.min(c,d));
//Find the minimum of the four values
gmax = Math.max (Math.max(a, b), Math.max(c, d));
//Find the maximum of the four values
if(gmin + gmax >=a+b+c+d) (gmin + gmax)///The
sum of the minimum rod and the maximum rod is
less than or equal to the sum of the other two rods;
if((a = = c)&&((b = = d)) { return 4; //If a = c and
b = d, must be “parallel double crank mechanism”
if(d = gmin) {return2; //When the shortest rod is
used as the frame, it must be a “double crank
mechanism”}
f(a = = gmin | l c = = gmin) {return 1; //Take the
adjacent rod of the shortest rod as the frame, which
must be “crank rocker mechanism”}
if(b = = gmin) {return 3; //When the opposite pole of
the shortest pole is used as the rack, it must be a
“double rocker mechanism”}
else//When the Grashof discriminant is not satisfied,
it must be a “double rocker mechanism” or a group
not, a four-bar mechanism
f(_global.11+_global.12+_global.13 +_global.14) &&(_
global.11+_global.12+_global.13)_global.14)
\{return 3; }
else return 5; }}

Calculate the coordinates of each point and use the line
drawing function to draw the mechanism. Frame 2 of the
action layer mainly does this work.

(i) root.createEmptyMovieClip("me", 1); //Dynamic
creation

(ii) MovieCip, so that the drawing function can be
used.

(iii) result = _global. Arbitrate(_global.11, _global.12,
_global.13)

(iv) al. 13, _global.14); //This function returns the
result according to the length of the four bars.

(v) if(result: =1){

(vi) type = “crank-rocker mechanism”;

(vii) A global.j = _global. Arguec(_ Global. i); //The
following calculation B,

(viii) The coordinates of points C and D;

(ix) _root.BB._x = Math.cos(Math.PI/180 * _global.i)
*x

(x) global.11+ - root.AA..x;
_root., BB._y = _root.AA._y− Math.Sin(Math.PI/180 * a global.i) * - global.11;
_root.DD._x = _root.AA._x + _global.14;
_root.DD._y = _root.AA._y;
_root.CC._x = _root.DD._x + Math.cos(Math.PI/180 * global.i)
* global.13;
_root.CC._y = _root.DD._y - Math.sin(Math.PI/180 * _global.13);
//below is the line drawing
root.me.clear();
with (_root. me) {
lineStyle(4,0 × 330099, 100);
moveTo(_root.DD._x, _root.DD._y);
lineTo(_root.CC._x, _root.CC._y);
lineStyle(2,0 × 339900, 100);
moveTo(_root.AA._x, _root.AA._y);
lineTo(_root.BB._x, _root.BB._y);
lineStyle(2,0 × 339900, 100);
moveTo(_root.BB._x, _root.BB._y);
lineTo(_root.CC._x, _root.CC._y);
(xii) }
(xiii) } f_this draws the four-bar mechanism.

Dynamically change the lengths of the four rods, and display them, borrow the work of the second frame, and draw the mechanism. The third frame of the action layer mainly completes this work.

(i) //The following code refreshes the parameter value
(ii) if(_global. change = true) {
(iii) 111 = _global.11;
(iv) 112 = _global.12;
(v) 113 = _global.13;
(vi) 114 = _global.14;
(vii) - global.change = false;
(viii) //Recalculate the position of each point ......
(ix) if(_global.i ≥ 360) { _global.i = 0; } if rod AB turns - a
(x) week, keep turning.
(xi) _global.i + = _global.17; //The angle rotated by the rod AB each cycle,
(xii) Namely speed control, 17 is the change factor.
(xiii) } Drag the slider to change the action of the parameter. As shown in Figure 3, with the length of AB
(xiv) Take degree L1 as an example. Add the following code to the slider:
(xv) onClipEvent(load) {
(xvi) top = this._y;
(xvii) left = _global.LEFT;
(xviii) night = _global. RICHT;
(xix) bottom = this._y;
(x) onClipEvent(enterFrame) {
if(dragging = = true) {
  r = this._X - left;
}
Double-click the slider to enter the button-level editing. As shown in Figure 2, add the following code:
on(res){
dragging = true;
startDrag("", false, left, top, right, bottom);
root.stop();
barx = _root.bar._x;
lineStyle(2,0 × 339900, 100);
moveTo(_root.AA._x, _root.AA._y);
lineTo(_root.BB._x, _root.BB._y);
lineStyle(2,0 × 339900, 100);
moveTo(_root.AA._x, _root.AA._y);
lineTo(_root.BB._x, _root.BB._y);
lineStyle(2,0 × 339900, 100);
moveTo(_root.BB._x, _root.BB._y);
lineTo(_root.CC._x, _root.CC._y);
(xii) }
(xiii) } } }

Let the sample and parameterized animation move, and make speed control. Frame 4 of the action layer mainly does this work. In fact, the program is between 3 and 4 frames. Repeatedly loop. Action code is gotoAndPlay(3). At this point, the program design is completed. In order to ensure the continuous and smooth animation, the default 12 fps (12 frames per second) can be changed to 30~50 fps during development [15]. The animation interaction design algorithm has passed the debugging of the above program, and the program is very suitable for online communication. If the animation is designed with frame animation, the parameters are fixed and cannot be changed.

3. Parametric Animation Interaction Design Algorithm

In actuality, the majority of animation interactions are not linear. The sample model generated by the mouse is supplied as a straight-line segment for the designer's convenience, and then the system "curves" it, that is, replaces it with a continuous polyline segment [16, 17]. The steps are as follows (Algorithm 1). The starting point and end point of the input straight-line segment are src and dst, respectively, the length of the polyline segment is $d_{SAMPLE}$, and the sampling angle range is $\theta_{DEV}$. These parameters can be set by the user. See the algorithm for the main steps.

The first-level area division is to find the closed part formed between animation interaction and boundary and
between sample and boundary. Algorithms for computing interactions in graph theory are employed.

3.1. Secondary Animation Interaction Design Growth Algorithm. The user decides the mode of each animation interaction design area in the first-level animation interaction design area, and the system generates the second-level animation interaction design using different animation interaction design generation algorithms. Create an animation interaction design connection strategy and parameterization approach between distinct sections so that the secondary animation interaction design is both aesthetically beautiful and dynamic [18].

For each animation interaction design intersection \( v_i \), the attribute information we need to obtain or calculate includes two-dimensional coordinate information \((a, b, c, d)\) in the image.

Levels \( h_a \) and \( h_b \) of the two streets pass through \( v_i \) (assuming at most two streets pass through each intersection, and each street is represented by a continuous polyline section).

On each street passing through \( v_i \), the lengths of adjacent road segments have a mean of \( \mu_d \) and a variance of \( \sigma_d^2 \).

On each street passing through \( v_i \), the mean \( \mu_a \) and variance \( \sigma_a^2 \) of the angle are formed by adjacent road segments.

3.2. Connection of Animation Interaction Design Area. Since the secondary animation interaction design is generated separately in each area, there may be inconsistencies at the boundary of adjacent areas [19]. For this reason, all animation interaction design algorithms in our adjacent areas that intersect the same area boundary are carried out. For judgment, merge the intersections that are too close, so that the animation interaction design between regions can be naturally connected.

The schematic diagram of the traversal algorithm is given, in which probability \( p \) is a binary Gaussian distribution function, and the specific form is given by formula (1), where \( \mu = (\mu_1, \mu_2) \) is the mean and \( \Sigma \) is the covariance matrix.

\[
p(x) = \frac{1}{(2\pi)\Sigma} \exp \left[ -\frac{1}{2}(x-\mu)^\top \Sigma^{-1} (x-\mu) \right]. \tag{1}
\]

It can be seen intuitively from formula (1) that the target node \( v_j \) with the highest probability found in each single-step traversal process has such properties; the design value between \( v_i \) and \( v_j \) is the attribute information \( \mu_\omega^0 \) closest to \( v_j \). The angle is formed by the animation interaction design segment to \( v_j \) and the previous segment. The attribute information \( \mu_\omega^0 \) closest to \( v_j \) is an application example of the traversal algorithm. Among them, the input animation sample is a new animation interaction design generated, and the modeling result maintains the style of the original road to a certain extent.

4. Animated Interaction Design Parametric

The parameterization method used in this article refers to the idea of mean parameterization proposed in the literature. First, all the animation interaction design intersection points in the sample are triangulated, and then each vertex is represented as a convex combination of its neighbour nodes, as in formula (2), where the weight coefficient is calculated according to (4) and (5), and formula (2) satisfies the identity relationship of formulas (2) and (3). The mapping topology in the new animation interaction design technique is created using the matrix calculation method after listing the relational equations between the intersection locations, and the parameterized result is presented.

\[
\sum_{i=1}^{k} \lambda_i v_i = v_0, \tag{2}
\]

\[
\sum_{i=1}^{k} \lambda_i = 1, \tag{3}
\]

\[
\lambda_i = \frac{\omega_i}{\sum_{j=1}^{k} \omega_j}, \tag{4}
\]

\[
\omega_i = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{\|v_i - v_0\|}. \tag{5}
\]

4.1. Calculation of Animation Interaction Design Area. The calculation process of the second-level animation interaction design area is basically similar to the first-level animation interaction design area. These animation interaction designs are further subdivided to obtain the third-level animation interaction design and sample data. When subdividing, we consider the two following practical situations. All animation interaction designs have at least multi-instance support and parametric scale. Animation interaction design traces are mostly represented by polygons, and most of them are quadrilaterals.
The main steps of the subdivision process are as follows: first, is finding the most common probability data of the sample and parameterization and calculating the longer central axis; second is sample on the central axis and the two long sides of the bounding box, respectively, and the average sampling distance Pre-set, allowing the distance between the actual sampling points to float slightly on the basis of the average value; third is connecting the corresponding sampling points to form a three-level animation interaction design [20, 21]. Calculate the closed area to obtain each building footprint, and analyze the results of each algorithm data sample to obtain the corresponding animation interaction design.

In order to improve the controllability of the modeling process and make the modeling effect more realistic, the system has designed an interactive interface, and the designer can modify and adjust a series of parameters. Global parameters and operations include the density of animation interaction design and animation interaction design style or animation interaction design image. Local parameter operations include animation interaction design and algorithm adjustment at all levels, as well as the position movement of animation nodes, deletion, addition of new parameters, etc., and all editing operations can be performed in real time.

5. Conclusion

This work offers an animation interaction design algorithm based on example and parameterization, develops various design techniques for various animation interaction designs, and provides a legality diagnosis strategy as well as an animation division and calculation method for animation interaction. Modeling builds the groundwork and gives a solid user interface while also introducing a new concept in animation interaction design modeling. The following are typical design phases for parametric animation: Determine which parameters should be altered. Establish a conceptual structure, investigate parameter relationships, and then develop code. We have designed parametric animation according to this method in the sample and parametric animation interaction design algorithm and have received good results. Parametric animation is a very good method for learners to conduct self-inquiry learning, because there can be countless changes of this type, and each learner may see different things. From the perspective of actual teaching, it greatly improves learning. At the same time, with a little modification, it can be transformed into a parameter input-type interactive animation, which can turn the learner’s thoughts into reality and achieve better learning and teaching effects.

Data Availability

The datasets used and analyzed during this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This essay is concerned about the special project of art education in Shandong Province: The Practical Research of 5G Technology in Digital Media Art Education Scenes (Y201911084).

References

[1] S. Starke, H. Norman, and J. Zhang, “Memetic Evolution for Generic Full-Body Inverse Kinematics in Robotics and Animation,” IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, vol. 41, 2019.
[2] Xi Wang, “Research on the Visual Language of VR Animation in the Multi-Screen Interactive Era,” in Proceedings of the 2020 INTERNATIONAL CONFERENCE ON VIRTUAL REALITY AND INTELLIGENT SYSTEMS (ICVRIS), Zhangjiajie, China, 18-19 July 2020.
[3] Y. Wu, “Cartoon Image Design System Based on Genetic Optimization Algorithm,” in Proceedings of the 2020 5TH INTERNATIONAL CONFERENCE ON SMART GRID AND ELECTRICAL AUTOMATION (ICSGEA), Zhangjiajie, China, 13-14 June 2020.
[4] Ji Y. Shi, T. Wu, Z. Y. Wen, and L. N. Wang, “Data glove design and virtual hand simulation based on single-chip microcomputer,” in Proceedings of the 2020 INTERNATIONAL CONFERENCE ON COMMUNICATIONS, INFORMATION SYSTEM AND COMPUTER ENGINEERING (CISCE), Kuala Lumpur, Malaysia, 03-05 July 2020.
[5] Li Liang, “Application of Cubic B-Spline Curve in Computer-Aided Animation Design,” COMPUTER-AIDED DESIGN AND APPLICATIONS, vol. 12, 2020.
[6] X. Fan, “Application of style transfer algorithm in the design of animation pattern special effect,” Journal of Physics, vol. 1, 2021.
[7] J. Liang, “Design and Realization of Animation Composition and Tone Space Conversion Algorithm,” COMPLEX, vol. 2021.
[8] X. Ji, “Design and implementation of virtual host based on machine learning algorithm,” Journal of Physics, vol. 1, 2021.
[9] H. Wu, G. Tian, J. Zhang, and Y. Chen, “On Unmanned Aerial Vehicles Light Show Systems: Algorithms, Software and Hardware,” APPLIED SCIENCES, vol. 11, 2021.
[10] H. Wu, S.-J. Wen, and J.-H. Yang, “Multimedia Character Modeling Design and Modeling of Cartoon Animation Based on Bayesian Sequence Recommendation Algorithm,” COMPUTATIONAL INTELLIGENCE AND NEUROSCIENCE, vol. 2022, Article ID 9205227, 2022.
[11] G. Xu, M. Li, M. Bernard et al., “Constructing IGA-Suitable Planar Parameterization from Complex CAD Boundary by Domain Partition and Global/local Optimization,” ARXIV, vol. 328, 2017.
[12] J. E. Arias, J. F. Rubio-Ramirez, and F. Daniel, “Waggoner: Inference Based on Structural Vector Autoregressions Identified with Sign and Zero Restrictions: Theory and Applications,” ECONOMETRICA, vol. 10, 2018.
[13] Y. Baba, “Spectral Cumulus Parameterization Based on Cloud-Resolving Model,” Climate Dynamics, vol. 2, 2018.
[14] T. Bagautdinov, C. Wu, J. Saraghi, F. Pascal, and Y. Sheikh, “Modeling Facial Geometry Using Compositional VAEs,” CVPR, 2018.
[15] P. A. O’Gorman and J. G. Dwyer, “Using Machine Learning to Parameterize Moist Convection: Potential for Modeling of
[16] M. Pan, F. Chen, and W. Tong, "Volumetric Spline Parameterization for Isogeometric Analysis," *ARXIV*, vol. 15, 2019.

[17] R. Galvelis, S. Doerr, J. M. Damas, M. J. Harvey, and G. De Fabritiis, "A Scalable Molecular Force Field Parameterization Method Based on Density Functional Theory and Quantum-Level Machine Learning," *ARXIV*, vol. 12, 2019.

[18] A. Francesco, G K-L. chan, and E. S. gustavo, "Exact Parameterization of Fermionic Wave Functions via Unitary Coupled Cluster Theory," *ARXIV*, vol. 10, 2019.

[19] P. H. S. Coutinho, R. F. Araujo, A. T. Nguyen, and R. M. Palhares, "A Multiple-Parameterization Approach for Local Stabilization of Constrained Takagi-Sugeno Fuzzy Systems with Nonlinear Consequents," *INF. SCI*, vol. 248, 2020.

[20] F. A. Evangelista, G. Kin-Lic Chan, and G. E. Scuseria, "Exact parameterization of fermionic wave functions via unitary coupled cluster theory," *The Journal of Chemical Physics*, vol. 2, 2020.

[21] A. A. Canutescu, L. Dunbrack, and J. Dunbrack, "Cyclic coordinate descent: a robotics algorithm for protein loop closure," *Protein Science*, vol. 12, no. 5, pp. 963–972, 2003.