A Benchmark of Force Quality in Haptic Rendering

Mikel Sagardia, Thomas Hulin, Carsten Preusche, and Gerd Hirzinger

DLR (German Aerospace Center)
Member of the Helmholtz Association
Institute of Robotics and Mechatronics
{Mikel.Sagardia, Thomas.Hulin, Carsten.Preusche, Gerd.Hirzinger}@dlr.de

Abstract. Several haptic rendering algorithms have been developed in the last years and a comparative study is needed to select the appropriate algorithm for each application. This work presents a set of testing scenarios and indicators for benchmarking haptic rendering algorithms using synthetically generated and recorded realistic assembly paths. Synthetic scenarios aim to objectively evaluate haptic rendering algorithms by comparing their response with expected analytical behaviors. Real assembly scenarios are oriented to evaluate the algorithms in complex environments. Examples of the benchmark are given using an improved version of the Voxmap-PointShell Algorithm.

Keywords: Haptics, Collision Detection, Force Computation, Benchmarking, Evaluation.

1 Introduction

Virtual Reality (VR) simulations with haptic feedback make possible training mechanics and verifying assembly processes without physical prototypes. In the last years several haptic rendering algorithms for such VR simulations have been developed. Choosing the most appropriate algorithm for each application calls for a standard benchmark suite. Yet, the literature on the topic is quite limited.

A guideline to classify and empirically evaluate force-feedback algorithms is introduced in [3], but without specifying benchmark scenarios and indicators. In [4] an evaluation method for haptic rendering systems is presented, comparing the force response with previously measured physical forces. Since the whole haptic system is taken into account, this method does not address exclusively haptic algorithms. A benchmark to evaluate the performance of collision detection algorithms using objective parameters and indicators is presented in [7]. Anyway, this benchmarking suite deals only with the performance of collision detection algorithms, without addressing force-feedback.

The evaluation approach presented in this work focuses exclusively on haptic rendering algorithms. The forces computed by the algorithms when moving a dynamic object along a specified set of paths are analyzed. Two types of paths are considered: synthetic paths, which are artificially generated, and real ones, which are recorded from interactive VR simulations. Whereas synthetic paths address typical weaknesses of haptic rendering algorithms, real paths test the algorithms in complex scenarios in which they are usually designed to be used.

The remaining content of this work is organized as follows: Section 2 describes the benchmark scenarios and the analysis method. Section 3 gives examples with a synthetic and a real path using an improved Voxmap-PointShell Algorithm described in the Appendix. And, finally, Section 4 discusses future work after summarizing the main features of the presented benchmark.

2 Description of the Benchmark

This section introduces a benchmark that makes possible evaluating haptic rendering algorithms. The following subsections specify testing scenarios and an analysis strategy for haptic algorithms.

2.1 Scenarios and Paths

Each testing scenario consists of a static and a dynamic object. The dynamic object is moved along a path, which can be classified as synthetic or real.

Synthetic paths aim to objectively evaluate haptic algorithms. Using simple objects collisions can be determined analytically and then the results of the haptic rendering algorithms can be compared to them. These paths are parametrically defined for a pair of geometries. Five synthetic paths are suggested on the left side of Fig. 1 and listed in the following lines:

(a) Translation with constant penetration.
(b) Rotation of a cylinder that collides with an object.
(c) Thin object penetration test.
(d) Rotation of a square pin in a square hole.
(e) Revolution of a “pins” object around the central axis of the holes object.

Paths (a) and (b) analyze collision responses for cases in which forces and/or torques should remain constant, whereas path (c) relates to thin objects, which are a problematic issue in haptic rendering, especially for volume- or penalty-based algorithms. Finally, paths (d) and (e) aim to test the behavior of haptic algorithms for large surface contacts.

In contrast to the synthetic paths, real paths target evaluating the performance of haptic rendering algorithms in real-like applications. These are recorded in a virtual assembly simulation of a complex model. The right side of Fig. 1 contains three parts of a VW Touran which define different assembly scenarios: (f) a coolant tank, (g) a brake and (h) a battery. It is planned gliding plane objects along surfaces, for instance, the battery of scenario (h) against the bodywork of the car. Each one of the real scenarios has specific properties regarding the difficulty in the assembly process and the complexity of the objects.

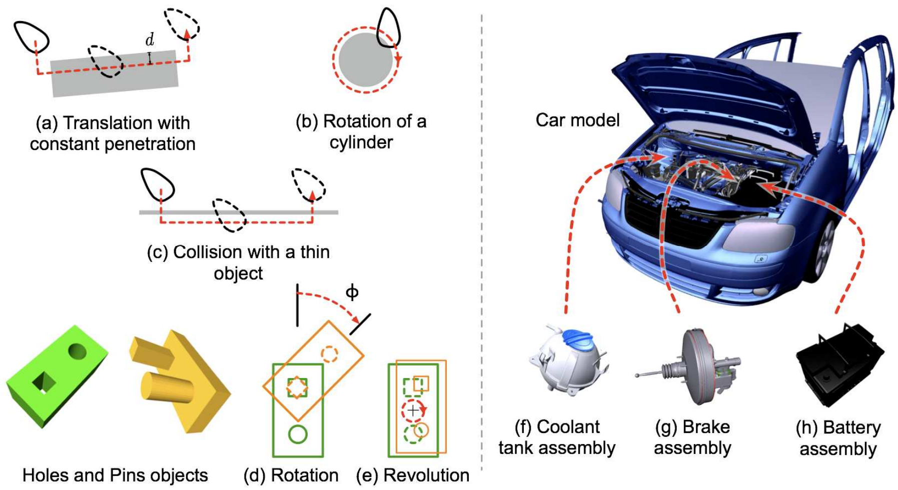


Fig. 1. Schematic picture of the paths in the benchmark. Left: synthetic paths; right: real assembly simulation paths.

2.2 Analysis

The following variables are stored for each sampling step of the trial: (*i*) the relative pose of the dynamic object with respect to the static one, (*ii*) collision forces and (*iii*) torques computed by the tested algorithm, (*iv*) the penetration – depending on the algorithm: depth, volume, etc. – and (*v*) the computation time.

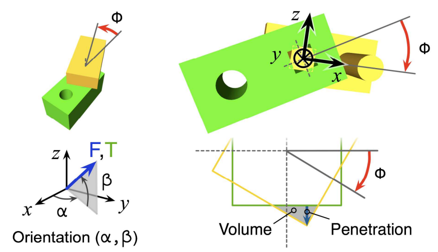
Additionally to verifying that the computation time remains below the recommended limit for haptic rendering of 1 ms, the benchmark analyzes the quality of forces and torques. In the case of the synthetic paths, this is done by measuring the standard deviation of the direction and magnitude curves from the curves of the chosen analytic model.

Given that it is difficult to establish expected models for assembly scenarios, comparative analyses are proposed for real paths.

3 Examples and Discussion

Figures 2 and 3 contain examples of the benchmark using the Voxmap-Pointshell Algorithm (VPS) described in the Appendix. A brief discussion is given below.

In the scenario (a), the computation time varies around its average value of 0.5475 ms, with a maximal peak of 0.5840 ms – see Fig. 2. Only torque is expected in the pin in the direction of the rotation axis ($-y \rightarrow \alpha = 270°$). However, the average α of the simulated torque is 256.70°, and there do appear forces. Expected torque characteristics for different models are shown and the standard deviation of the simulated torques from these curves is also tabulated in the table.



	Linear model	Volume model	Volume root model	Penetration model
Expected F	0	0	0	0
Std. deviation	0.6259	0.6259	0.6259	0.6259
Expected α (F)	-	-	-	-
Std. deviation	-	-	-	-
Expected β (F)	-	-	-	-
Std. deviation	-	-	-	-
Expected T	~ Φ	~ vol.	~ $\sqrt[3]{\text{vol.}}$	~ penetr.
Std. deviation	0.2413	0.2792	0.4128	0.2676
Expected α (T)	270°	270°	270°	270°
Std. deviation	42.65°	42.65°	42.65°	42.65°
Expected β (T)	-	-	-	-
Std. deviation	-	-	-	-

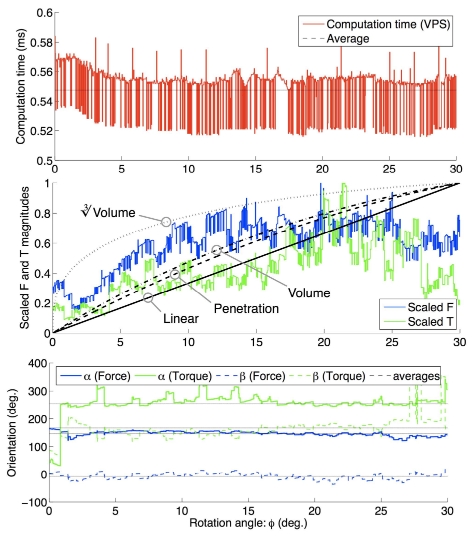


Fig. 2. Analysis of the synthetic scenario (a) using the VPS Algorithm.

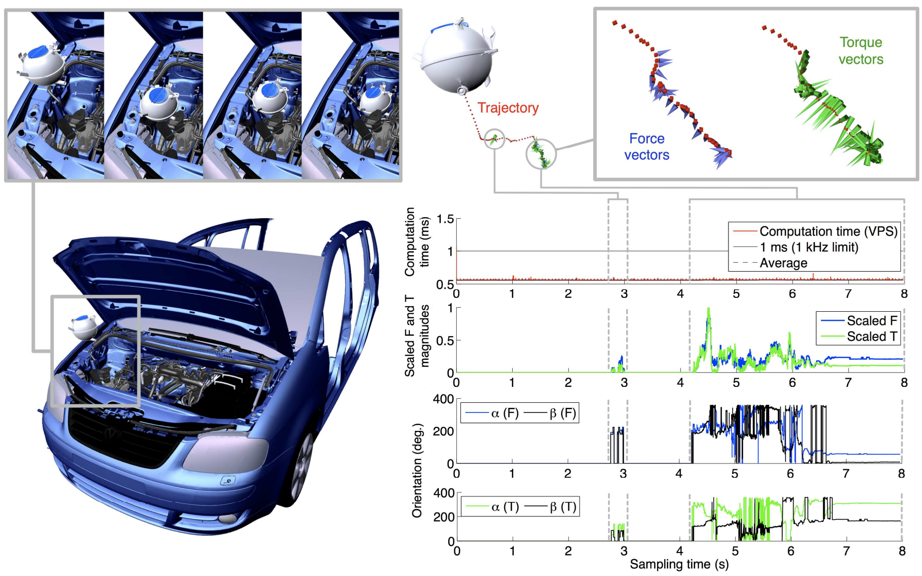


Fig. 3. Analysis of the real scenario (f) using the VPS algorithm.

In the scenario (f), the computation time varies around its average value of 0.5597 ms, with a maximal peak of 1.0630 ms at the beginning – see Fig. 3. On the top right side of the picture force and torque vectors are displayed along the trajectory.

4 Conclusions and Future Work

A benchmarking proposal for haptic rendering algorithms was presented in this work, based on synthetic and real scenarios and paths. The benchmark addresses both known weaknesses of the algorithms as well as their suitability in real complex applications. Given examples illustrate the proposed procedure.

Future work consists in integrating available haptic rendering libraries to perform a thorough comparative study. Additionally, a user-study would help to understand the suitability of algorithms in real applications. The research may lead to the definition of a degree of force quality for haptic algorithms.

Acknowledgments. The VRLab of the Volkswagen AG supported this work in part, and provided the car assembly scenario.

References

1. McNeely, W.A., Puterbaugh, K.D., Troy, J.J.: Six Degree-of-Freedom Haptic Rendering Using Voxelmap Sampling. In: Proc. of SIGGRAPH, pp. 401 – 408 (1999)
2. McNeely, W.A., Puterbaugh, K.D., Troy, J.J.: Voxel-Based 6-DOF Haptic Rendering Improvements. In: Haptics-e. vol. 3, no. 7 (2006)
3. Raymaekers, C., De Boeck, J., Coninx, K.: An Empirical Approach for the Evaluation of Haptic Algorithms. In: Proc. of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 567-568 (2005)
4. Ruffaldi, E., Morris, D., Edmunds, T., Barbagli, F., Kai, D.K.: Standarized Evaluation of Haptic Rendering Systems. In: Proc. of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (2006)
5. Sagardia, M., Hulin, T., Preusche, C., Hirzinger, G.: Improvements of the Voxmap-PointShell Algorithm – Fast generation of Haptic Data-Structures. In: 53rd Internationales Wissenschaftliches Kolloquium, Ilmenau (2008)
6. Trenkel, S., Weller, R., Zachmann, G.: A Benchmarking Suite for Static Collision Detection Algorithms. In: International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), Plzen, Czech Republic (2007)

Appendix: Voxmap-PointShell Algorithm

The haptic rendering algorithm used in Sec. 3 is an improved version [5] of the original Voxmap-PointShell (VPS) Algorithm [1],[2]. This algorithm uses layered voxelmaps – in analogy to pixelmaps, in 3D – to represent the static objects. Dynamic objects are modeled with points on the surface of the object that are assigned inwards pointing normal vectors. Every millisecond the vectors of the colliding points are summed after being weighted by their penetration in the voxelmap. The obtained force value is then sent to the control processor, which computes the appropriate force feedback using the proper stiffness and damping values that correspond to the haptic interface, assuring stability.