

99 行代码实现《只因你太美》

Taichi Hackathon 2022 猫猫头队 作品

技 术 文 档



作者:

Denver Pilphis

MuGdxy

Weishao

2022 年 12 月 03 日

目 录

99 行代码实现《只因你太美》	1
1 选题背景.....	3
2 技术路线.....	5
2.1 初始试样准备——二维非凸区域颗粒填充算法推导	5
2.2 Taichi DEM 代码整理.....	9
2.3 “时序-分组-体力” (SGF) 列表	10
2.4 调试参数，确保试样正确响应	11
3 项目创新点、特色与展望	13
参考文献.....	14

1 选题背景

2022 年 11 月，Denver Pilphis 和 MuGdxy 在先前 Taichi DEM 代码优化大赛中，以“使用 Taichi DEM 实现工程定量仿真”作品荣获第一名。团队在该次比赛中，通过实战学习了 Taichi 语言的使用。作品实现了离散单元法 (DEM) 中胶结材料的力学响应 (理论详见文献[1])，即物体由若干个胶结起来的颗粒组成，胶结可以受拉、受压和受剪，进行力-位移响应，而当胶结在拉、压、剪方向的应力超过其强度时，胶结将破坏，胶结两端的颗粒脱开。使用胶结模型可以做出很多很有趣的例子，例如图 1 所示的软体兔子，就是作品的其中一个 Demo：

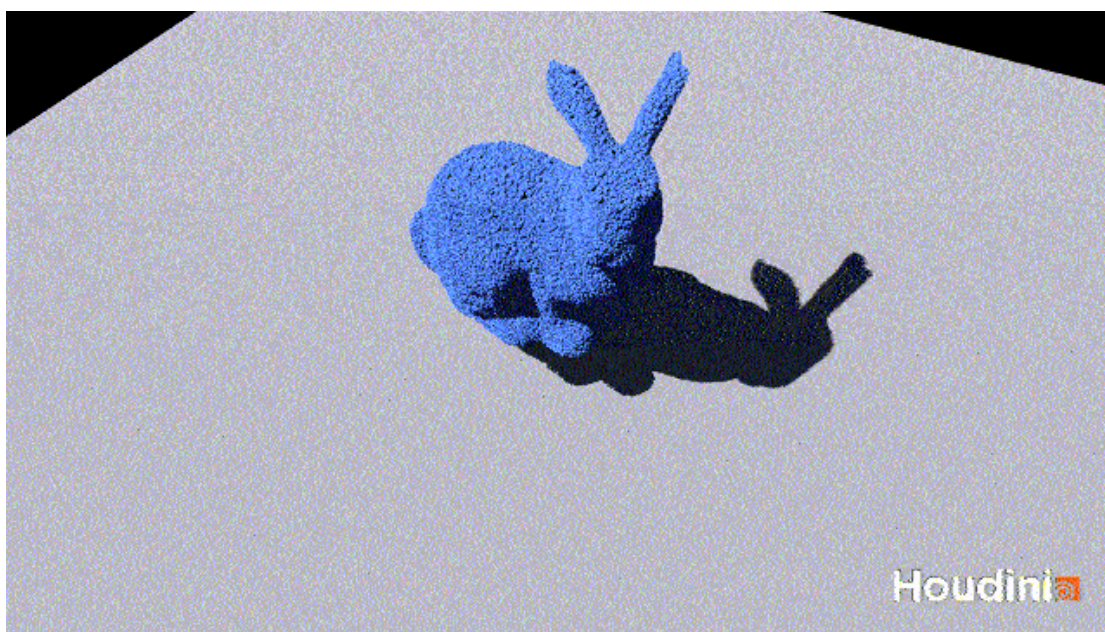


图 1 软体兔子蹦蹦跳

其中兔子是一个类似软体，胶结弹性模量较小而胶结强度较大，兔子的各处可以产生较大的弹性变形，而兔子自身不会破碎。

本次比赛项目组成员观看图 1 时，联想到中国内地歌手蔡徐坤老师曾在 2016 年综艺节目《偶像练习生》第三期中，为其代表作《只因你太美》编舞，并简单秀了 Demo^[2]。在如今的中文互联网上，该 Demo 依旧十分火爆。Demo 中，蔡

老师其中有一个舞步动作 (图 2)，与图 1 中的兔子耳朵的灵动十分相似：

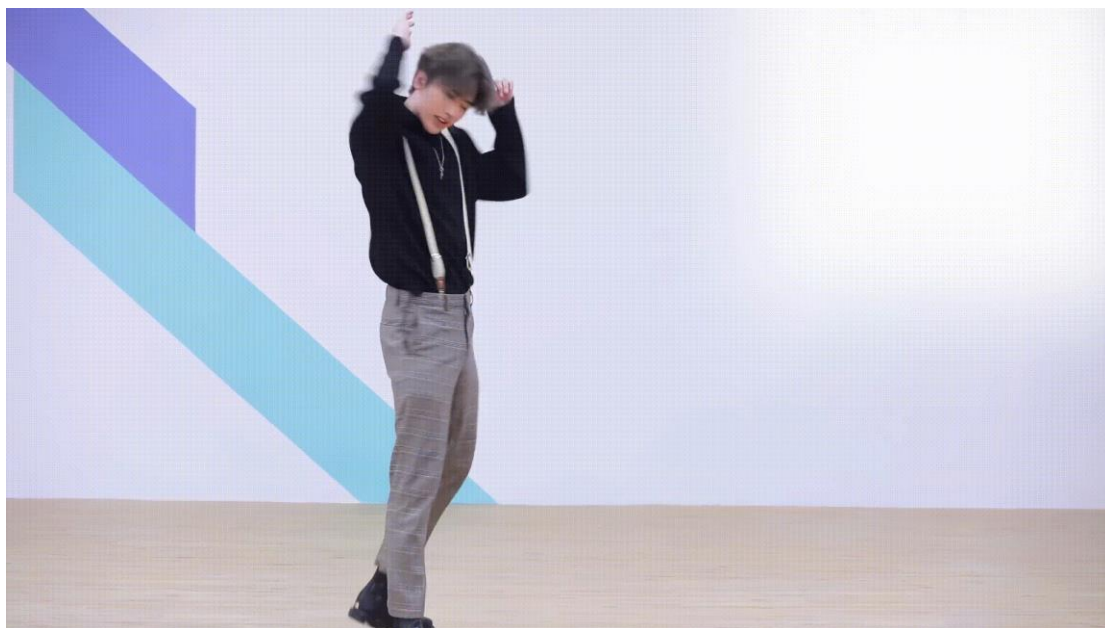


图 2 《只因你太美》舞步

于是我们在想，通过前期 Taichi DEM 的工作，加上一些新的功能，来实现这一动作效果。然而图 1 的兔子是自由下落，图 2 的蔡老师的动作要更加复杂。因此，我们需要添加 2 个功能：(1) 将特定的颗粒分为一组 (group)，并对组内所有颗粒施加外部体力 (external body force)；(2) 在特定时间内进行上述操作，即存取一个序列 (sequence)，在仿真过程中调用。

本次比赛我们团队将编码迭代 Taichi DEM，实现上述 2 个功能，并借助其他工具和方法，准备初始试样、调节胶结参数、划定颗粒范围、调节给力 sequence，最终实现图 2 的效果。具体技术路线详见下一节。

2 技术路线

2.1 初始试样准备——二维非凸区域颗粒填充算法推导

本项目第一步是制备初始帧的情况下蔡老师的软体试样。截取蔡老师舞步开始的状态, 然后通过 Adobe Photoshop^[3] 软件的磁性套索功能选区, 如图 3 所示。将选区内处理为黑色像素块 (图 4), 为一个二维平面封闭区域 A 。利用 OpenCV^[4] 的库函数对区域 A 描边, 得到顺时针相连的二维点序列 $A_{b,i} = [x_i \ y_i]^T$ 。需要说明的是, 本次比赛项目使用的 Taichi DEM 是三维仿真, 但本例展示二维平面效果, 垂直屏幕的 z 方向的动力学参数均为零。



图 3 初始舞步



图 4 黑色像素块区域 A

接着, 对上述得到的封闭区域填充颗粒。为简单起见, 颗粒为等大小球体, 半径 $r = 1.0 \text{ px}$, 呈正交规则排列 (图 5)。使用蒙特卡洛方法获得填充区域 A 的球心, 具体逻辑为:

$$\forall \mathbf{X}_j = [x_j \ y_j]^T \begin{cases} \in A & \text{保留} \\ \notin A & \text{舍去} \end{cases} \quad (1)$$

式中 \mathbf{X}_j 为球心。而对 $\mathbf{X}_j \in A$ 的判断, 可找到距离 \mathbf{X}_j 最近的点 $A_{b,i}$, 令 $B_1 = A_{b,i} -$

$A_{b,i-1} = [B_{x1} \ B_{y1}]^T, B_2 = A_{b,i+1} - A_{b,i} = [B_{x2} \ B_{y2}]^T$, 然后通过下式:

$$\begin{cases} (RB_1)^T B_2 < 0 & \text{若 } \forall k \in \{1, 2\} \text{ 有 } B_{yk}(x_j - x_i) - B_{xk}(y_j - y_i) > 0, \text{ 则 } X_j \in A \\ (RB_1)^T B_2 > 0 & \text{若 } \exists k \in \{1, 2\} \text{ 有 } B_{yk}(x_j - x_i) - B_{xk}(y_j - y_i) > 0, \text{ 则 } X_j \in A \end{cases} \quad (2)$$

式中 $R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ 为二维旋转矩阵, 可将向量逆时针旋转 90° ; $(RB_1)^T B_2 < 0$ 和 $(RB_1)^T B_2 > 0$ 分别对应 $A_{b,i}$ 为凸点和凹点的情况 (图 6), 要分情况判断; $B_{yk}(x_j - x_i) - B_{xk}(y_j - y_i) > 0$ 表明 X_j 位于方向向量为 $[B_{xk} \ B_{yk}]^T$ 、经过点 $A_{b,i}$ 的直线的右侧; 而对于 $A_{b,i}$ 为凸点和凹点的情况下 $\forall k \in \{1, 2\}$ 与 $\exists k \in \{1, 2\}$ 的不同, 图 7 可以清楚说明此问题。

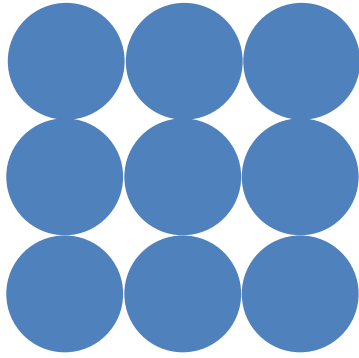


图 5 正交规则排列颗粒

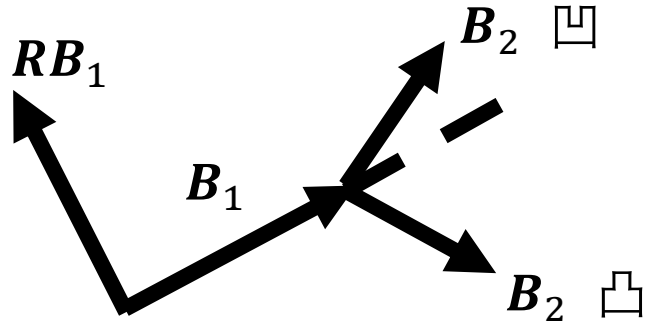
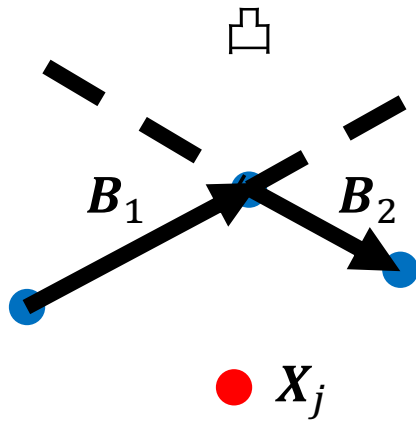
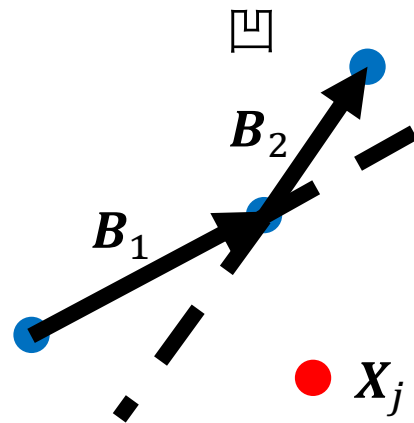


图 6 判断 $A_{b,i}$ 为凸点或凹点, 并分别处理



X_j 需同时在两线段
所在直线右侧



X_j 只需在其中任意一条
线段所在直线右侧

图 7 $A_{b,i}$ 为凸点与凹点的不同情况下, $X_j \in A$ 的逻辑处理

但是实测发现, 会有一些 corner case 需要处理。一个是 $A_{b,i-1}, A_{b,i}, A_{b,i+1}$ 三

点共线的情况 (图 8), 会导致 $(\mathbf{R}\mathbf{B}_1)^T \mathbf{B}_2 \equiv 0$ 。此时只需将 $(\mathbf{R}\mathbf{B}_1)^T \mathbf{B}_2 = 0$ 的临界情况并入式 (2) 中 $(\mathbf{R}\mathbf{B}_1)^T \mathbf{B}_2 > 0$ 与 $(\mathbf{R}\mathbf{B}_1)^T \mathbf{B}_2 < 0$ 两个条件的其中之一即可。另一个情况是 $(\mathbf{X}_j - \mathbf{A}_{b,i}) \perp \mathbf{B}_k, k = 1, 2$, 如图 9 所示, 此时 $B_{yk}(x_j - x_i) - B_{xk}(y_j - y_i) \equiv 0$ 。此时只需将式 (2) 中的 $B_{yk}(x_j - x_i) - B_{xk}(y_j - y_i) > 0$ 更改为 $B_{yk}(x_j - x_i) - B_{xk}(y_j - y_i) \geq 0$ 即可 cover 上述情况。考虑 corner case 后, 式 (2) 变为式 (3):

$$\begin{cases} (\mathbf{R}\mathbf{B}_1)^T \mathbf{B}_2 \leq 0 & \text{若 } \forall k \in \{1, 2\} \text{ 有 } B_{yk}(x_j - x_i) - B_{xk}(y_j - y_i) \geq 0, \text{ 则 } \mathbf{X}_j \in \mathbf{A} \\ (\mathbf{R}\mathbf{B}_1)^T \mathbf{B}_2 > 0 & \text{若 } \exists k \in \{1, 2\} \text{ 有 } B_{yk}(x_j - x_i) - B_{xk}(y_j - y_i) \geq 0, \text{ 则 } \mathbf{X}_j \in \mathbf{A} \end{cases} \quad (3)$$

使用式 (3) 即可完成试样制备 (图 10)。试样含颗粒数 3587。

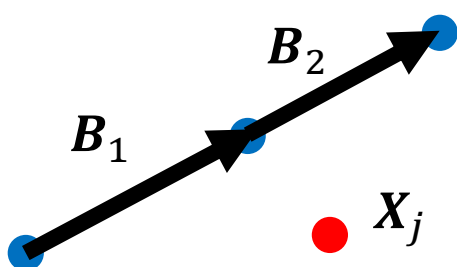


图 8 Corner case: 三点共线

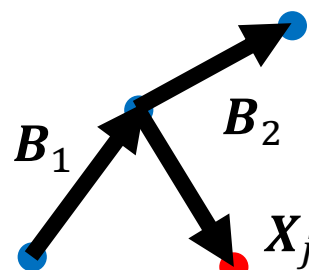


图 9 Corner case: 垂直情况

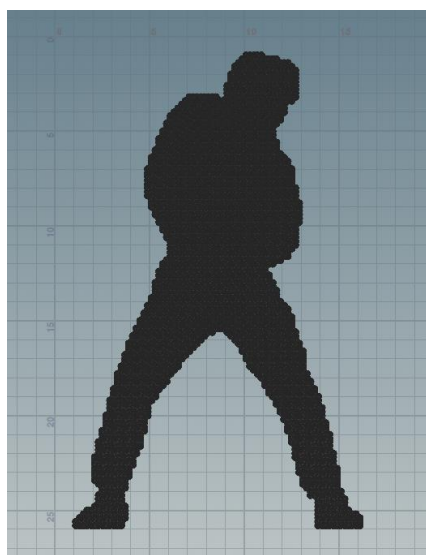


图 10 制备完成的试样 (Houdini^[5]渲染, 打开 input_data/visualizer_input.hipnc 可查看)

团队成员 Weishao 提出, 图 10 的试样中, 所有颗粒均为黑色。为什么蔡老师的白色背带没有被标记出来? 标记背带范围的颗粒 ID 并不是一件难事。遍历

图 10 中的每个颗粒中心点位置，然后在图 3 中找到对应像素点，以下式作为判断是否为背带像素点：

$$p_R > t \text{ 且 } p_G > t \text{ 且 } p_B > t \quad (4)$$

式中 p_R, p_G, p_B 分别为像素点的红、绿、蓝值， t 为控制阈值，本例中取 $t = 160$ 。记录背带像素点对应颗粒 ID，并存放于 input_data/belt.txt 文件中。得到的带背带试样如图 11 所示，图 11 使用 BIMBase 渲染。BIMBase 是中国建筑科学研究院有限公司下属北京构力科技有限公司自主研发的 BIM 图形平台，基于该平台二次开发，可以导入本项目数据文件、可视化并交互处理^[6]。

团队成员 MuGdxy 提出，既然我们标记背带，为何不把原图 3 中的像素点颜色 map 到图 10 的颗粒中呢？于是进一步迭代，对每个颗粒，寻找其中心点坐标对应像素点，类似使用取色管工具，取出每个像素点的 p_R, p_G, p_B ，和颗粒 ID 关联存放于 input_data/color_diffuse.csv 中。这样可以使试样的视觉效果接近真实的蔡老师，如图 12 所示。

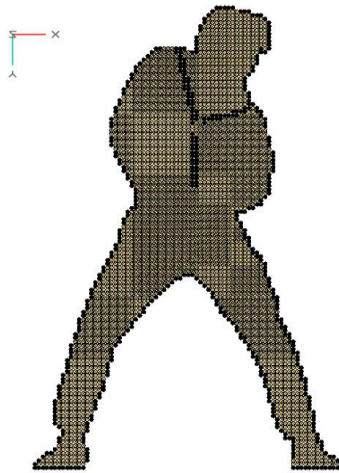


图 11 带背带试样 (BIMBase 渲染)

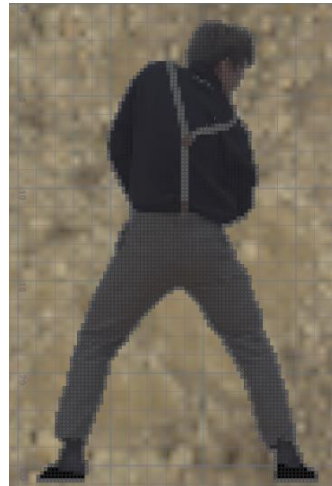


图 12 像素颜色 map 后试样

颗粒填充算法为自行推导，制样代码位于工程文件夹下 specimen 目录中。

2.2 Taichi DEM 代码整理

因为先前 Taichi DEM 代码挑战赛规则要求，所有代码合入单个文件中，但是随着代码量的增多，单个文件显得臃肿而维护不便，所以我们将各模块拆分为不同文件，并在 **Taichi Hackathon 2022** 中改动、升级过的部分以注释“**Taichi Hackathon 2022**”进行标记。

先前 Taichi DEM 中，爱丁堡胶结模型 (EBPM) 与 Hertz-Mindlin 接触模型位于同一个 kernel，利用 if 进行分支，这对于 GPU 并行是一个痛点。本项目我们将其拆分为两个不同的 kernel，实现了胶结颗粒-颗粒接触与非胶结颗粒-颗粒接触的解耦，以更好地匹配 GPU 架构，更佳发挥计算性能。此外，先前项目因为 EBPM 与 Hertz-Mindlin 模型有耦合，胶结与非胶结接触每一个时间步均需要通过宽碰撞检测 (BPCD)，而胶结受拉可以变长，使得最大可接触颗粒间距比其半径之和要大，故 BPCD 时划分的 cell size 与胶结最大拉伸应变 $\epsilon_t = \sigma_t/E_b$ 呈线性正相关以确保不会漏检，式中 σ_t 为胶结抗拉强度， E_b 为胶结弹性模量。但如上一段所述，对于抗拉强度 σ_t 极大、不会破坏的胶结，计算得到的 cell size 将会很大，反过来拖慢计算速度。在 EBPM 与 Hertz-Mindlin 模型解耦后，由于胶结接触的生成是单步命令“只跑一次”，可单独处理，宽碰撞检测可只对非胶结接触颗粒对展开，此时 cell size 的算法将与胶结参数无关，可以接近于略大于 $4r_{max}$ ，式中 r_{max} 为颗粒最大半径。然而，这样做的后果是，Hertz-Mindlin 模型与 EBPM 模型独立判断，以并联的形式进行颗粒间接触传力。试想两个已经胶结的颗粒，因为本次算例为软体试样，胶结的压缩或者拉伸变形会很大，完全可能出现两个胶结颗粒运动过程中物理上发生接触，在 EBPM 接触模型的基础上并联 Hertz-Mindlin 模型的情况。这是一种错误的力学响应，会导致错误的结果。不过，本次算例的

所有颗粒间接触均为胶结接触且不会发生破坏，所以，考虑关闭 Hertz-Mindlin 模型的力学响应及对应的宽碰撞检测代码，在每一个时间步内并行计算所有胶结接触即可。在以后的使用中，会添加接触模型选择与组合功能。

2.3 “时序-分组-体力” (SGF) 列表

这是 Taichi Hackathon 2022 中实现的新的特色功能。前面提到了本次 Hackathon 我们需要实现 2 个功能，即颗粒分组赋予体力与其时序控制。为此，我们构建了“时序 (Sequence) - 分组 (Group) - 体力 (Force)”列表，简称 **SGF 列表**，作为实现这 2 个功能的数据结构。SGF 列表由 CSV 文件记录，程序运行时，CSV 文件并行读取。CSV 文件名记录时序的起点步和终点步，CSV 文件内记录颗粒分组与所赋值体力（以全局坐标系 x, y, z 分量表示）的关系表。**SGF 列表操作的相关 Utility 代码在 dem/TimeSequence.py 文件中。**

在项目设计过程中，团队遇到了两种技术方案：是应该采用空间包围盒 (box range)，还是应该采用颗粒 ID 号，作为颗粒分组的准则？经过讨论，我们认为应该使用颗粒 ID 号。虽然初始帧的时候，给定一个空间包围盒，可以得到确定的颗粒集合，但是由于软体变形很大，在仿真开始后，该给定包围盒内的颗粒集合可能发生变动，这会导致分组的不稳定。而使用颗粒 ID 集合的话，则没有此问题。这也是 DEM 是基于颗粒的 (particle based) 数值方法这一特点所决定的。此外，给定颗粒 ID 的另一个好处是，后面调试确定具体给予试样的力值时，我们现成的辅助工具使用起来更加方便，具体将在 2.4 节中叙述。

给颗粒赋予体力的功能，类似于开源 DEM 软件 LIGGGHTS 的“fix setforce”命令^[7]。然而，LIGGGHTS 该命令并无时序功能，仅为单步命令重置颗粒所受合力。

2.4 调试参数，确保试样正确响应

在代码完成后，我们需要仔细调节试样的参数，包括胶结的强度、模量以确保其为一个软体；更重要的是，我们需要调出一个较好的“时序-分组-力”列表，以确保软体的运动足够接近蔡老师的舞步。

胶结参数方面，因为岩石类胶结材料的弹性模量大约是GPa数量级，本项目我们将其降低2个数量级，为 $E_b = 10\text{MPa}$ ；设定胶结抗压、抗拉、抗剪强度 $[\sigma_{cb}] = [\sigma_{tb}] = [\tau_b] = 10E_b = 100\text{MPa}$ ，以保证胶结不会破坏。

为了更加方便选取颗粒分组，我们使用BIMBase二次开发插件^[6]来作为辅助，包括点击查看具体某个颗粒的ID(图13)：

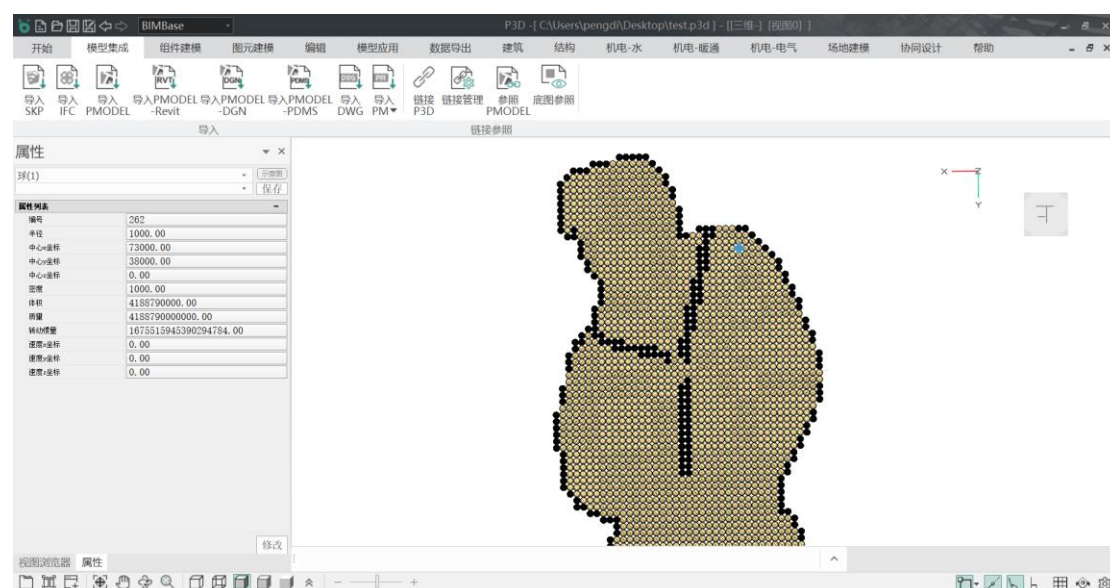


图 13 BIMBase 查阅颗粒 ID

因为图像坐标系与屏幕笛卡尔坐标系 y 轴反向，所以图13的试样相比图3沿 y 轴对称，而图10-12已进行了翻转处理，所以显示同图3。虽然在BIMBase中，也可以框选若干颗粒，但是我们的SGF表中，颗粒组并非全部为矩形框范围(图14)，SGF对颗粒分组使用包围盒的技术路线将会十分不灵活。这也说明了SGF颗粒分组使用颗粒ID的优势。

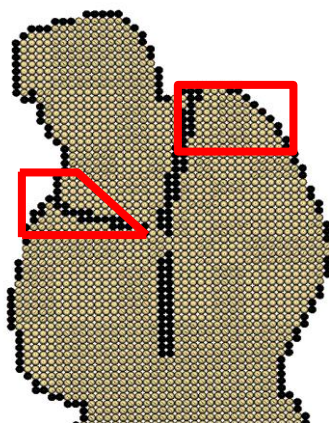


图 14 颗粒分组选区

经过调试和试算，我们得到 Demo 如图 15。试样近似反映了蔡徐坤老师舞步的动作。要复现此 Demo，只需在 repo 根目录运行 `launch.py`，运行完成后使用 Houdini 打开同目录下的 `visualizer.hipnc` 即可。

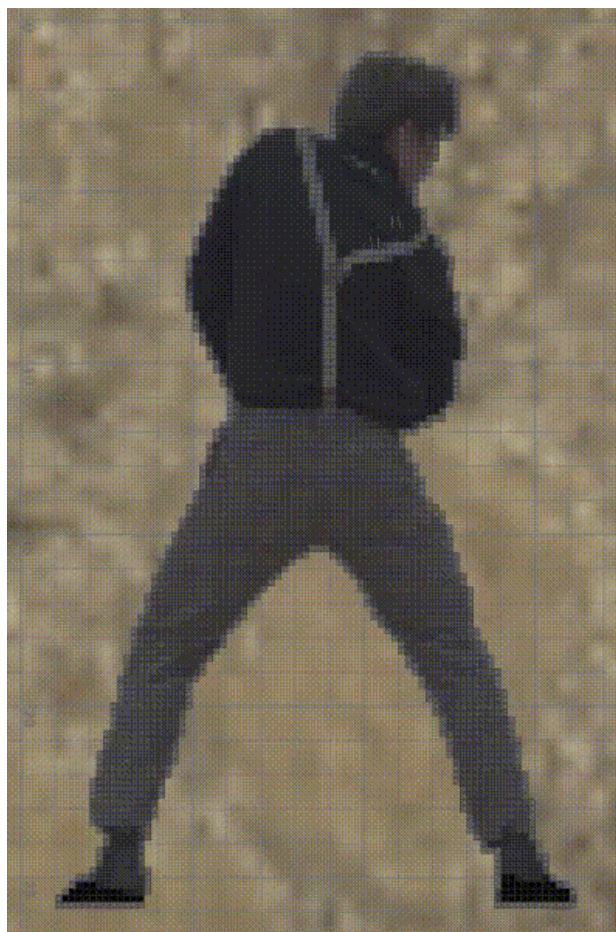


图 15 最终 Demo

3 项目创新点、特色与展望

本项目基于先前 Taichi DEM 成果,主要实现了“时序-分组-力”(SGF) 列表,用于仿真过程中颗粒分组赋予体力与其时序控制,从而实现拟动画效果。此外,自行推导的二维非凸区域蒙特卡洛颗粒填充制样算法具有较好的鲁棒性。

需要说明的是,颗粒分组赋予体力与时序控制两个功能,分别在**工业生产和创意设计**中有实际需求。对于折叠屏而言,如要通过离散元数值仿真研究其折叠区裂纹拓展机理,需要模拟折叠屏由开到闭的过程,此时需要对折叠屏两端的颗粒给与固定的体力,以使屏幕合上。如使用墙单元给力,因为墙是理想完美平整的,墙与由不规则颗粒胶结而成的屏幕表面的接触和传力并不稳定(原因同 2.2 节),更多的时候折叠屏会被挤飞出去,导致模拟进行不下去。因此,有必要加入外部体力的功能。此外,在特定时间对特定范围对象操作,这样一个序列类型的数据,非常像三维动画工程文件存储的元数据,例如在第 100 到第 110 帧,对某个角色的手进行某项操作。因此,定义并实现一个序列数据的存取机制,相当于实现了相关创意设计软件的数据研发 Demo。上述两个应用场景为本项目提供了工程支撑,赛后我们将把本次比赛的成果进一步用于工业生产中去。

参考文献

- [1] 【活动已结束】 Taichi DEM 代码优化挑战赛来了! AirPods 胶囊咖啡机还有更多社区周边等你来拿~ <https://forum.taichi-lang.cn/t/topic/2975/21>.
- [2] 【4K 顶级画质 60FPS】蔡徐坤《只因你太美》原版完整版现场! 一晃眼 6 年过去了. <https://www.bilibili.com/video/BV1ct4y1n7t9>.
- [3] Official Adobe Photoshop | Photo and Design Software.
<https://www.adobe.com/products/photoshop.html>.
- [4] Bradski, G. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000, 2236121. <https://github.com/itseez/opencv>.
- [5] Houdini | 3D Procedural Software for Film, TV & Gamedev | SideFX.
<https://www.sidefx.com/products/houdini>.
- [6] BIMBase 开发平台 – BIMBase Platform. <https://app.pkpm.cn>.
- [7] fix setforce command — LIGGGHTS v3.X documentation.
https://www.cfdem.com/media/DEM/docu/fix_setforce.html.