To cite this version :

Hugo MEDELLIN-CASTILLO, C. FLETCHER, Theodore LIM, James M. RITCHIE, Samir GARBAYA - An evaluation of physics engines and their application in haptic virtual assembly environments - In: The 37th International MATADOR Conference, Royaume-Uni, 2012-07-25 - The 37th International MATADOR Conference held in Manchester 25th – 27th July, 2012 - 2012

An evaluation of physics engines and their application in haptic virtual assembly environments

G. González-Badillo[1], H. I. Medellín-Castillo[1], C. Fletcher[2], T. Lim[2], J. Ritchie[2], S. Garbaya [3]
[1] Universidad Autónoma de San Luis Potosí, S.L.P., México
[2] Innovative Manufacturing Research Centre, Heriot-Watt University, Edinburgh, UK
[3] Arts et Métiers ParisTech, Le2i, CNRS, Institut Image, 2 Rue Thomas Dumorey, 71100 Chalon-sur-Saône, France

Abstract. Virtual Reality (VR) applications are employed in engineering situation to simulate real and artificial situations where the user can interact with 3D models in real time. Within these applications the virtual environment must emulate real world physics such that the system behaviour and interaction are as natural as possible and to support realistic manufacturing applications. As a consequence of this focus, several simulation engines have been developed for various digital applications, including VR, to compute the physical response and body dynamics of objects. However, the performance of these physics engines within haptic-enabled VR applications varies considerably. In this study two third party physics engines - Bullet and PhysX^{tm}- are evaluated to establish their appropriateness for haptic virtual assembly applications. With this objective in mind five assembly tasks were created with increasing assembly and geometry complexity. Each of these was carried out using the two different physics engines which had been implemented in a haptic-enabled virtual assembly platform specifically developed for this purpose. Several physics-performance parameters were also defined to aid the comparison. This approach and the subsequent results successfully demonstrated the key strengths, limitations, and weaknesses of the physics engines in haptic virtual assembly environments.

Keywords: virtual reality (VR); physics engine; Bullet; PhysX; haptics; haptic assembly; virtual assembly.

1. Introduction

Physical based modelling (PBM) uses physics simulation engines to provide dynamic behaviour and collision detection to virtual objects in virtual environments emulating the real world. This results in better appreciation and understanding of part functionality and can also lead to improved training of manual tasks [1,2]. However, there are several challenges when haptics is integrated with physics engines, e.g. synchronization, non-effective collision detection, high computational cost and a negative impact on the performance of the application [3]. This is due to the fact that simulation engines are not adapted to haptic rendering, mainly because the typical frequency of haptics simulations is over 1 kHz and around 100 Hz for physics simulations [4,5].

This work presents an evaluation of two physics engines for haptic environments to assess their performance in haptic assembly tasks. The experiments are aimed to identify the strengths and weaknesses of each simulation engine.

2. Related work

Physics simulation engines have been used in many applications from computer games through to movies. Laurell [6] identified five key points in any physics engine: contact detection, contact resolution, force calculation, integrating motion and the impact of real time constraints (time step) where anything below 25 frames per second (fps) is perceived as slow and stammering. Additionally, the update rate of the whole system, both graphics and physics, must be less than 40 milliseconds per cycle.

Howard and Vance [7] found that while mesh to mesh assembly enabled accurate collision detection, realistic physical response was not demonstrated particularly when objects had continuous contact with each other since excessive surface stickiness and model penetration was observed. The physics update rate was found to be directly related to the number of contacts generated between colliding geometries.

Seth, et al. [8] identified three main challenges that virtual assemblies must overcome to increase the level of realism: collision detection, inter-part constraint detection and physics-based modelling.

Seugling and Rölin [3] compared three physics engines - Newton, ODE and PhysX - against the following run-time executions: friction on a sliding plane, gyroscopic forces, restitution, stability and scalability of constraints, accuracy against real, scalability of contacts (pile of boxes), stability of piling (max number of stacked boxes), complex contact primitive-mesh, convex-mesh and mesh-mesh. According to their results PhysX was the best evaluated simulation engine except in the stability of piling test and the mesh-mesh collision detection due to unwanted behaviour.

Boeing and Bräunl [9] carried out an investigation to compare PhysX (formerly Novodex), Bullet, JigLib, Newton, ODE, Tokamak and True Axis using PAL (Physics Abstraction Layer). Their comparison criteria included: integrator performance, material properties, friction, constraint stability, collision system and the stacking test. They concluded that PhysX had the best integrator method whereas Bullet provided the most robust collision system.

On the other hand Coumans and Victor [10] made a simple comparison analysis of the following physics engines: PhysX, Havok, ODE and Bullet. Collision detection and rigid body features were used as the comparison criteria. According to the authors PhysX was the most complete engine.

Glondu et al. [4] introduced the possibilities of implementing a modular haptic display system that relies on physical simulation and haptic rendering. With this in mind, four physical simulation libraries are evaluated: Havok, PhysX, Bullet and OpenTissue. The performance criterion was based on computation time, stability and accuracy. PhysX showed penetration in some of the tests whilst Havok showed the best average computation time, stability and friction accuracy.

The previous background study has revealed that several research works have been conducted to evaluate different simulation engines. In general, it is concluded that PhysX is the most complete simulation engine. However, these works have not considered the use of haptic rendering in the virtual environment being evaluated. Thus, it can be said that the performance evaluation of simulation engines in haptic enabled virtual environments is still needed. Hence, the objective in this work is to conduct a series of experiments to find the most appropriate simulation engine for a specific haptic application. It is envisaged that the work reported in this paper can contribute to the haptic research community.

3. System overview

A haptic assembly virtual platform, named as HAMMS, has been developed and is shown in Fig. 1. The HAMMS system (Fig. 1) comprises the Visualization Toolkit libraries (VTK 5.8.0) and the Open Haptics Toolkit v3.0. Two physics engines i.e., PhysX^{TM} v. 2.8.4 and Bullet v. 2.79, have been integrated and the user can select between the two during run time. Single and dual haptic is provided using Sensable's Omni haptic device. One of the main characteristics of HAMMS is a control panel where the user can modify in real time simulation parameters; haptic properties like stiffnes, damping and friction; and physical properties like mass, restitution, tolerance, etc.

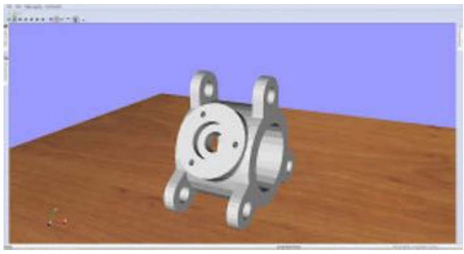


Fig. 1. Virtual haptic assembly application

4. Comparative analysis

In order to identify the usefulness and capability of the two physics engines in haptic virtual assembly environments, a set of virtual assembly tasks were defined and carried out using the two physics engines.

4.1. Model representation

Collision detection is a key aspect of assembly analysis and it is directly related to the model representation in the physic simulation engine [11]. Assembly tasks may comprise several objects or components with different shapes. In general, objects can be divided into two groups: convex and concave objects, being the last the most common objects in assembly tasks.

Bullet 2.79 use GIMPACT libraries to calculate collisions for concave objects represented by a triangular mesh, its representation is very similar to the graphic model as shown in Fig. 2. A convex decomposition algorithm such as HACD [12] can also be used to create concave shapes.

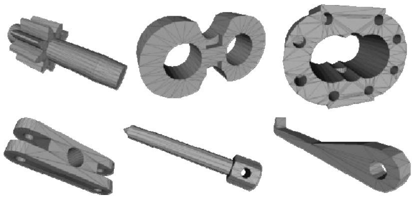


Fig. 2. Physic representation of objects using GIMPACT

PhysX v2.8.4 does not support collision detection for triangular meshes; however, an algorithm to create a concave compound object from a triangle mesh, convexFT (CFT), is provided. The algorithm transforms each triangle of the mesh into a convex element, so the final shape has as many convex hulls as triangles in the original mesh.

4.2. Assembly tasks

Five assembly tasks were selected to analyse the performance of each physics engine in HAMMS:

(1) A pile of boxes assembly task was selected to evaluate the manipulation and performance of primitive

shapes, it is also used to analyse the simulation engine performance and stability where multiple and accumulative contacts are considered, Fig. 3 (a).

(2) The packing boxes assembly task, Fig. 3 (b), is useful to identify the physics engine performance using different representation algorithms such as convex decomposition or triangular meshes. The purpose of this task is to observe the collision response and stability when multiple contacts in different directions are present.

(3) The peg and hole assembly task which is commonly used in assembly tests because it represents a generalized case of cylindrical parts' assembly, Fig. 3 (c).

(4) A more complex pump assembly task, Fig. 4a, and (5) a bearing puller [Fig. 4 (b)] are selected as they represent the virtual models of real components with complex shapes.


(a) (b) (c)

Fig. 3. Assembly tasks: a) Pile of boxes b) Packing box c) Peg & hole

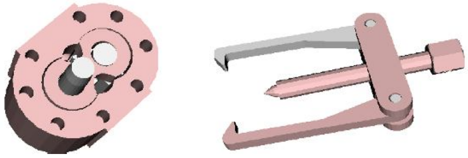

(a) (b)

Fig. 4. Assembly tasks: a) Pump b) Bearing Puller

Each of these five assembly tasks was performed by an experienced user in both haptics and virtual assembly. Five repetitions were carried out for each task, all the tests were performed using a single haptic device to manipulate virtual objects and the mouse to manipulate the camera.

In a physics simulation engine, the integrator method is refered to the numerical methods that it uses for calculate the new position of the object on each time step during the simulation. In order to assess the integrator performance under different conditions, virtual free fall experiments were carried out, measuring the time on reach the floor when the object was dropped from an elevation of 500 units.

5. Results and discussion

The results of the free fall test are shown in Table 1, where it is shown that when the number of triangles of the model is smaller than 300 the integrator performance of PhysX is not affected, whereas in the case of Bullet an increment of 50% was observed. When the object comprises about 2000 triangles the time performance is greatly affected (about 100% increase) for both Bullet and PhysX.

Table 1. Free-falling time with respect to shape (number of triangles)

Shapes	Triangles	Bullet (sec)	PhysX (sec)
Box	12	0.993	2.11
Pin	44	1.011	2.105
Big Cog	276	1.445	2.115
Housing	1934	2.918	4.199

Table 2 shows the influence of the model representation on time performance, these results were obtained with the haptic rendering loop on. The results indicate that when primitive shapes are used in PhysX, free fall time is 5.6 seconds compared to Bullet's 0.9 second average. ConvexFT and convex decomposition (HACD) [12] showed similar results and the best performance for PhysX. Bullet showed a time increment related to the increment of the model shape complexity. Model representation using primitives showed the best performance.

Table 2. Free-falling time with respect to model representation

Model	Representation	Bullet (sec)	PhysX (sec)
Box	Box	0.999	5.627
	Trimesh- ConvexFT	0.993	2.11
Pin	Cylinder	0.998	5.608
	Trimesh- ConvexFT	1.011	2.105
Cog	Trimesh- ConvexFT	1.445	2.115
	Convex dec. HACD	1.428	2.09

Table 3 shows the percentage of increase in time in the free-falling test when the haptic rendering loop was running with respect to a situation where only physics and graphics loops were running, Bullet showed a time increment of 50% when the haptic rendering loop was on and the model was complex, compared to PhysX that showed only an increment of 2%; however, the falling time in all test was smaller using Bullet than using PhysX, this suggest that PhysX rendering loop is more adapted to be used together with haptics. The theoretical falling time is 0.316 seconds.

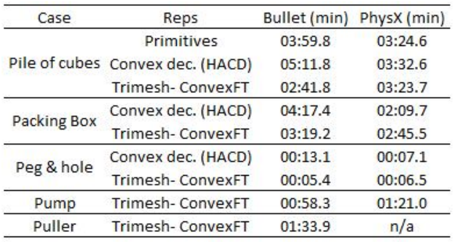
Table 3. Influence of the haptic loop on free-falling time (%)

Model	Representation	Bullet	PhysX
Box	Box	3.65	0.195
Pin	Cylinder	0.91	0.139
Cog	Trimesh	50.42	2.16
	Convex Dec HACD	48.28	1.08

Table 4 presents the task completion time (TCT) in minutes for each assembly tasks, different model representations and each physics engine. It can be observed that for the assembly tasks of pile of cubes, packing box and peg & hole, when primitives or convex decomposed model representation is used, PhysX posted the least TCT, however when using triangular meshes and Bullet, TCT was least in all the tasks, except the packing box due to unnatural collision response. Real and virtual tests were carried out on the pump assembly. A mean

TCT of 37 seconds was obtained in the real assembly task whilst in the virtual platform the TCT value was 58.3 seconds using Bullet (56% more than the real assembly) and 1.21 minutes using PhysX, this difference may be due to several factors such as the manipulation of virtual models trough the haptic device, physical properties of materials (friction, restitution, mass, etc.).

Table 4. Task completion time

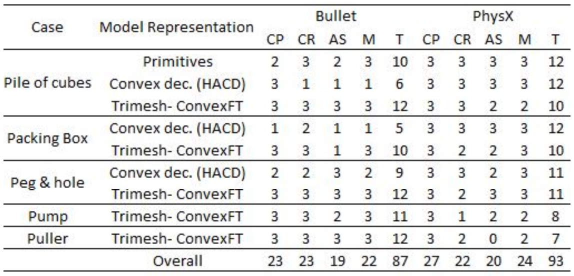


Case	Reps	Bullet (min)	PhysX (min)
Pile of cubes	Primitives	03:59.8	03:24.6
	Convex dec. (HACD)	05:11.8	03:32.6
	Trimesh- ConvexFT	02:41.8	03:23.7
Packing Box	Convex dec. (HACD)	04:17.4	02:09.7
	Trimesh- ConvexFT	03:19.2	02:45.5
Peg & hole	Convex dec. (HACD)	00:13.1	00:07.1
	Trimesh- ConvexFT	00:05.4	00:06.5
Pump	Trimesh- ConvexFT	00:58.3	01:21.0
Puller	Trimesh- ConvexFT	01:33.9	n/a

The results obtained for haptic and physics update rates indicate that PhysX offers better update rates when using non complex geometries represented by primitives or by convex decomposition. However Bullet physics showed better update rates when simulating complex parts represented as triangular meshes.

Assembly performance parameters were evaluated using a scale from 0 to 3, where 0 represents the worst performance and 3 the best. Users assign a value to each parameter according to their perception of the assembly task. Performance parameters include: Collision precision (CP) indicates penetration of virtual models when colliding with other virtual objects. Collision response (CR) is the reaction and how natural objects behave. Assembly stability (AS) indicates if the objects are stable once the assembly is completed. Manipulability (M) indicates how easy the models can be manipulated, and the total (T) indicates the overall score of CP+CR+AS+M. The results are shown in Table 5.

Table 5. Performance evaluation



Case	Model Representation	Bullet					PhysX				
		CP	CR	AS	M	T	CP	CR	AS	M	T
Pile of cubes	Primitives	2	3	2	3	10	3	3	3	3	12
	Convex dec. (HACD)	3	1	1	1	6	3	3	3	3	12
	Trimesh- ConvexFT	3	3	3	3	12	3	3	2	2	10
Packing Box	Convex dec. (HACD)	1	2	1	1	5	3	3	3	3	12
	Trimesh- ConvexFT	3	3	1	3	10	3	2	2	3	10
Peg & hole	Convex dec. (HACD)	2	2	3	2	9	3	3	2	3	11
	Trimesh- ConvexFT	3	3	3	3	12	3	2	3	3	11
Pump	Trimesh- ConvexFT	3	3	2	3	11	3	1	2	2	8
Puller	Trimesh- ConvexFT	3	3	3	3	12	3	2	0	2	7
	Overall	23	23	19	22	87	27	22	20	24	93

It is notable that PhysX displayed better performance than Bullet in simple assembly tasks such as the pile of cubes, packing box and peg and hole. However, in complex assembly tasks like the pump and puller assembly, Bullet showed better performance, less assembly time (58.3 seconds) and better evaluation by the user (11 points of 12 possible) than PhysX (assembly time 1:21.0 min and a total evaluation of 8 points). Moreover, in PhysX the puller assembly tasks could not be completed because the puller screw could not be inserted in the puller base, due to a poor model representation.

6. Conclusions and future work

A performance evaluation of two different physics simulation engines for haptic assembly has been presented. The results have suggested that for assembly tasks that involve non complex geometries like boxes and cylinders (primitives), the use of PhysX offers a better performance than Bullet; however when the assembly comprises more complex shape components, Bullet has better performance than PhysX. A more comprehensive study must be carried out including the effect of simulation parameters, the use of a dual haptics configuration, and others physics simulation engines such as Havok or ODE.

Acknowledgments: Authors acknowledge the financial support from CONACYT (National Science and Technology Council of Mexico) and from SIP-UASLP (Research and Postgraduate Secretariat of Universidad Autonoma de San Luis Potosi, Mexico).

References

[1] Zerbato D, Baschirotto D, Baschirotto D, Debora Botturi D, (2011) GPU-based physical cut in interactive haptic simulations. Internetional Journal CARS 6:265-272.

[2] Wang Y, Jayaram S, Jayaram U, (2001) Physically based modeling in virtual assembly. The int. journal of virtual reality vol 5, No. 1, pp. 1-14.

[3] Seugling A, Rölin M, (2006) Evaluation of physics engines and implementation of a physics module in a 3D auhoring tool. Master's thesis in computing science Umea University.

[4] Glondu L, Marchal M, Dumont G, (2010) Evaluation of physical simulation libraries for haptic rendering of contacts between rigid bodies. Proceedings of ASME , WINVR 2010, may 12-14, Ames, Iowa, USA, WINVR201-3726, pp. 41-49.

[5] Ritchie JM, Lim T, Medellin H, Sung RS, (2009) A haptic based virtual assembly system for the generation of assembly process plans. XV congreso internacional SOMIM, 23- 25 September 2009, Cd. Obregon, Sonora. México.

[6] Laurell B, (2008) The inner workings of real-time physics simulation engines. IRCSE '08 IDT workshop on interesting results in computer science and engineering; Mälardalen univerity, Vasteras, Sweden.

[7] Howard BM, Vance JM, (2007) Desktop haptic virtual assembly using physically based mod.. Vir. Rea. 11:207-215.

[8] Seth A, Vance JM, Oliver JH, (2011) Virtual reality for assembly methods prototyping: a review. Virtual rea. 15:5-50.

[9] Boeing A, Bräunl T, (2007) Evaluation of real-time physics simulation systems. Graphite '07 Proc. of the 5th Int. Conf. on Comp. graph and int tech. Aus. and SW Asia, pp: 281-288.

[10] Coumans E, Victor K, (2007) COLLADA Physics. Web3D '07 Proc. of the 12 int. Conf. on 3D web tech., pp. 101-104.

[11] Chen T, (2010) Virtual Assembly of mechanical component and collision detection. Int. Conf. on electronics and information engineering (ICEIE 2010) IEEE vol.1: 443-447.

[12] Khaled M, Faouzi G, (2009) A simple and efficient approach for 3D mesh approximate convex decomposition, 16th IEEE Int. Conf. on image processing (ICIP '09), Cairo, 7-10 November 2009.