

# 基于 Babylon.js 的 $n$ 阶 Bézier 曲线轨迹设计

徐爱群,王名宏

(浙江科技学院 机械与能源工程学院,杭州 310023)

**摘要:**【目的】为在 Babylon.js 环境下实现三维物体任意运动轨迹的生成,给出一种三维物体  $n$  阶 Bézier 曲线运动轨迹的设计方法。【方法】首先根据 Bézier 曲线的原理设计一种循环递归的插值计算方法,对任意中间控制点数的 Bézier 曲线路径都可快速绘制;然后对比二阶、三阶及六阶 Bézier 函数的实际绘制效果;最后在 Babylon.js 的官方训练场中对物体其他阶次 Bézier 曲线轨迹进行绘制和运动仿真,并对轨迹形状进行可视化交互设计与调整。【结果】试验中本文方法与传统方法的曲线轨迹重合,表明本文算法实现了  $n$  阶 Bézier 曲线轨迹的绘制;小球能绕所设计轨迹运动验证了本文方法在物体轨迹设计上的有效性;轨迹调整前后对比结果验证了本文方法可视化交互设计的可行性。【结论】本文方法能有效快速实现虚拟环境中的三维物体复杂运动路径的设计与生成,从而为采用 Bézier 曲线来规划网页端三维物体的运动轨迹提供了一种新方法。

**关键词:** Babylon.js; Bézier 曲线; 轨迹生成; 可视化交互设计

**中图分类号:** TP391.7      **文献标志码:** A      **文章编号:** 1671-8798(2023)01-0040-08

## Trajectory planning of $n$ -order Bézier curve based on Babylon.js

XU Aiqun, WANG Minghong

(School of Mechanical and Energy Engineering, Zhejiang University of  
Science and Technology, Hangzhou 310023, Zhejiang, China)

**Abstract:** [Objective] In order to realize arbitrary trajectory generation of a 3D object in the environment of Babylon.js, a trajectory planning method of  $n$ -order Bézier curve was proposed on the part of a 3D object. [Method] First, a recursive interpolation algorithm was designed according to the principle of Bézier curve, capable of quickly drawing the path of Bézier curve with any intermediate control points; then, a comparison was made among the actual drawing effects of second-order, third-order and sixth-order Bézier functions; finally, Bézier curve trajectories of other orders of the object were drawn and simulated in the official training ground of Babylon.js, carrying out visual interaction design and adjustment of the trajectory shapes. [Result] In the experiment, the curve trajectories of the proposed method coincide with those of the traditional method, which shows that the algorithm can draw the trajectory of  $n$ -order Bézier curve; the ball can move around the designed trajectory, which verifies the effectiveness

收稿日期: 2021-11-21

基金项目: 浙江科技学院研究生教学改革研究项目(2020yjsjg01); 浙江科技学院研究生科研创新基金(2020yjskc07)

通信作者: 徐爱群(1968—),男,浙江省龙游人,教授,博士,主要从事微机电系统研究。E-mail: zdxq@163.com。

of the proposed method in trajectory planning of the object; the comparison results before and after trajectory adjustment verify the feasibility of the visual interaction design of the proposed method. [Conclusion] This method can effectively and quickly realize the design and generation of complex motion paths of a 3D object in the virtual environment, providing a new method for using Bézier curve to plan the motion trajectories of the 3D object on web pages.

**Keywords:** Babylon.js; Bézier curve; trajectory generation; visual interaction design

随着网页技术和虚拟现实技术的快速发展,基于 Web3D<sup>[1]</sup> 技术进行网页端虚拟现实开发越来越流行。WebGL(web graphics library,网络图形库)源于 OpenGL ES 2.0(open graphics library for embedded systems 2.0,嵌入式系统开放图形库 2.0)标准,可使用 HTML5(hyper text markup language 5.0,超文本标记语言 5.0)的 Canvas 元素作为 DOM(document object model,文件对象模型)接口访问的一种实现 Web 交互式三维图形渲染技术<sup>[2-3]</sup>,具有无插件、跨平台、可直接驱动硬件加速等优势<sup>[4]</sup>。WebGL 因其上述优点受到广大厂商的支持,进一步衍生出一些优秀的渲染引擎,发展于 WebGL 的 Babylon.js 引擎因其强大的渲染功能已成为目前网页端虚拟现实开发选择的主流三维引擎<sup>[5-6]</sup>,目前在 Babylon.js 等三维虚拟引擎中实现物体运动轨迹的规划要考虑重力、摩擦力等物理条件,才能在视觉上体现三维物体运动的真实性,实现起来较困难,会受到很多因素的干扰与制约。为简单高效地实现三维物体复杂的曲线运动轨迹,可在虚拟引擎中使用 Bézier 曲线提前绘制出合适的目标轨迹,再让物体沿此轨迹运动。在现实生活中,Bézier 曲线已被运用于物体的路径规划中,Han 等<sup>[7]</sup>提出了一种基于 Bézier 曲线的路径规划器,能形成最优避障路线,实现了汽车的防碰撞行为。杨彬等<sup>[8]</sup>运用三阶 Bézier 曲线对无人驾驶车辆起始点至终点之间的轨迹进行拟合,通过对 Bézier 曲线控制点参数进行合理的赋值,可实现对拟合轨迹的局部调整。Tharwat 等<sup>[9-10]</sup>采用 Bézier 曲线的路径规划模型,实现了对移动机器人的最优平滑路径规划。已有一些研究将 Bézier 曲线应用于虚拟现实,如孙晨阳<sup>[11]</sup>在虚拟现实利用双三次 Bézier 曲面和多边形对小麦各个器官进行建模,可以以低成本和较高的效率模拟小麦生长过程,模型表面光滑且可以实现小麦不同生长期的建模;Wu 等<sup>[12]</sup>将 Bézier 曲线算法与三维物体的网格顶点相结合,实现对三维物体的杆件弯曲。综上所述,将 Bézier 曲线运用于虚拟现实三维物体的轨迹设计是可行的,但目前虚拟现实应用所涉及的曲线形状相对比较简单,只需通过固定阶次 Bézier 曲线即可实现绘制,灵活性较弱,且在 Web3D 技术上 Bézier 曲线应用极少。

通过物体运动原理及数学公式确定物体运动轨迹的方法给虚拟现实网页端开发人员的图形设计带来不便,其原因是:原理不易理解且参数过多;复杂无规律轨迹使用数学公式也难以快速描述,且轨迹调整需重新进行计算。而 Bézier 曲线通过调节控制点的数量和位置就可快速完成轨迹的绘制与调整,故研究基于 Babylon.js 引擎对复杂轨迹进行可视化设计并进行运动仿真很有必要。目前,Babylon.js 环境中仅有二阶和三阶 Bézier 曲线的实现方法,并没有设计任意阶次的实现方法。将 Bézier 曲线灵活应用于 Babylon.js 环境下的虚拟三维物体轨迹设计,需进行算法设计和 Typescript 编程,以最终实现对三维物体运动轨迹的可视化交互控制。为此,我们设计了一种基于 Babylon.js 的  $n$  阶 Bézier 曲线的绘制方法,可自定义中间控制点的数量和位置,从而实现三维物体复杂运动轨迹的绘制。

## 1 Bézier 曲线定义

法国工程师 Pierre Bézier 在 1962 年提出了一种通过连接向量来表示曲线的方法,这样的曲线被称为 Bézier 曲线。由于表达直观、结构简单,Bézier 曲线是计算机辅助几何设计中表示曲线和曲面的重要工具<sup>[13]</sup>,常被用来绘制复杂的图形形状<sup>[14]</sup>和模型样式<sup>[15]</sup>。使用 Bézier 曲线方法绘制曲线方便快捷,只要给定曲线的起点和终点,然后通过调节中间点的数量及位置来控制曲率,就可得到目标曲线。

在空间中给定  $n+1$  个控制点  $P_i, i=0, 1, 2, \dots, n$ , 则  $n$  阶 Bézier 曲线可描述如下:

$$P(t) = \sum_{i=0}^n P_i B_{i,n}(t), t \in [0, 1]. \quad (1)$$

式(1)中; $t$  为独立变量; $B_{i,n}(t)$  为第  $i$  个  $n$  阶 Bernstein 多项式,

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}。 \quad (2)$$

Bézier 曲线求导公式如下:

$$P'(t) = \sum_{i=0}^n n[B_{i-1,n-1}(t) - B_{i,n-1}(t)]P_i = n \sum_{i=1}^n B_{i,n-1}(t)(P_{i+1} - P_i)。 \quad (3)$$

## 2 方法的设计与实现

### 2.1 方法描述

在绘制三维物体运动轨迹的实际应用中,针对不同需求往往需运用不同阶次的 Bézier 曲线,开发者只需改变函数中的参数即可绘制不同阶次的曲线轨迹,目前在 Babylon.js 环境中已实现二阶、三阶 Bézier 曲线轨迹的绘制,并封装为接口供用户使用,但其他阶次 Bézier 曲线还未实现。本研究给出一种  $n$  阶 Bézier 曲线运动轨迹自动生成方法,其核心思想是设计 getBezierArr 函数来生成以各控制点位置为基础的 Bézier 曲线轨迹;其解决思路是设计一种循环递归的计算方法来完成 Bézier 曲线轨迹点的自动插值。首先,在基于 WebGL 网页端开发中,在 Babylon.js 环境下运用 Typescript 语言进行程序编写,以此实现 getBezierArr 函数的基本功能;然后,在起点、终点和中间控制点相同的情况下,比较 Babylon.js 环境中采用 BABYLON.Curve3.CreateQuadraticBezier 和 BABYLON.Curve3.CreateCubicBezier 函数所生成的曲线与本文方法所生成的 Bézier 曲线的一致性;再设计 transformNode 方法让小球沿所生成曲线轨迹运动,验证三维物体在所生成轨迹上运动的可行性;最后,以 getBezierArr 函数的曲线拟合功能为基础,通过鼠标操作控制点位置,实时更新曲线轨迹的形状,以完成曲线的可视化交互设计。

### 2.2 函数设计

使用 getBezierArr 函数获取组成 Bézier 曲线路径的插值线段点集合,它的参数为 getBezierArr(ctrlPosArr:Array<BABYLON.Vector3>,precision:number)。ctrlPosArr 表示由起点、终点和所有中间控制点组成的数组,precision 表示通过多少个顶点来描述 Bézier 曲线,这些点随式(1)中参数  $t$  的变化依次均匀分布在 Bézier 曲线的路径上。

将式(2)进行变形:

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i} = (C_{n-1}^i + C_{n-1}^{i-1})t^i (1-t)^{n-i} = (1-t)C_{n-1}^i t^i (1-t)^{(n-1)-i} + tC_{n-1}^{i-1} t^{i-1} (1-t)^{(n-1)-(i-1)} = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t)。 \quad (4)$$

式(4)表明: $n$  阶 Bernstein 多项式是  $n-1$  阶 Bernstein 多项式的线性组合,具有递归性。

把式(4)代入式(1)展开得:

$$P(t) = \sum_{i=0}^n P_i B_{i,n}(t) = \sum_{i=0}^n P_i [(1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t)] = \sum_{i=0}^n [(1-t)P_i + tP_{i+1}]B_{i,n-1}(t)。 \quad (5)$$

由此可推出 Bézier 曲线的递推公式如下:

$$P(t) = P_0^n; \quad (6)$$

$$P_i^k = \begin{cases} P_i, k=0; \\ (1-t)P_i^{k-1} + tP_{i+1}^{k-1}, k=1, 2, \dots, n。 \end{cases} \quad (7)$$

式(6)、(7)中: $P_0^n$  为首个  $n$  次 Bézier 曲线迭代多项式; $P_i^k$  为第  $i+1$  个  $k$  次 Bézier 曲线迭代多项式, $i=0, 1, \dots, n-k$ 。

Bézier 曲线递推计算公式可理解为每相邻两个控制点插值出新的控制点,相邻的两个新控制点再插值出新的控制点,循环  $n$  次后得出最终的  $P(t)$  插值点,下面对一阶、二阶、三阶 Bézier 公式进行分析。

对一阶 Bézier 公式展开得

$$P_0^1 = P_0(1-t) + P_1 t。 \quad (8)$$

对二阶 Bézier 公式展开得

$$P_0^2 = P_0(1-t)^2 + P_1 \times 2(1-t)t + P_2 t^2。 \quad (9)$$

逐级简化分解后得:

$$P_0^2 = P_0^1(1-t) + P_1^1 t; \quad (10)$$

$$\begin{cases} P_0^1 = P_0(1-t) + P_1 t; \\ P_1^1 = P_1(1-t) + P_2 t。 \end{cases} \quad (11)$$

对三阶 Bézier 公式展开得

$$P_0^3 = P_0(1-t)^3 + P_1 \times 3(1-t)^2 t + P_2 \times 3(1-t)t^2 + P_3 t^3。 \quad (12)$$

逐级简化分解后得:

$$P_0^3 = P_0^2(1-t) + P_1^2 t; \quad (13)$$

$$\begin{cases} P_0^2 = P_0^1(1-t) + P_1^1 t; \\ P_1^2 = P_1^1(1-t) + P_2^1 t; \end{cases} \quad (14)$$

$$\begin{cases} P_0^1 = P_0(1-t) + P_1 t; \\ P_1^1 = P_1(1-t) + P_2 t; \\ P_2^1 = P_2(1-t) + P_3 t。 \end{cases} \quad (15)$$

从一阶、二阶、三阶 Bézier 曲线的逐级简化分解公式推导可以发现,这是一个逐级循环递归的过程,每经过一次循环计算,控制点将会减少一个,直到最后只剩 2 个控制点,即可完成  $t$  时刻  $n$  阶插值点的计算。对  $n$  阶 Bézier 公式可利用计算机进行如下操作,先计算一阶的控制点  $P_i^1$  ( $n$  个),再计算二阶的控制点  $P_i^2$  ( $n-1$  个),经过  $n-1$  次逐级循环后计算  $n-1$  阶的控制点  $P_i^{n-1}$  (2 个),最后第  $n$  次循环计算出  $n$  阶的控制点  $P_0^n$ 。

### 2.3 方法的实现

不同场景中不同阶次 Bézier 曲线运动轨迹的使用都可通过以下步骤实现。首先,以循环递归的插值方法实现某一位置处插值点坐标值的求取;然后,根据所需精度,即插值点的数量,依次计算轨迹上所有插值点的坐标值,形成一个 Bézier 曲线轨迹点集合;最后,设置三维物体的位置,使物体依次经过轨迹点集合中的每个点。 $n$  阶 Bézier 曲线算法流程如图 1 所示。

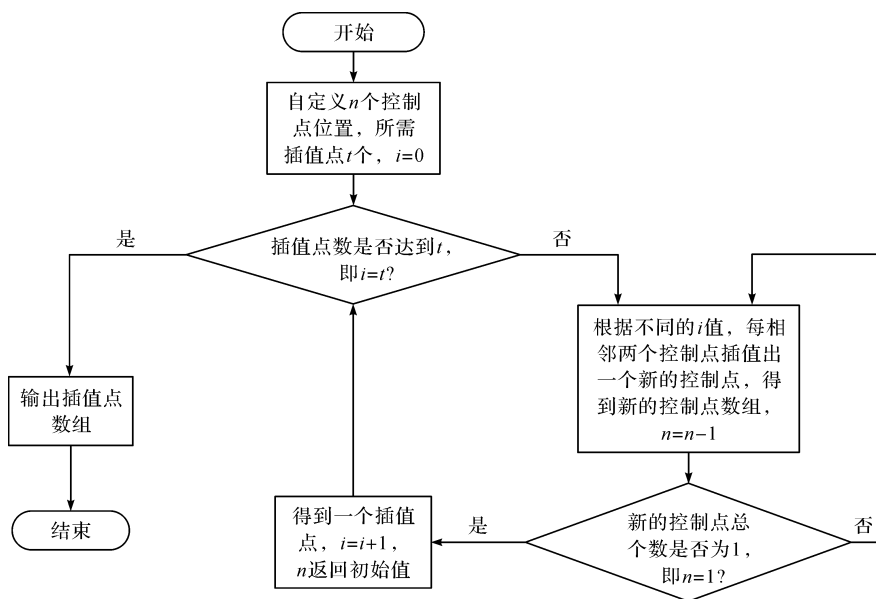


图 1  $n$  阶 Bézier 曲线算法流程

Fig. 1 Algorithm flow of  $n$ -order Bézier curve

### 3 试验与对比分析

在 Babylon.js 环境下,使用 BABYLON.Mesh.CreateLines 方法绘制 Bézier 曲线。CreateLines 主

要用于三维世界中线段的渲染,可在线段绘制时进行很多操作,比如设置宽度、颜色。在 Babylon.js 环境下对比采用本文方法生成的  $n$  阶 Bézier 曲线与内置方法生成 Bézier 曲线的一致性,来验证设计方法的正确性与稳定性。因为内置方法并没有实现  $n$  阶 Bézier 曲线的生成方法,只有 CreateQuadraticBezier(二阶)和 CreateCubicBezier(三阶)两种方法,所以本研究选择  $n=2$ 、 $n=3$  时的  $n$  阶 Bézier 曲线生成方法与内置二阶、三阶进行对比;最后与经典六阶 Bézier 曲线进行对比以验证本文方法高阶的有效性,为后续小球仿真试验做铺垫。对于三维空间中的一组顶点,CreateLines 方法可使相邻两个顶点之间渲染生成一条直线,当顶点数量较多时,可生成一条相对平滑的曲线。因此,将 getBezierArr 方法计算出的曲线插值点坐标数组代入 CreateLines 中生成所需的目标曲线。

### 3.1 二阶 Bézier 曲线对比

当  $n=2$  时,设置相同的顶点数和 Bézier 曲线控制点集合,然后将本文 getBezierArr 方法与内置 CreateQuadraticBezier 方法进行对比,为了有明显的视觉对比效果,设置了不一样的曲线样式。二阶 Bézier 曲线试验参数见表 1。通过表 1 的点集绘制生成二阶 Bézier 曲线,本文 getBezierArr 方法用绿色线条,内置 CreateQuadraticBezier 方法用一系列连续的点表示,二阶 Bézier 曲线轨迹效果对比如图 2 所示。

表 1 二阶 Bézier 曲线试验参数  
Table 1 Experimental parameters of second-order Bézier curve

参数	getBezierArr	CreateQuadraticBezier
顶点数/个	25	25
起点坐标/mm	(0,0,0)	(0,0,0)
终点坐标/mm	(50,30,10)	(50,30,10)
控制点坐标/mm	(20,50,0)	(20,50,0)

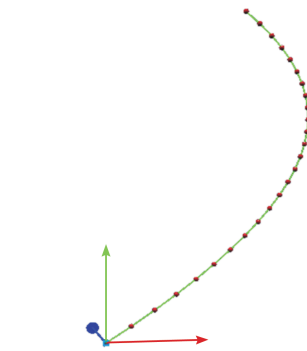


图 2 二阶 Bézier 曲线轨迹效果对比

Fig. 2 Comparison of trajectory effect of second-order Bézier curve

### 3.2 三阶 Bézier 曲线对比

当  $n=3$  时,设置相同的顶点数和 Bézier 曲线控制点集合,将本文 getBezierArr 方法与内置 CreateCubicBezier 方法进行对比,三阶 Bézier 曲线试验参数见表 2。通过表 2 的点集绘制生成三阶 Bézier 曲线,本文 getBezierArr 方法用绿色线条,内置 CreateCubicBezier 方法用一系列连续的点,三阶 Bézier 曲线轨迹效果对比如图 3 所示。

表 2 三阶 Bézier 曲线试验参数  
Table 2 Experimental parameters of third-order Bézier curve

参数	getBezierArr	CreateCubicBezier
顶点数/个	60	60
起点坐标/mm	(0,0,0)	(0,0,0)
终点坐标/mm	(-30,60,30)	(-30,60,30)
控制点 1 坐标/mm	(30,30,40)	(30,30,40)
控制点 2 坐标/mm	(-60,50,-40)	(-60,50,-40)

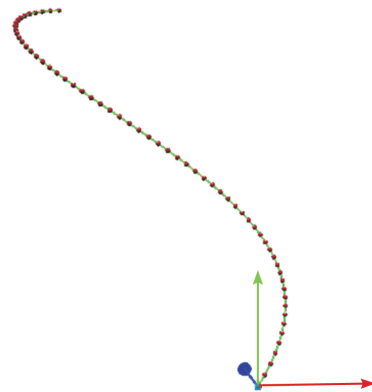


图 3 三阶 Bézier 曲线轨迹效果对比

Fig. 3 Comparison of trajectory effect of third-order Bézier curve

### 3.3 高阶 Bézier 曲线对比

以  $n=6$  为例,验证本文 getBezierArr 方法对高阶 Bézier 曲线生成的有效性。由式(1)可得出六阶

Bézier 曲线计算公式如下:

$$P^6(t) = \sum_{i=0}^6 P_i B_{i,6}(t), t \in [0,1]. \quad (16)$$

以式(16)为理论基础,可实现经典的六阶 Bézier 曲线拟合算法,绘制函数为 CreateSixBézier。设置相同的顶点数和 Bézier 曲线控制点集合,与本文 getBezierArr 方法进行对比,六阶 Bézier 曲线试验参数见表 3。通过表 3 的点集绘制生成六阶 Bézier 曲线,本文 getBezierArr 方法用绿色线条,CreateSixBézier 方法用一系列连续的点,六阶 Bézier 曲线轨迹效果对比如图 4 所示。

表 3 六阶 Bézier 曲线试验参数

Table 3 Experimental parameters of sixth-order Bézier curve

参数	getBezierArr	CreateSixBézier
顶点数/个	100	100
起点坐标/mm	(80,0,0)	(80,0,0)
终点坐标/mm	(0,0,0)	(0,0,0)
控制点 1 坐标/mm	(50,0,-50)	(50,0,-50)
控制点 2 坐标/mm	(75,40,0)	(75,40,0)
控制点 3 坐标/mm	(50,80,50)	(50,80,50)
控制点 4 坐标/mm	(0,80,0)	(0,80,0)
控制点 5 坐标/mm	(-25,40,-50)	(-25,40,-50)

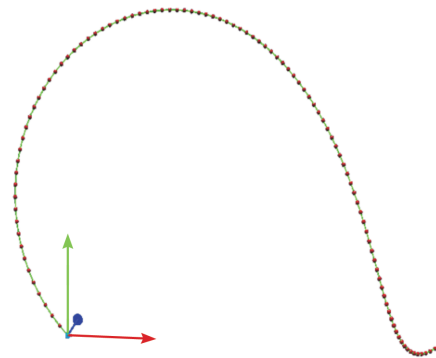


图 4 六阶 Bézier 曲线轨迹效果对比

Fig. 4 Comparison of trajectory effect of sixth-order Bézier curve

### 3.4 试验对比结果

在起点、终点、中间控制点一致的情况下,Babylon.js 环境下本研究 Bézier 曲线形成方法与经典二、三阶和六阶 Bézier 曲线算法所形成的运动轨迹完全重合,验证了本研究  $n$  阶 Bézier 曲线形成方法在低阶和高阶上的有效性,从而为下一阶段小球沿  $n$  阶 Bézier 曲线轨迹运动仿真试验与曲线形状可视化交互设计打下基础。

## 4 仿真试验

### 4.1 仿真试验设计

在 Babylon.js 环境中设计 transformNode 方法使小球沿所形成的 Bézier 高阶曲线路径运动,transformNode 方法能改变物体的三维坐标,如果此方法刷新较快的话,就会在视觉上形成物体移动的效果。由式(1)~(3)可知,Bézier 阶数越高,则曲线上各点的可导阶数更高,曲线连续性更强,曲线更光滑,物体的运动视觉效果肯定更好,但要调整的点比较多;因此,可针对不同情况采取合适的阶次,在需要光滑度较高的地方可采用高阶次 Bézier 曲线设计,精度要求不高的地方可采取低阶 Bézier 曲线设计。为实现不同高阶的小球模拟和可视化操作效果,在此通过 getBezierArr 方法生成一个由两段高阶 Bézier 曲线组成的小球轨迹图,第一段为五阶,第二段为六阶,然后让小球沿此路径移动,观察小球是否能沿此路线运动。首先设置每段的起点和终点坐标,为实现连续运动,让第一段的终点和第二段的起点坐标相同,第一段设计 4 个中间控制点,第二段设计 5 个中间控制点的位置坐标,小球 Bézier 曲线试验参数见表 4。首先通过本文方法拟合出一条三维弯曲轨道;然后在 Babylon.js 的 playground 中运用 transformNode 组件方法使小球沿所形成的 Bézier 高阶曲线路径运动;最后设计曲线轨迹的调整控制方法,当鼠标光标位于某个控制小球上方时,就会触发该控制点的三维坐标轴,长按鼠标左键选取某个轴就可对该轴方向的控制点坐标进行调整,松开鼠标左键时根据新的控制点坐标计算出新的 Bézier 曲线轨迹点,绘制出新的三维轨迹。

表 4 小球 Bézier 曲线试验参数

Table 4 Experimental parameters of Bézier curve of the ball

参数	第一段(五阶)	第二段(六阶)
顶点数/个	100	100
起点坐标/mm	(-50,0,0)	(0,0,0)
终点坐标/mm	(0,0,0)	(50,0,0)
控制点 1 坐标/mm	(-40,15,-10)	(10,-15,5)
控制点 2 坐标/mm	(-30,30,-15)	(20,-30,10)
控制点 3 坐标/mm	(-20,50,-20)	(25,-50,20)
控制点 4 坐标/mm	(-10,30,-10)	(30,-30,10)
控制点 5 坐标/mm		(40,-15,5)

通过表 4 中的第一段控制点  $C_1$ 、 $C_2$ 、 $C_3$ 、 $C_4$  和第二段控制点  $B_1$ 、 $B_2$ 、 $B_3$ 、 $B_4$ 、 $B_5$  成功绘制小球的高阶 Bézier 曲线运动轨迹,如图 5 所示,也通过 transformNode 方法成功实现了两段小球的平滑曲线运动,如图中  $T_0$ 、 $T_1$ 、 $T_2$ 、 $T_3$ 、 $T_4$ 、 $T_5$  处的小球轨迹点所示。

在图 5 的基础上,使用鼠标对图 5 中第一段控制点  $C_1$ 、 $C_2$ 、 $C_3$ 、 $C_4$  和第二段控制点  $B_1$ 、 $B_2$ 、 $B_3$ 、 $B_4$ 、 $B_5$  的坐标值进行拖拽调整,每拖拽一次就实时改变曲线的形状,实现可视化交互操作,最终实现沿目标轨迹曲线运动的效果。Babylon.js 三维场景下,绿色线条为五阶 Bézier 曲线,红色线条为六阶 Bézier 曲线,点状线为控制点位置调整前所生成曲线效果,实线为调整后曲线效果,控制点位置调整前后轨迹效果对比如图 6 所示。将调整前后插值点数据导入 MatLab 中,生成三维坐标效果图,轨迹调整前后对比如图 7 所示。

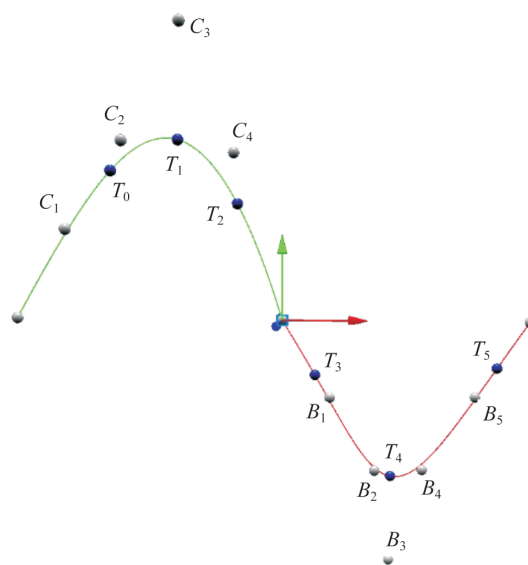


图 5 小球的高阶 Bézier 曲线运动轨迹

Fig. 5 High-order Bézier curve trajectory of the ball

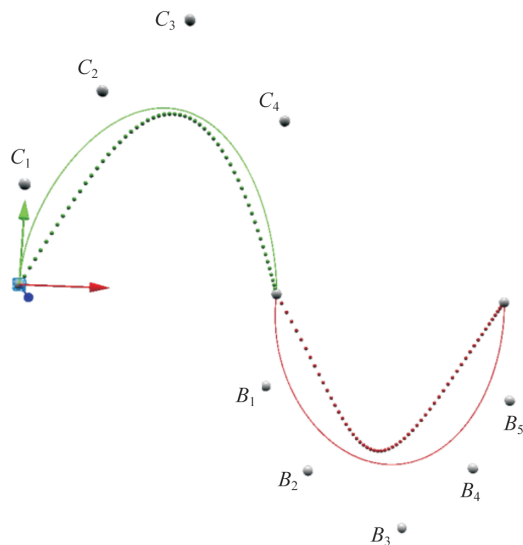


图 6 控制点位置调整前后轨迹效果对比

Fig. 6 Comparison of trajectory effect before and after adjustment of control point positions

## 4.2 仿真结果

图 5 中,小球可以围绕生成的两段高阶 Bézier 曲线轨迹运动,证明了本文方法运用于三维物体轨迹生成的有效性。从图 6 和图 7 中的三维曲线绘制结果可知,一些复杂曲线轨迹可通过本文方法快速实现,只需改动控制点位置就可快速调整并绘制出整条轨迹,且对于不同阶次的 Bézier 曲线都能有效进行调整,从而实现了轨迹形状的可视化交互操作。

## 5 结 语

本研究提出一种基于 Babylon.js 的任意阶 Bézier 曲线生成方法来设计网页端三维物体的运动轨迹。在 Babylon.js 环境下进行小球两段高阶 Bézier 曲线运动仿真,证明本文方法能很好地生成复杂的三维物体运动轨迹,通过鼠标可对不同阶次 Bézier 曲线轨迹形状进行可视化交互操作,简化了开发人员的工作量,提高了工作效率,也为应用 Bézier 曲线去规划基于网页端的三维物体运动轨迹提供了参考。

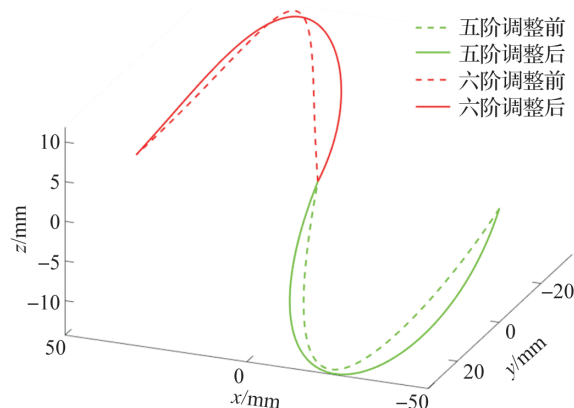


图 7 三维空间中轨迹调整前后对比

Fig. 7 Comparison before and after trajectory adjustment in 3D space

## 参考文献:

- [1] 王磊,高珏,金野,等. 基于 Web3D 无插件的三维模型展示的研究[J]. 计算机技术与发展,2015,25(4):217.
- [2] 李琨,贾立伟,石晓明. 基于 WebGL 技术的医学虚拟实验室的设计与实现[J]. 电脑知识与技术,2017,13(23):202.
- [3] 马亮,王彬,龚强,等. WebGL 3D 技术及其在高校 B/S 系统中的应用研究[J]. 电子测试,2021(17):69.
- [4] 何珊,卿鄰波,何小海,等. 基于 WebGL 的岩心图像的三维重建及应用[J]. 现代计算机,2017(20):73.
- [5] MOREAU-MATHIS J. Babylon.js Essentials[M]. Birmingham: Packt Publishing Ltd,2016:127.
- [6] SHEPILIEV D S, MODLO Y O, YECHKALO Y V, et al. WebAR development tools: an overview[J]. CEUR Workshop Proceedings,2020,2832:84.
- [7] HAN L, YASHIRO H, NEJAD H T N, et al. Bezier curve based path planning for autonomous vehicle in urban environment[C]//2010 IEEE intelligent vehicles symposium. La Jolla: IEEE,2010:1036.
- [8] 杨彬,宋学伟,张勇,等. 模拟驾驶人行为的无人驾驶车辆避障轨迹规划[C]//2021 中国汽车工程学会年会暨展览会. 上海:中国汽车工程学会,2021:170.
- [9] THARWAT A, ELHOSENY M, HASSANIEN A E, et al. Intelligent Bézier curve-based path planning model using Chaotic Particle Swarm Optimization algorithm[J]. Cluster Computing,2019,22(2):4745.
- [10] SONG B, WANG Z D, ZOU L. An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve[J]. Applied Soft Computing,2021,100(1):106960.
- [11] 孙晨阳. 小麦生长的三维可视化建模方法研究[D]. 郑州:郑州大学,2019.
- [12] WU X L, HUANG X N. Using Bézier curve to bend 3D objects in unity[J]. Modern Computer,2016,7(15):1.
- [13] 施法中. 计算机辅助几何设计与非均匀有理 B 样条:CAGD & NURBS[M]. 北京:北京航空航天大学出版社,1994:121.
- [14] 张颖,练章华. 基于 Bezier 曲线的特殊螺旋密封面参数化设计[J]. 中国安全生产科学技术,2019,15(11):24.
- [15] 孔令德,康凤娥. 基于双三次 Bezier 曲面在球体建模中的应用[J]. 计算机应用与软件,2017,34(5):86.

~~~~~  
(上接第 6 页)

- [13] MANGASULI S N, HOSAMANI K M, DEVARAJEGOWDA H C, et al. Synthesis of coumarin-theophylline hybrids as a new class of anti-tubercular and anti-microbial agents[J]. European Journal of Medicinal Chemistry, 2018,146(25):747.
- [14] 陈清勉,罗俏,陈建华,等. 新型香豆素类  $Hg^{2+}$  荧光探针的合成及性能研究[J]. 化学试剂,2020,42(2):199.
- [15] PATAGAR D, UTTARKAR A, PATRA S M, et al. Spiro benzodiazepine substituted fluorocoumarins as potent anti-anxiety agents[J]. Russian Journal of Bioorganic Chemistry,2021,47(2):390.
- [16] WANG X W, LI P, XIAO H, et al. A simple method for synthesis of 5- $CF_3$  substituted dienamides via rearrangement of 2H-pyran derivatives[J]. Tetrahedron,2011,67(39):7618.
- [17] 畅志鑫,袁征,韩小瑜. 膦催化的吡咯烷-螺吡啶酮类化合物的合成[J]. 浙江科技学院学报,2019,31(3):213.
- [18] KUMAR S, SHARMA S P, VASHISHT N, et al. One-pot green synthesis of 2-oxo-2H-chromene-3-carbonitriles using dual-frequency ultrasonication[J]. Russian Journal of Organic Chemistry,2021,57(9):1508.