Optimal Design Problem of Dragon Dance Team Path Based on Rotation Matrix

Xinghan Lu *

Department of Queen Mary University of London Engineering School, Northwestern Polytechnical University, Xi'an, China, 710129

* Corresponding Author Email: luxinghan@mail.nwpu.edu.cn

Abstract. In this paper, an accurate collision detection method is proposed for the collision problem during the traveling of the traditional event - Bench Dragon Dance Team. It needs to consider the collision problem when the Bench Dragons are traveling, but the existing literature has not provided an effective detection means for this specific model. Through geometric analysis, this paper reveals the mechanism of collision occurrence, models the dragon body approximately as a rectangle, and utilizes the rotation matrix transformation for collision detection. In order to further improve the detection accuracy, this paper adopts an iterative search algorithm to optimize and calibrate the model. This paper analyzes the critical termination moment, at this moment, the Dragon Dance Team cannot continue coiling along the isometric spiral. With this model, collisions among the benches can be accurately predicted and avoided. Eventually, when the time t = 566.2 s, the Dragon Dance Team's coiling-in maneuver is completed, and no collision occurs between the benches, at which time the coordinates of the dragon's head position are (0.01, -3.42). The model proposed in this paper effectively solves the collision problem during the traveling process of the bench dragon, and provides a scientific collision detection method for similar scenarios.

Keywords: Collision detection, rotation matrix, iterative search.

1. Introduction

The bench dragon, a traditional folk cultural activity in Zhejiang and Fujian regions, is a activity, where the dragon head leads the body and the tail to travel in different traveling areas and speeds, thus producing different viewing effects.

The activity mainly involves collision detection. The purpose of collision detection is to determine whether different objects collide or overlap with each other, i.e., whether they share the same spatial area. Diao [1] et al. proposed an obstacle avoidance path planning method based on the kinematics and the hybrid enclosing box method that can be used in both dynamic and static situations, which transforms collision detection into solving for the shortest distance. Liu [2] et al. simplified the robotic arm model by adopting a spherical body and capsule enclosing box to enclose obstacles, and according to the intersection between the enclosing box and the encircling box, a robotic arm model can be simplified. According to the intersection between the enclosing boxes to determine whether a collision occurs, but they only consider the static collision problem. Currently, in the use of iterative algorithms to reduce the error, Wang [3] et al. used an iterative search method of the inclusion region to solve the accurate evaluation of the in-plane straightness error. The iterative search of the inclusion region is realized after deciding the initial inclusion region direction, dividing the inclusion region, and calculating the size and direction of the iteration angle. However, the steps are cumbersome, while it is not clear whether the model is stable or not.

Based on the above background, the purpose of this paper is to explore an efficient detection method of the inability to travel of the dragon head (i.e., collision), and to scientifically plan the overall motion path. First, a mathematical geometric approach is used to simplify the model of the problem and construct the geometric relationship between edges and radians; then, collision detection is carried out based on the rotation matrix model and iterative search algorithm, and the calculation of the geometric relationship is incorporated into it. If the conditions are met, the path curve is further optimized by the iterative search method. In order to reduce the difficulty and complexity of collision detection, this paper simplifies the calculation process by determining whether the vertexes of the head enter the specific region delineated by the inverse rotation matrix. In addition, the iterative search model combined with the annealing algorithm effectively reduces the time required for the initial search.

This paper solves the collision detection problem in the queue discing in clockwise motion along the isometric spiral. The collision detection method constructed based on the rotation matrix model is proposed for the first time, which provides an efficient and simple solution idea and optimized algorithm for the detection of complex collision scenes in real life.

In order to better introduce the model proposed, this paper is divided into five chapters to introduce the model building process in detail. The first chapter is the introduction, which introduces the concepts of collision detection and the rotation matrix model; the second chapter is the related theory, which details the collision detection using the rotation matrix model; the third chapter is the experiment, which carries out the building and solving of the model; the fourth chapter is the results, which solves the limiting moments according to the model; and the fifth chapter is the conclusion, which solves the problem of determining the collision moments in the activity, also enriches multiple scenarios of collision detection.

2. Related Theory

This paper studies the collision detection problem in the marching of the Dragon Dance Team, constructs a rotation matrix and an iterative search model. Iterative search, rotation matrix and collision detection are important concepts in computer science and engineering, which are widely used in different fields such as game development, robotics, and computer graphics.

Iterative search: Iterative search is a method of finding a solution to a problem by progressively approximating or repeatedly performing a process [4]. It is commonly used in areas such as optimisation problems (e.g., finding the minimum or maximum value of a function), numerical analysis (e.g., root finding), and so on. In a broad sense, any process that needs to be repeated until a specific condition is satisfied is an iterative search. For example, let f(x) be the objective function, and it is desired to find the x* that makes f(x*) ≤ f(x), x (for a minimisation problem) by iterative search. The update rule for each iteration may be $x_{n+1} = g(x_n)$[5], where g is the update function defined on a case-by-case basis. In this paper, due to the precision problem of floating-point operations in computers, appropriate tolerance processing is required to avoid judgement errors caused by small rounding errors [6]. At the same time, it can reduce the error brought about by the lack of precision in the calculation of time.

Rotation Matrix: The rotation matrix is a special type of square matrix used to represent the rotation of an object in three-dimensional space. Given an angle and an axis of rotation, the position of any point on the object after that rotation can be calculated by constructing the corresponding rotation matrix. A rotation matrix is orthogonal, i.e. its transpose is equal to its inverse [7], and the determinant is 1, ensuring that the rotation does not change the volume of the object. For a rotation on a two-dimensional plane, the rotation matrix R (θ) is expressed as:

$$\mathrm{R}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{1}$$

The matrix of the inverse rotation (i.e., the opposite angle of the rotation θ is - θ) is the inverse of the original rotation matrix. Since the rotation matrix is orthogonal, thus its inverse is equal to its transpose, the inverse rotation matrix is R (-θ):

$$\mathrm{R}(-\theta) = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{2}$$

This paper determines whether the point (x_1, y_1) obtained after reverse rotation is within this rectangle, which can detect whether the collision has occurred.

Collision detection: collision detection refers to the process of determining whether two or more objects have come into contact [8], the basic idea is to compare the positional information of the research objects and determine whether there is an overlap region. In this paper, through the analysis of the collision moment and the analysis of the collision form, the object under study is subjected to a two-dimensional rectangular assumption to find the limit of its non-collision with other dragons within the same conchoidal layer.

The march of the Bench Dragon: the dragon head is in the front, the dragon body and the dragon tail follow each other in a circle, and the whole is in the shape of a disc. The bench dragon consists of 223 sections of benches, with the first section being the dragon's head, the next 221 sections being the dragon's body, and the last section being the dragon's tail. The length of the dragon's head is 341cm, the length of the dragon's body and tail is 220cm, and the width of all the benches is 30cm. There are two holes in each section of the bench, the diameter of the holes is 5.5cm, and the center of the holes is 27.5cm away from the head of the nearest bench. Adjacent benches are connected by handles which are inserted into the holes. For the sake of unity and convenience of calculation, the position and speed of the Dragon Dance Team in this paper refer to the position and speed of the dragon head, the front handles of each section of the dragon body and the center of the rear handles of the dragon tail.

3. Experiment

In this experiment, the collision is divided into collision moment and collision form. After determining the moment and form of the collision, the rotation matrix model is established, combined with the use of iterative search to reduce the error of time calculation, so as to achieve the prediction of the Dragon Dance Team just does not collide, and to achieve the positioning of the route of the Dragon Dance Team. It is known that each handle is located on the isometric spiral, the pitch is 55cm, the dragon head is located in the 16th circle of the screw thread to disc in clockwise, and the travelling speed of the handle in front of the dragon head is always 1m/s.

The flow of this experiment is shown in Figure 1:

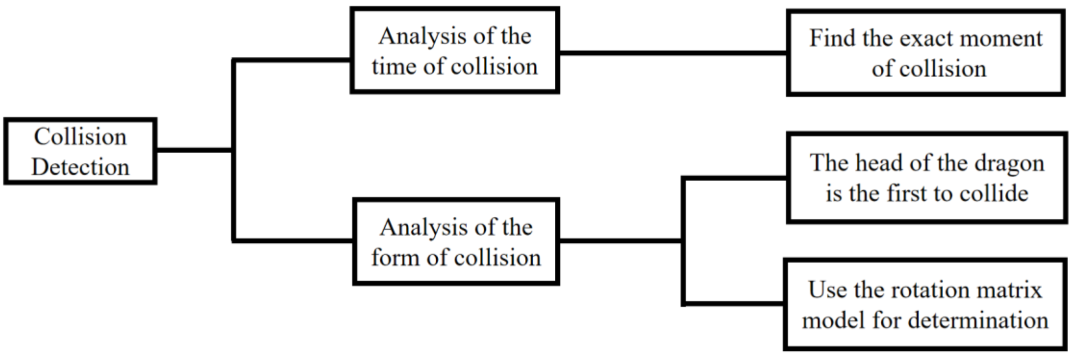

Figure 1. Flow chart of collision detection.

3.1. Analysis of the time of collision

The limiting moment of no collision between benches is also the moment at which a collision between benches happens to occur.

3.2. Analysis of the form of collision

Because every section of the dragon body after the dragon head will continue the path that the dragon head has travelled (but at different speeds), thus the first part that may collide is the dragon head. So this paper will consider the situation where the dragon head collides with the dragon body.

It is worth noting that the actual situation of benches needs to be taken into account [9], because when the collision is about to take place, it is one of the two vertices at the front of the dragon's head that collides with the dragon body. Thus, it is necessary to consider the actual situation of benches to construct a suitable geometric and mathematical model to determine whether the vertices of the dragon head, enter into the rectangular region enclosed by the four vertices of the dragon body.

3.3. Establishment of right-angle coordinates of rectangular vertices

Since it is unlikely that the dragon head will collide with its neighbouring body (the 1st section of the dragon body after the dragon head), so it is assumed that there may be a critical case of collision starting from the 2nd section of the dragon body.

Consider the bench on which the dragon body is located as a rectangle with a length L of 220 cm and a width M of 30 cm, and assume the four vertices of the rectangle are A, B, C, and D, where $M(x_M, y_M)$ is the center of the rectangle. In order to determine the positional coordinates of M, it is necessary to first find the positional coordinates where the front and rear handles of the bench are located, and let the front handle positional coordinates be(x_I, y_i), then the position of the rear handle is(x_{i+1}, y_{i+1}). Determine the positional coordinates of the point M based on the midpoints of the front and rear handle positional coordinates:

$$\begin{cases} x_M = \dfrac{x_i + x_{i+1}}{2} \\ y_M = \dfrac{y_i + y_{i+1}}{2} \end{cases} \tag{3}$$

The line joining the locations of the front handle (x_I, y_i) and the rear handle (x_{i+1}, y_{i+1}) constitutes the direction of the bench in the lateral (L-direction) at this moment in time, which in turn is normalised to a unit vector given the direction vector$\vec{v} = (v_x, v_y)$:

$$\vec{u} = (u_x, u_y) = \left(\frac{v_x}{\sqrt{{v_x}^2 + {v_y}^2}}, \frac{v_y}{\sqrt{{v_x}^2 + {v_y}^2}}\right) \tag{4}$$

Further represent the unit vector in the vertical direction:

$$\overrightarrow{u_\perp} = (-u_y, u_x) \tag{5}$$

Thus, the coordinates of each vertex of the rectangle can be calculated. The unit vectors and their perpendiculars can represent the directions of the long and wide edges of the rectangle, respectively. Setting the length of the rectangle to be L and the width to be W, the coordinates of A, B, C, and D can be found:

$$A = \left(x_M - \frac{L}{2}u_x - \frac{W}{2}(-u_y), y_M - \frac{L}{2}u_y - \frac{W}{2}u_x\right) \tag{6}$$

$$B = \left(x_M + \frac{L}{2}u_x - \frac{W}{2}(-u_y), y_M + \frac{L}{2}u_y - \frac{W}{2}u_x\right) \tag{7}$$

$$D = \left(x_M - \frac{L}{2}u_x + \frac{W}{2}(-u_y), y_M - \frac{L}{2}u_y + \frac{W}{2}u_x\right) \tag{8}$$

$$C = \left(x_M + \frac{L}{2}u_x + \frac{W}{2}(-u_y), y_M + \frac{L}{2}u_y + \frac{W}{2}u_x\right) \tag{9}$$

In turn, each side of the rectangle is represented:$\overrightarrow{AB}, \overrightarrow{AD}$ etc. Use $\overrightarrow{AB}, \overrightarrow{AD}$ to calculate the angle θ.

3.4. Creation of Rotation Matrix and Inverse Rotation Matrix

For rotations in the two-dimensional plane, the rotation matrix $R(\theta)$ is shown as:

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{10}$$

The matrix of the inverse rotation (i.e., the opposite angle of the rotation θ is - θ) is the inverse of the original rotation matrix, and since the rotation matrix is orthogonal, and thus its inverse is equal to its transpose, the inverse rotation matrix is R (-θ):

$$\mathrm{R}(-\theta)=\begin{pmatrix}\cos(-\theta) & -\sin(-\theta)\\ \sin(-\theta) & \cos(-\theta)\end{pmatrix}=\begin{pmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{pmatrix} \tag{11}$$

3.5. Determining whether a point after inverse rotation is inside the rectangle

Determine whether the point (x_1, y_1) obtained after inverse rotation is inside this rectangle. The specific steps are as follows:

Step1: Represent the minimum and maximum coordinate values of the four vertices of the rectangle after inverse rotation. First assume that the rectangle is parallel to the coordinate axes, i.e., the boundaries are horizontal and vertical, respectively. Next set the coordinates of the lower-left and upper-right corners of the matrix to be (x_{min}, y_{min}) and(x_{max}, y_{max}), respectively.

Step2: Apply the inverse rotation matrix: convert the point (x_1, y_1) (here it should be the rotated point, but for the convenience of explanation, assume it has been given and needs to be inverse rotated back to the original coordinate system) back to the point (x,y) in the original coordinate system through the inverse rotation matrix, and compute:

$$\begin{cases} x = x_1\cos\theta + y_1\sin\theta \\ y = -x_1\sin\theta + y_1\cos\theta \end{cases} \tag{12}$$

Step3: Determine whether the point is inside the rectangle: compare the obtained (x,y) coordinates with the boundary of the rectangle[10], if:

$$\begin{cases} x_{min} \le x \le x_{max} \\ y_{min} \le y \le y_{max} \end{cases} \tag{13}$$

Then the point (x,y) (i.e., the point after inverse rotation) is inside the rectangle; otherwise, the point (x,y) is not inside the rectangle.

Step4: If the dragon head does not collide with the 2nd section of the body, then continue to retrieve the 3rd section of the body and so on. Since collision exists only within the same spiral layer. If the dragon head collides with the dragon body of section i, it is necessary to ensure that the radian formed by the front handle of the dragon body of section i and the collision point of the dragon head (one of the first two vertices) is $< 2\pi$. If the radian is $\geq 2\pi$, it is proved to be beyond the same spiral layer, and thus no more iterations are performed.

3.6. Iterative Search Modelling and Error Reduction

Considering the accuracy of floating-point arithmetic, appropriate tolerance processing may be required to avoid judgement errors due to small rounding errors. In addition, an iterative search model should be used to prevent the calculation of time from being less accurate.

In this case, it is necessary to derive the velocity of the dragon body using the associated velocity formula of the interconnected rigid bodies:

$$v_2 = \frac{r_2}{r_1} v_1 \tag{14}$$

After getting the first termination time, an iterative search model was used to continue the search and correct the model to finally determine the termination time for the Dragon Dance Team to advance.

After the above experiments and research analyses, the rotation matrix and iterative search model are established, experiments are conducted, and results are obtained as shown below.

4. Results

In this paper, practical problems in isometric spirals are solved by rotation matrix model, iterative search and mathematical models of rigid body associated velocities. The position and velocity of each handle are calculated under the velocity of dragon head determined, the critical collision situation is found by collision point detection, and the model is subjected to sensitivity analysis and error detection, which ultimately allows for the evaluation and generalization of the model introduced in this paper.

After analyzing the geometric relationships, rotation matrix modelling and iterative searching for experimental derivations, it is concluded that when t=566.2s, the Dragon Dance Team just stops without making the benches collide, and at this time, the coordinates of the position where the dragon head is located are (0.01, -3.42). The positions and velocities of several sections of the dragon body and the dragon tail are derived at the termination moment based on the connection-velocity relationship of the rigid body, as shown in Table 1:

Table 1. Results of position and velocity at the moment of termination.

	Horizontal coordinate x(m)	Vertical coordinate y(m)	Velocity(m/s)
The front handle of dragon head	0.19546316	-3.161868556	1
The front handle of 1st dragon body	2.525456883	-2.030838383	1.022983982
The front handle of 51st dragon body	4.074915769	2.904679942	1.579658956
The front handle of 101st dragon body	-5.726998942	2.595629588	1.98482968
The front handle of 151st dragon body	-4.773767685	5.587899558	2.319952019
The front handle of 201st dragon body	7.928423551	2.371285887	2.612275558
The rear handle of dragon tail	-1.088445104	-8.582715812	2.73097185

The following shows the distribution of each point of the Dragon Dance Team (dragon head, dragon body, dragon tail handles) at the termination moment t=566.2s, at this time, the coordinates of the dragon head's position are (0.01, -3.42), and the distribution of each point is shown in Figure 2

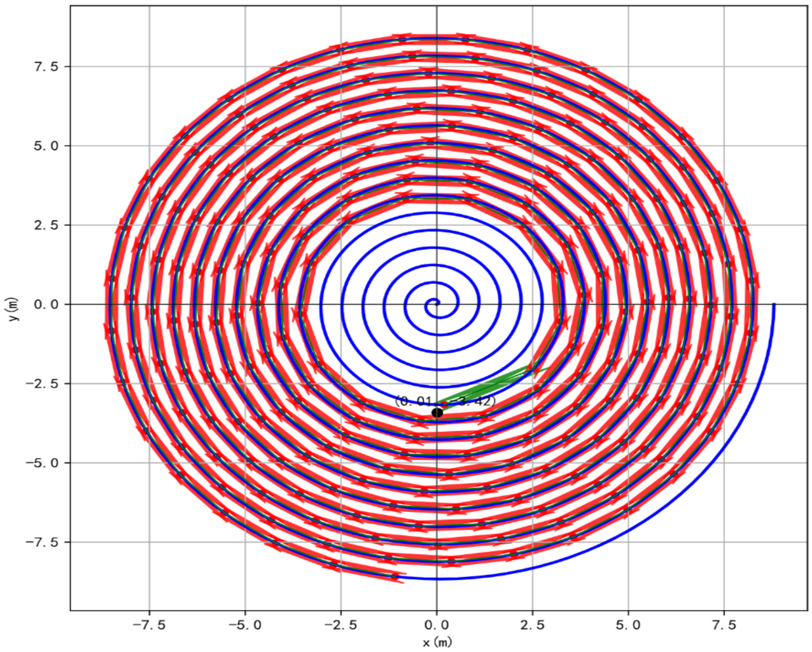

Figure 2. Distribution of points at the termination moment (t=566.2s).

According to the results, when t=566.2s, the front left vertex of the dragon head happens to collide with the dragon body located at (0.01, -3.42), which indicates that a vertex of the rectangle where the dragon head is located happens to fall within the area delineated by the inverse rotation matrix, which is the critical point for no collision, and is also the limiting moment when the Dragon Dance Team can no longer continue to move forward.

The experimental results show that the use of the rotation matrix combined with the iterative search model can effectively improve the accuracy and efficiency of collision detection. The convergence of the iterative search algorithm makes the real-time collision detection guaranteed, while the accuracy of the rotation matrix in dealing with the simulation of the object collision scene significantly improves the reliability of the detection.

By comparing the results of different methods, it is found that both the rotation matrix and the iterative search model show superior performance in a variety of experimental settings, especially when facing high-frequency object collisions, the response time of the model is significantly lower than that of the other methods. In addition, the present model can be improved to make it more regular in calculating the speed of different taps, improve the accuracy of the search and reduce the response time of the model.

5. Conclusion

In this paper, how to effectively avoid the collision problem between the dragon head and the dragon body during the Dragon Dance Team's clockwise discing in along the isometric spirals is studied. Rotation matrix and iterative search model are used for collision detection, and the motion trajectory and speed of the Dragon Dance Team are deeply analysed. A collision detection model is firstly established to monitor the relative position and motion status between the handles of the Dragon Dance Team. By setting the rotation matrix and inverse rotation matrix, the iterative search is able to determine whether the current position of the dragon head will lead to a collision. After many experiments and simulations, the reliability of the rotation matrix model as well as the collision detection method is verified.

Through the above research, this paper not only solved the problem of dragon head collision, but also provided effective technical support for future Dragon Dance Team performances. The successful application of this model provides new ideas for collision detection in related fields, they can be extended to other types of practical problems in reality, and through the simulation and operation of the model, and human and financial resources can be reduced to avoid huge losses caused by collisions.

References

[1] DIAO Hui, CHEN Gui, CAO Feihu, et al. Obstacle avoidance path planning of double manipulator based on collision detection [J]. Machine Tool & Hydraulics, 2024, 52(15): 43-49.

[2] LIU Jianchun, QIN Kun, LIN Yanfeng, et al. Research on Collision Detection Algorithm of Dual Manipulator [J]. Mechanical Transmission, 2021, 45(01): 40-44+70.

[3] WANG Can, XU Bensheng, HUANG Meifa, et al. A Zone iterative search for error evaluation of planar straightness [J]. Manufacturing Automation, 2015, 37(10): 4-7.

[4] XIAO Wanxia, DONG Xingye, LIN Youfang. An Iterated Local Search Algorithm for Aircraft Recovery Problem [J]. Computer and Modernisation, 2019, (09): 1-6.

[5] Kong Y F. An improved iterative local search algorithm for the regionalization problem [J]. Journal of Geo-information Science, 2022, 24(09): 1730-1741.

[6] [6] HU Xiaoyang, YAO Xifan, HUANG Peng, et al. Improved iterative local search algorithm for solving multi-AGV flexible job shop scheduling problem [J]. Computer Integrated Manufacturing Systems, 2022, 28(07): 2198-2212.

[7] Wu Hua, Shao GuangZhou. Singular Value Decomposition of Matrix and Generalized Inverse Solution of Coefficient Matrix of Inhomogeneous Linear Equations [J]. College Mathematics, 2024, 40(02): 81-86.

[8] WANG Shunchao, LI Zhibin, CAO Qi, et al. Reciprocal velocity obstacle algorithm for collision risk avoidance of intelligent connected vehicles [J]. Journal of Traffic and Transportation Engineering, 2023, 23(05): 264-282.

[9] WAN Yanying, WU Qilin, CAO Zhongjia, et al. Cobot Collision Detection Based on Momentum Observer and Dual Encoders [J]. Machine Tools & Hydraulics, 2023, 51(15): 83-87.

[10] WANG Li, YU Youpeng, HE Jian, et al. Collision Detection Method Based on Hierarchical Bounding Box of Sheet Metal Bending [J]. Mechanical Manufacturing and Automation, 2023, 52(04): 185-188.