

# 一种基于半透明颜色叠加与深度值的碰撞检测算法

李 普 孙长乐 熊 伟 王海涛

(大连海事大学船舶机电装备研究所 辽宁 大连 116026)

**摘 要** 为了快速验证虚拟装配仿真过程中零部件的可装配性,改进了一种基于图像空间的快速碰撞检测算法。该算法首先通过半透明颜色叠加实现对无碰撞装配体的快速滤除,并确定出潜在的碰撞区域;然后利用像素深度值求出遮挡对象与待装配物体在其运动方向的最小分离距离,弥补了基于图像空间的碰撞检测算法只能判断物体间是否发生碰撞而不能求距的缺点;最后针对求距环节提出了一种像素区域划分策略,以提高算法的检测精度。测试结果表明,所提算法在整体上能够满足虚拟装配系统实时性和精确性的要求。

**关键词** 虚拟装配,碰撞检测,半透明颜色叠加,深度值,像素区域划分

**中图分类号** TP391.41 **文献标识码** A

## Collision Detection Algorithm Based on Semi-transparent Color Overlay and Depth Value

LI Pu SUN Chang-le XIONG Wei WANG Hai-tao

(Institute of Ship and Electromechanical Equipment, Dalian Maritime University, Dalian, Liaoning 116026, China)

**Abstract** A fast collision detection algorithm based on image-space was proposed with the view of verifying the assemblability of the assembling parts in the process of virtual assembly. Firstly, it filters the non-collision parts through the way of overlaying translucent colors, and identifies the potential collision areas. Then it calculates the minimum separation distance between the covered objects and the assembly objects in the direction of its movement, which can make up for the disadvantages of collision algorithm based on image-space, which just can judge whether a collision occurred but can't calculate the distance. Finally, this paper put forward a strategy of partitioning pixel region which focuses on the procedure of getting the distance, in order to improve the detection precision of the algorithm. Test results show that the algorithm can satisfy the requirements of real-time and accuracy of virtual assembly system on the whole.

**Keywords** Virtual assembly, Collision detection, Semi-translucent color overlay, Depth value, Partitioning pixel region

大型盾构机、卫星等单件小批量生产的复杂机械产品,其本身结构复杂、零部件种类繁多,容易导致装配过程繁琐、难度大的问题;另一方面,产品的设计有误以及产品在装配过程中采用不合理的装配工艺、夹具等,会导致产品无法装配的问题,从而增加了产品的生产成本,延长了产品的生产制造周期。虚拟现实(Virtual Reality, VR)技术的快速发展为研究人员探索解决上述问题提供了新的途径,而被迅速应用于机械制造业中,虚拟装配(Virtual Assembly, VA)技术也由此产生。虚拟装配的实质是产品在设计之后、生产之前进行的装配仿真过程,也是产品数字化的过程。碰撞检测是虚拟装配的关键技术之一,也是进行虚拟装配的前提,在机器人路径规划、3D 游戏、物理仿真等研究领域都有重要的应用,国内外的专家学者对其进行了大量的研究。

目前现有的碰撞检测算法分为基于几何空间和图像空间两种主流算法。其中基于几何空间的碰撞检测算法是采用构造不同的数据结构来判断物体之间是否发生干涉,存在检测效率高、精度低的特点,适用于初检阶段来快速剔除明显不相交的物体。Chang 等<sup>[1]</sup>提出组合使用包围球与轴向包围盒

OBB 的方法提高了碰撞检测的效率,但算法的检测精度较低;潘海鸿等<sup>[2]</sup>针对多机器人碰撞检测耗时过多的问题,提出了并行动态包围体层次树的碰撞检测算法,明显提高了算法的执行效率。基于图像空间的碰撞检测算法主要是利用图形处理器(Graphic Processing Unit, GPU)的加速绘制与计算来提高算法的效率,首先将物体模型通过图形处理器投影绘制到特定的图像平面上,降维得到二维的图像空间,然后通过分析该空间中保存的各种缓存信息,来判断物体之间是否发生碰撞<sup>[3]</sup>,该类算法具有对场景复杂性不敏感、检测精度受到图像分辨率约束的特点。Baciu 等<sup>[4-5]</sup>提出 RECODE 算法通过组合使用深度缓存和模板缓存来判断两凸体之间是否发生碰撞,但算法不能有效地处理非凸复杂物体间的碰撞检测;范昭炜等<sup>[6]</sup>提出了一种基于图像的快速碰撞检测算法,该算法综合了表面凸分解、层次二叉树以及 RECODE 算法在处理碰撞检测问题时所具有的优势,有效提高了碰撞检测的效率,而且能够处理任意形状的多面体;于海军等<sup>[7]</sup>提出的基于图像空间的碰撞检测算法,首先通过构造凸块的 OBB 找到潜在的碰撞检测集,然后在 RECODE 算法的基础上确定出相互干涉

本文受国家自然科学基金(51305052),中央高校基本科研业务费专项资金(3132017027)资助。

李 普(1991—),男,硕士生,主要研究方向为虚拟装配, E-mail: dmulp2015@163.com; 孙长乐(1978—),男,博士,副教授,硕士生导师,主要研究方向为 CAD 模型重用、虚拟装配、协同设计技术, E-mail: xxcl1999@126.com(通信作者); 熊 伟(1972—),男,博士,教授,博士生导师,主要研究方向为机械电子、虚拟装配; 王海涛(1973—),男,博士,教授,博士生导师,主要研究方向为气动技术、虚拟装配。

的物体,提高了碰撞检测的效率;Govindaraju 等<sup>[8]</sup>提出的 CULLIDE 算法通过利用 GPU 的可见性查询功能快速地裁剪出潜在的碰撞集,最后在 CPU 上针对三角面片进行精确的碰撞检测,算法不涉及帧缓存的读取,但检测精度受图像分辨率的影响,而且不提供距离计算和刺穿信息等;Kim 等<sup>[9]</sup>基于多核 CPU、GPU 与层次包围体提出 HPCCD 算法,适用于柔性体之间的连续碰撞检测,但该算法不能返回精确的碰撞点;杜鹏等<sup>[10]</sup>提出了一种基于数据分块思想的并行碰撞检测算法,提高了复杂场景中碰撞检测的效率;邹益胜等<sup>[11]</sup>根据光线与封闭物体间的相交特性提出 Image\_CD 算法,虽然可以直接处理非凸体,但算法的精度受图像分辨率的影响较大;杜鹏等<sup>[12]</sup>提出了组合利用 GPU、碰撞流和三角形相交检测的方法,提高了虚拟拆卸过程中零部件之间碰撞检测的效率;Du 等<sup>[13]</sup>提出了一种基于高性能 GPU 集群的并行连续性碰撞检测算法,采用动态分配的方法来平衡系统负载,但存在无法处理对象自我碰撞检测的缺点。

由于以上碰撞检测算法应用领域的差异,存在无法同时保证碰撞检测效率快和精度高的缺点,大多数基于图像空间的碰撞检测算法是作为初检在 GPU 上计算出潜在的相交集,最后在 CPU 上针对三角面片进行精确的检测,而虚拟装配中的碰撞检测是对仿真过程中装配顺序和装配轨迹的合理性、可装配性进行验证,以判断产品在虚拟环境中装配时是否发生干涉,只有当碰撞检测的速度高于人机交互时的速度且检测精度大于产品的设计精度时,才能真实地发现产品在装配过程中存在的问题,因此虚拟装配技术对碰撞检测的实时性和精确性有着严格的要求。近年来,随着计算机图形硬件 GPU 以超过摩尔定律的速度迅猛发展,极大地提高了计算机图形处理的速度和质量,基于 GPU 的碰撞检测算法迅速成为研究的热点,该类算法是通过分担 CPU 的计算负载及对图像的快速处理来提升整体算法的运行效率,因此考虑在利用 GPU 自身优势的前提下,针对虚拟装配系统(Virtual Assembly System, VAS)中零部件之间的碰撞检测问题,改进了一种基于图像空间的碰撞检测算法。

### 1 碰撞检测的研究目的与研究思路

图 1 为 VAS 平台的仿真界面示意图,主要包括仿真主操作区、菜单功能区、装配树生成区、状态栏、正向视点观察平面区和垂直视点区 6 部分,每部分又细分为多个子模块,通过各个子模块的具体操作实现对机械产品模型的装配仿真。



图 1 VAS 平台仿真界面示意图

图 2 为在 VAS 平台上针对产品装配的操作流程图,首先将产品的零部件模型导入到 VAS 平台中,导入后的零部件模型为静态的对象(相互分离),用鼠标选取将要装配的零部件作为可移动的待装配物体,剩余零部件则为静态的环境对象

(包括已装配和未装配的零部件),然后针对待装配物体设置运动轴(或自定义约束轴),削减自由度,目的是使 VAS 平台的虚拟装配环境是在计算机屏幕上的二维显示,而虚拟装配环境中的三维物体本身具有 6 个自由度,设置运动轴后的物体只具有 2 个自由度,即沿轴向的移动和转动,因此恰好匹配于鼠标的 2 个自由度,即可通过鼠标来进行模型位姿的调整、拖拽装配等人机交互操作。

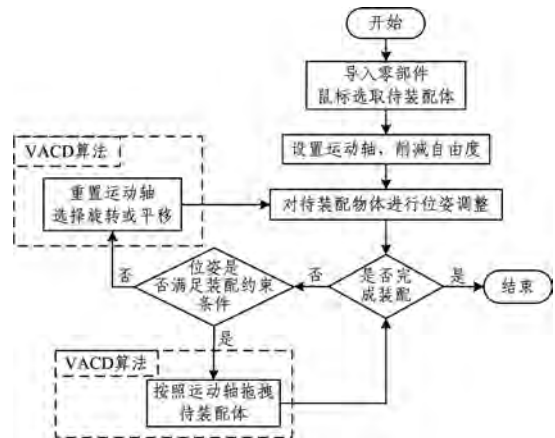


图 2 在 VAS 平台上针对产品装配的操作流程图

虚拟装配系统 VAS 中的碰撞检测问题可以描述如下:首先对待装配物体进行位姿调整,人为判断是否完成装配,若未完成,则进一步判断当前位姿是否满足装配约束条件,若满足,则沿设置的运动轴进行拖拽装配;若不满足装配约束条件,则重置运动轴,再次对待装配物体进行位姿调整,直至完成对当前零部件的装配为止。因此,VAS 平台中的碰撞通常发生在当前进行装配的零部件与静态的环境对象之间,VACD(Virtual Assembly Collision Detection)算法的提出能够有效避免待装配的零部件在调整位姿和拖拽装配的过程中与其他零部件发生碰撞,图 3 为 VACD 算法的基本原理流程图。

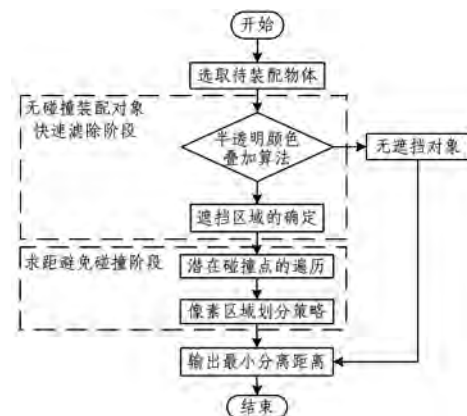


图 3 VACD 算法的基本原理流程图

VACD 算法主要包括无碰撞装配对象快速滤除和求距避免碰撞两个阶段,第一阶段包括半透明颜色叠加算法和遮挡区域的确定两部分,首先根据待装配物体设置的运动轴,判断在其运动方向上是否存在遮挡对象,若不存在遮挡对象,则待装配物体可以在运动方向上运动任意长度;若存在遮挡对象,则执行求距避免碰撞阶段,利用遮挡区域结合像素区域划分策略快速地遍历出潜在碰撞点,求出待装配物体与遮挡对象在运动方向上的最小分离距离。

## 2 无碰撞装配对象快速滤除阶段

### 2.1 半透明颜色叠加算法

针对计算机二维屏幕上的两个有界面 A 和 B,填充区域分别设置单一颜色为红色、绿色,如图 4 所示,难以判断出两者边界区域是什么图形,现对可能存在的两种情况进行举例。

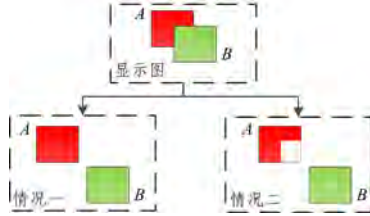


图 4 有界面无半透明颜色叠加时可能存在的位置关系图

情况 1 为两个有界矩形区域 A 和 B,显示图是由于面 B 部分区域位于面 A 的前部;情况 2 为有界的六边形区域 A,有界的矩形区域 B,显示图是由于面 B 的左上角恰好补缺面 A 的右下角。半透明颜色叠加算法的出现,有效地解决了上述问题,即半透明颜色叠加算法能够有效地区分以上列举出的两种情况,如图 5 为情况 1 在半透明颜色叠加算法下的效果图。



图 5 情况 1 在半透明颜色叠加算法下的效果图

半透明颜色叠加算法是指将半透明的图形叠加到不透明的图形上,从而实现两种不同颜色的混合叠加。两颜色混合叠加需要的信息有:不透明颜色的背景像素 RGB 值、叠加颜色的半透明像素 RGB 值及其半透明像素的半透明值  $\alpha, 0 \leq \alpha \leq 1$ ,其中 0 代表完全透明,0~1 代表半透明,1 代表不透明。

如图 6 为二维平面上半透明颜色的叠加原理,把矩形  $m$  (叠加颜色)叠加到圆形  $n$  (背景颜色)上,已知圆形  $n$  区域和矩形  $m$  区域像素的 RGB 值分别为  $(R_n, G_n, B_n)^T$  和  $(R_m, G_m, B_m)^T$ ,且矩形  $m$  的半透明值为  $\alpha_0$ ,最后由颜色叠加公式:

$$\begin{pmatrix} R_{mn} \\ G_{mn} \\ B_{mn} \end{pmatrix} = \begin{pmatrix} R_n \\ G_n \\ B_n \end{pmatrix} \times (1 - \alpha_0) + \begin{pmatrix} R_m \\ G_m \\ B_m \end{pmatrix} \times \alpha_0 \quad (1)$$

便可求得叠加后的  $mn$  区域每个像素的新颜色 RGB 值。以上为二维平面上半透明颜色的叠加原理,该半透明颜色叠加算法同样适用于三维空间的实体模型中。

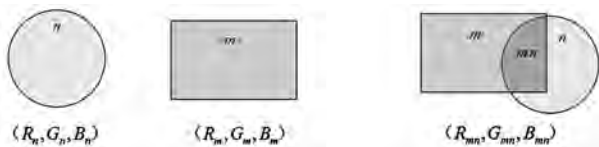


图 6 二维平面上半透明颜色的叠加原理

### 2.2 遮挡区域的确定

该阶段主要采用半透明颜色叠加算法来判断待装配物体在其运动方向上是否存在静态对象的遮挡,图 7 为遮挡区域的确定阶段流程图。

用鼠标选取待装配物体 A,并将其设置为半透明状态(半透明值为  $\alpha_0$ ),装配方向为运动轴方向,取垂直于装配方向的

法平面对应于 VAS 平台仿真界面的正向视点观察平面区,其中待装配物体 A 在正向视点观察平面上的正投影区域为待检测区域。

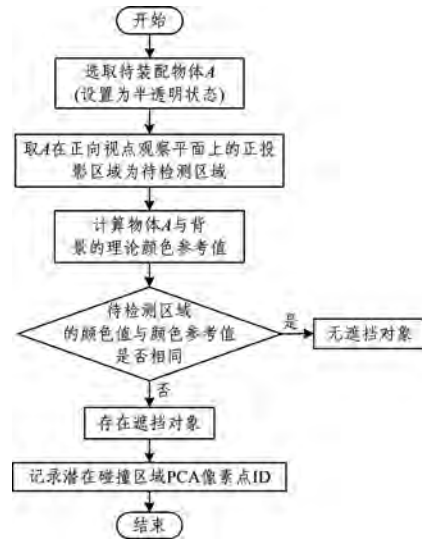


图 7 遮挡区域的确定阶段流程图

首先记录待检测区域在正向视点观察平面上的所有像素点 ID,然后假设当前待装配物体运动方向上不存在遮挡对象,已知装配环境的背景颜色值为  $(R_{背}, G_{背}, B_{背})^T$ ,则待检测区域像素颜色值  $(R_0, G_0, B_0)^T$  为当前半透明待装配物体的颜色值  $(R_{待}, G_{待}, B_{待})^T$  与背景颜色值  $(R_{背}, G_{背}, B_{背})^T$  的叠加,即:

$$\begin{pmatrix} R_0 \\ G_0 \\ B_0 \end{pmatrix} = \begin{pmatrix} R_{背} \\ G_{背} \\ B_{背} \end{pmatrix} \times (1 - \alpha_0) + \begin{pmatrix} R_{待} \\ G_{待} \\ B_{待} \end{pmatrix} \times \alpha_0 \quad (2)$$

然后把叠加后的颜色值  $(R_0, G_0, B_0)^T$  称为正向视点观察平面上待检测区域的理论颜色参考值,因为该理论颜色参考值是在待装配物体运动方向上无遮挡对象存在时计算得到的。图 8 为待装配物体颜色与背景颜色的叠加示意图。



图 8 待装配物体颜色与背景颜色的叠加示意图

在遮挡对象确定前,待检测区域颜色值存在两种情况:1)待检测区域上的颜色值与理论颜色参考值相同,说明当前装配物体的运动方向上无遮挡对象存在;2)待检测区域上全部或局部的颜色值与理论颜色参考值不同,说明当前装配物体的运动方向上至少存在一个遮挡对象,如图 9 所示。

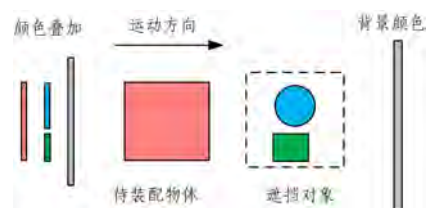


图 9 待装配物体、遮挡对象以及背景颜色的叠加示意图

因此,通过查询待检测区域是否存在不同于理论颜色参

考值的像素点,便可以判断出当前装配物体的运动方向上是否存在遮挡对象,最后把不参与半透明颜色叠加的其他无碰撞装配对象滤除,从而实现遮挡对象的确定。最后记录待检测区域上不同于理论颜色参考值的像素点  $ID$ ,这些像素点所组成的区域为遮挡区域,遮挡区域便为待装配物体  $A$  与其运动方向上遮挡对象潜在的碰撞区域(Potential Collision Area, PCA)。

### 3 求距避免碰撞阶段

当待装配物体  $A$  的运动方向上不存在遮挡物体时,把无穷大的可移动距离反馈给当前待装配物体,即此时待装配物体可以运动任意长度的距离;当待装配物体  $A$  的运动方向上存在遮挡物体时,由于待装配物体与遮挡对象处于相互分离的状态,针对潜在碰撞区域  $PCA$  进一步求取它们在运动方向上的最小分离距离。图 10 给出了存在不同遮挡对象时,遮挡对象在垂直视点方向的两种分布情况,其中  $A$  为可移动的待装配物体,其余物体为静态的环境对象。

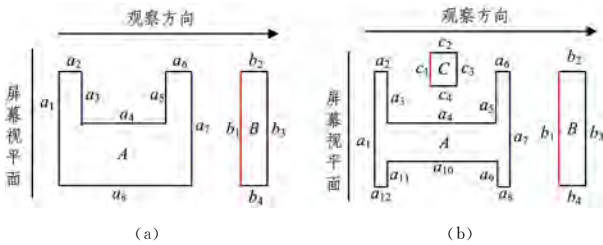


图 10 遮挡对象在垂直视点方向的分布图

求距避免碰撞阶段是针对物体的几何面层次进行的,即场景中的物体是由自身的几何面集组成,设待装配物体几何面集  $A = \{a_1, a_2, \dots, a_{m-1}, a_m\}$ , 遮挡对象的几何面集  $B = \{b_1, b_2, \dots, b_{n_1-1}, b_{n_1}\}$ ,  $C = \{c_1, c_2, \dots, c_{n_2-1}, c_{n_2}\}, \dots$ , 规定由物体内部指向外部的方向为每个几何面的正法向量,图 11 为物体几何面的正法向量示意图。

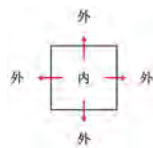


图 11 物体几何面的正法向量示意图

首先针对待装配物体与遮挡对象集分别进行几何面层次的拓扑分解,其中针对待装配物体几何面的拓扑分解为:求解各个几何面正法向量与待装配物体运动方向向量点乘值,保留点乘值为正(两向量夹角为锐角)的几何面,删减剔除其余的几何面,即保留拓扑分解后的待装配物体几何面为集合  $M(a_m \dots)$ ,如图 10(a)中的  $a_3, a_7$  以及图 10(b)中的  $a_3, a_7, a_{11}$ ;同理,求解各个几何面正法向量与待装配物体运动方向向量点乘值,保留点乘值为负(两向量夹角为钝角)的几何面,删减剔除其余的几何面,即保留拓扑分解后的遮挡对象几何面为集合  $N(b_n, c_n, \dots)$ ,如图 10(a)中的  $b_1$  以及图 10(b)中的  $b_1, c_1$ 。

#### 3.1 潜在碰撞点的遍历

算法取待装配物体  $A$  的正向视点观察平面为局部的屏幕视平面,待装配物体的运动方向与屏幕像素的  $Z$  轴深度方向共线,像素的深度值是指沿着观察系统的  $Z$  轴来计算各物

体距离观察平面的深度,主要用来进行可见性的判断,实现对不透明物体的消隐,深度缓存算法只能显示距离视点最近的不透明物体<sup>[14]</sup>。图 12 为潜在碰撞点的遍历流程图。

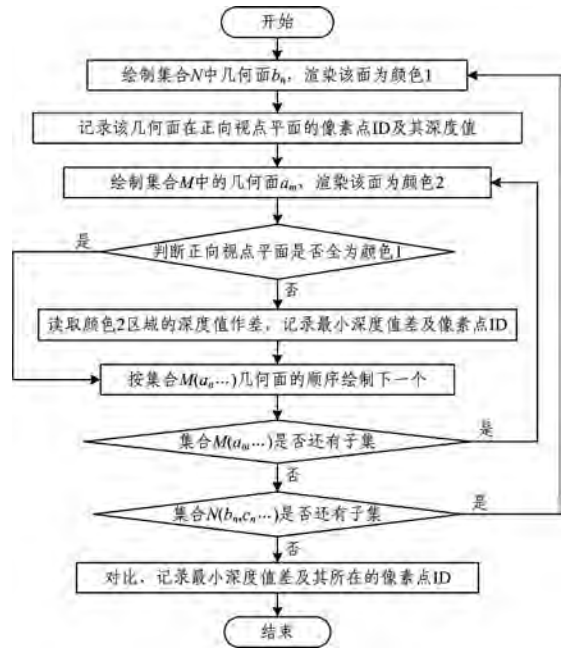


图 12 潜在碰撞点的遍历流程图

遍历潜在碰撞点之前进行了基于物体几何面层次的拓扑分解,已知遮挡对象几何面集合  $N(b_n, c_n \dots)$ 和待装配物体几何面集合  $M(a_m \dots)$ ,首先对遮挡对象几何面集合  $N$  中的首个几何面  $b_n$  进行绘制,渲染该面为颜色 1,并记录在正向视点平面的像素点位置及其深度值,然后对待装配物体的几何面集合  $M$  中的首个几何面  $a_m$  进行绘制,渲染该面为颜色 2,判断当前正向视点平面是否全为颜色 1,若是,则证明几何面  $a_m$  全部位于几何面  $b_n$  的后面,不可见(即几何面  $b_n$  的深度值小于几何面  $a_m$  的深度值),把当前几何面  $a_m$  从集合  $M$  中删减剔除,然后按照集合  $M(a_m \dots)$  几何面的存储顺序绘制下一个几何面。若不全为颜色 1,则说明几何面  $a_m$  局部位于几何面  $b_n$  的前面,读取颜色 2 区域的深度值,将对应像素点深度值作差,记录本次求距最小深度值差及其像素点  $ID$ ,把当前几何面  $a_m$  从集合  $M$  中删减剔除。依次将几何面  $b_n$  与待装配物体集合  $M(a_m \dots)$  中的几何面进行求距处理,然后对比、记录最小深度值差及其所在的像素点  $ID$ 。

循环  $M * N = M(a_m \dots) * N(b_n, c_n \dots)$  次执行潜在碰撞点的遍历,最终在循环过程中求得的最小深度值差组成了深度差集合,通过比较最终确定出最小的深度值差及其所在的像素点  $ID$ 。

#### 3.2 像素区域划分策略

由于在基于图像空间的碰撞检测算法中,图像本身是空间离散采样,其精度普遍受到图像分辨率的影响,通过调整观察坐标系中的原点模拟视点(摄像机的位置)到投影平面的距离,即视距,可以使得屏幕视平面上的图像按照对应的比例来实现缩放图像面积的效果。

像素区域划分是在遍历像素深度值差和缩放图像之间寻找一种时间优化策略,从而提高算法的整体效率。在遍历出潜在的碰撞区域后,执行潜在碰撞点的遍历环节,从而求出最小深度值差及其像素点  $ID$ ,但当遮挡区域在屏幕视平面上所占用的像素点数量达到较高的数量级时,执行潜在碰撞点遍历

环节的时间也将延长。在遍历像素深度值差之前,通过增大观察视距(视距变换),把遮挡区域所占用的像素点数量减少至底数量级,遍历视距变换后的遮挡区域像素点,像素点数量的减少意味着图像精度的降低。精度定义为最大深度值差和最小深度值差的差值,对比是否满足精度要求,可为检测精度设定阈值,若精度低于阈值,则记录最小深度值差及其像素点位置;若精度高于阈值,则需要针对遍历出的最小深度值差所在的像素点进行像素区域划分,如图 13 为像素区域划分示意图。

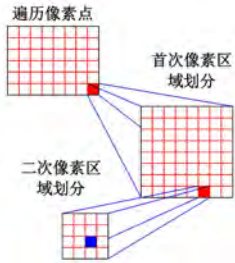


图 13 像素区域划分示意图

首先减小视距,进行首次像素区域划分,将该像素所在区域的边长增长至原来的  $x(x$  大于或等于 2) 倍,则像素点所占的面积增大为原像素点面积的  $x^2$  倍,并将其均匀划分为  $x^2$  部分,对应视平面的  $x^2$  个像素点,进一步遍历出最小深度值差所在的像素点,进行像素区域划分策略的迭代求距,直至满足精度要求,并把此时求得的最小深度值差与模型实际尺寸相对应,即为求得的最小分离距离,该最小分离距离是当前待装配物体避免与遮挡对象发生碰撞的最大可移动距离,最后将该距离反馈给当前待装配物体。

### 4 算法测试结果及分析

测试硬件条件:Inter(R) Core(TM)i3-4370 CPU @3.80 GHz,4GB 内存,AMD Radeon R7 350 显卡,Windows 系统。实验中半透明颜色叠加算法的半透明度均取值 0.5,屏幕分辨率为  $1920 \times 1080$ 。测试内容主要包括:像素区域划分策略对碰撞检测效率和精度的影响、在同一装配场景中不同算法的性能比较。

为了检验 VACD 算法在像素区域划分策略上对碰撞检测算法效率、精度的影响效果,分别对 4 种像素区域划分方法进行效率对比,首先采用 OpenGL 应用程序接口测试出变换一次视距的时间数量级为  $3.8 \times 10^{-1}$  ms,而遍历单个像素点的深度值差的时间数量级为  $7 \times 10^{-2}$  ms,设计 4 种像素划分策略: $x=10,15,20,25$  的情况被分别命名为策略 1—策略 4,即策略分别针对单个像素进行  $10^2$  倍、 $15^2$  倍、 $20^2$  倍、 $25^2$  倍划分求距,不同策略下算法效率和检测精度的关系如图 14 所示。

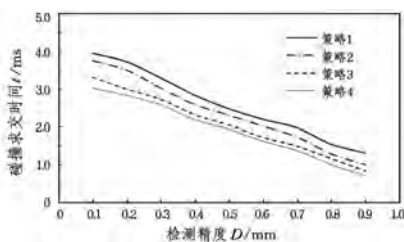


图 14 不同策略下算法效率和检测精度关系图

测试结果表明,算法的效率受到检测精度的影响,由图 14 可知,随着检测精度降低,算法的效率呈上升趋势,即检测

精度越低,算法执行效率越高。而且 4 种不同的像素区域划分策略对算法的效率影响也存在差异,由于像素区域划分策略由视距变换和遍历像素点深度值差两部分组成,视距变换的时间大于遍历单个像素点深度值差的时间,且在时间上相差一个数量级,因此在像素区域划分时,尽量减少变换视距的次数,增加遍历像素点的个数,即在变换视距和遍历像素点深度值差之间寻找平衡点,从而使得算法的效率最优化。

通过 VAS 平台建立好的装配环境,分别采用不同的碰撞检测算法:RECODE 算法、RAPID 算法、FZWCD 算法<sup>[6]</sup>以及本文的 VACD 算法,在检测精度为 1mm 的前提下,记录运行 1000 步的平均时间  $T$ 。另外,由于 VACD 算法本身在遍历潜在碰撞点时受到像素区域划分策略的影响,本次测试选择  $x=25^2$  的划分策略。图 15 给出了 4 种算法在场景中随着场景复杂度(三角面片数量)的增加平均时间  $T$  的变化情况。

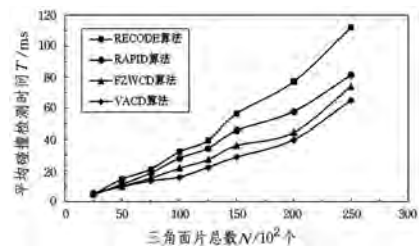


图 15 4 种算法在场景中的测试结果

由图 15 可以看出,本文 VACD 算法在效率上比其他 3 种经典碰撞检测算法有较小幅度的提高,随着场景复杂度的不断增加,VACD 算法所消耗的检测时间也随之增加,两者之间存在线性关系,而非指数增长的关系。本文的 VACD 算法不仅在效率上有所提高,而且还弥补了多数算法只能检测碰撞与否而不能求距的缺点,具有求距特点的 VACD 算法在应用到课题组自主开发的 DMU 虚拟装配系统 VAS 后,能够有效避免零部件仿真时碰撞的发生,更好地满足了虚拟装配的仿真要求。

**结束语** 本文提出了一种新的基于图像空间的快速碰撞检测算法——VACD 算法,通过利用帧缓存中的缓存信息,求出待装配物体在运动方向上与遮挡对象之间的最小分离距离,能够有效地避免模型间装配时碰撞的发生。算法利用图形硬件 GPU 的加速渲染与计算功能,有效地提高了碰撞检测的效率;针对基于图像空间的碰撞检测算法的精度普遍受到分辨率影响的问题,本文提出了一种像素区域划分策略,利用该策略来提高模型间碰撞检测的精度,能够很好地满足虚拟装配系统对实时性和精确性的要求。下一步将在 CPU 和 GPU 之间的负载分配上进行策略优化,充分利用图形处理器强大的并行处理能力,同时实现像素区域划分策略对效率和精度的自适应优化,以进一步提高算法的性能。

### 参考文献

[1] CHANG J W, WANG W, KIM M S. Efficient collision detection using a dual OBB-sphere bounding volume hierarchy[J]. Computer-Aided Design, 2010, 42(1): 50-57.  
 [2] 潘海鸿,戴璇,陈琳,等.多机器人并行动态包围体层次树碰撞检测算法[J].计算机辅助设计与图形学学报,2014,26(11):1948-1956.

表 2 特征识别的准确率比较

车型	(单位:%)		
	CaffeNet+SR	PCA+SIFT	HOG
奥迪_A6L_07	95.2	85.4	87.7
本田_CR-V_07	95.1	87.6	90.9
别克_君越_12	95.2	86.5	87.4

**结束语** 本文基于深度学习自编码器模型和稀疏表示实现了一种图像超分辨率重构算法。该算法将深度卷积神经网络与车型识别问题相结合,提出了基于卷积网络的车型识别框架,并将超分辨率引入车型识别的图像预处理部分,增加了车辆特征的鲁棒性。通过在市内公路监控数据上的对比实验显示,本文提出的车型识别框架和算法取得了良好的效果;但对于无法从外观信息分类的车辆图像,本文提出的方法还不够完善,有待后续研究。另外,车型识别数据库中共有 240 类车型,由于每年车型都有新增,因此需要不断更新车型数据库。

### 参 考 文 献

- [1] BUCH N, VELASTIN S A, ORWELL J. A review of computer vision techniques for the analysis of urban traffic[J]. IEEE Transactions on Intelligent Transportation Systems, 2011, 12(3):920-939.
- [2] DALAL N, TRIGGS B, SCHMID C. Human detection using oriented histograms of flow and appearance[C]// European Conference on Computer Vision. 2006:428-441.
- [3] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. Kluwer Academic Publishers, 2004, 60(2):91-110.
- [4] MA X, GRIMSON W. Edge-based rich representation for vehicle classification[C]// Tenth IEEE International Conference on Computer Vision. 2005:1185-1192.
- [5] OJALA T, PIETIK, INEN M, et al. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns[C]// IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002:971-987.
- [6] ZHANG L, LI S Z, YUAN X, et al. Real-time object classification in video surveillance based on appearance learning[C]// IEEE Conference on Computer Vision & Pattern Recognition. 2007:1-8.
- [7] 孙志军, 薛磊, 许阳明, 等. 深度学习研究综述[J]. 计算机研究应用, 2012, 29(8):2805-2810.
- [8] HINTON G E, SALAKHUTDINOV P R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786):504-507.
- [9] 沈焕锋, 李平湘, 张良培, 等. 图像超分辨率重建技术与方法综述[J]. Optical Technique, 2009, 35(2):196-203.
- [10] YANG J C, WRIGHT J, HUANG T, et al. Image super-resolution as sparse representation of raw image patches[C]// IEEE Conference of Computer Vision and Pattern Recognition (CVPR). 2008:1-8.
- [11] CHO K. Simple Sparsification Improves Sparse Denoising Autoencoders in Denoising Highly Noisy Images[C]// Proceedings of the 30th International Conference on Machine Learning (ICML-13). 2013:432-440.
- [12] ALAIN D, OLIVIER S. Gated Autoencoders with Tied Input Weights[C]// International Conference on Machine Learning. 2013:154-162.
- [13] VINCENT P, LAROCHELLE H, BENGIO Y, et al. Extracting and composing robust features with denoising autoencoders[C]// International Conference on Machine Learning. 2008:1096-1103.
- [14] TRIESCH J. A gradient rule for the plasticity of a neuron's intrinsic excitability[C]// International Conference on Artificial Neural Networks: Biological Inspirations. Springer-Verlag, 2005:65-70.
- [15] YANG J C, WANG Z W, LIN Z, et al. Coupled dictionary training for image super-resolution[C]// IEEE Transactions on Image Processing. 2012:2359-2390.
- [16] JIA Y, SHELHAMER E, DONAHUE J, et al. Caffe: Convolutional Architecture for Fast Feature Embedding[J]. Eprint Arxiv, 2014:675-678.
- (上接第 197 页)
- [3] 水泳. 虚拟现实连续碰撞检测算法研究[D]. 合肥:中国科学技术大学, 2013.
- [4] BACIU G, WONG S K, SUN H. RECODE: an image-based collision detection algorithm[J]. Journal of Visualization & Computer Animation, 1999, 10(4):181-192.
- [5] BACIU G, WONG W S K. Image-based techniques in a hybrid collision detector[J]. IEEE Transactions on Visualization and Computer Graphics, 2003, 9(2):254-271.
- [6] 范昭炜, 万华根, 高曙明. 基于图像的快速碰撞检测算法[J]. 计算机辅助设计与图形学学报, 2002, 14(9):805-810.
- [7] 于海军, 马纯永, 张涛, 等. 基于图像空间的快速碰撞检测算法[J]. 计算机应用, 2013, 33(2):530-533.
- [8] GOVINDARAJU N K, REDON S, LIN M C, et al. CULLIDE: Interactive collision detection between complex models in large environments using graphics hardware[C]// Proceedings of the ACM SIGGRAPH/ EUROGRAPHICS Conference on Graphics Hardware. Eurographics Association, 2003:25-32.
- [9] KIM D, HEO J P, HUH J, et al. HPCCD: Hybrid parallel continuous collision detection using CPUs and GPUs[C]// Computer Graphics Forum. Blackwell Publishing Ltd, 2009, 28(7):1791-1800.
- [10] 杜鹏, 唐敏, 童若锋. 多核加速的并行碰撞检测[J]. 计算机辅助设计与图形学学报, 2011, 23(5):833-838.
- [11] 邹益胜, 丁富国, 周晓莉, 等. 一种基于图像空间的碰撞检测算法[J]. 系统仿真学报, 2011, 23(5):944-949.
- [12] DU P, ZHAO J Y, PAN W B, et al. GPU Accelerated Real-Time Collision Handling in Virtual Disassembly[J]. Journal of Computer Science and Technology, 2015, 30(3):511-518.
- [13] DU P, LIU E S, SUZUMURA T. Parallel continuous collision detection for high-performance GPU cluster[C]// Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. ACM, 2017:4.
- [14] 徐文鹏, 王玉琨, 刘永和. 计算机图形学基础(OpenGL版)[M]. 北京:清华大学出版社, 2014:213-232.