A Collision Detection Chip on Reconfigurable Hardware*

Nuzhet Atay John W. Lockwood Burchan Bayazit
Washington University in St. Louis
{atay,lockwood,bayazit}@cse.wustl.edu

Abstract

In this paper, we present an FPGA (Field Programmable Gate Array) based collision detection chip. The chip can be used as a co-processor for a traditional computer or several of them can be utilized to work in parallel to create a very fast collision detection server for real time environments. In our experiments we have seen speeds-up of 36 with respect to a fast Pentium 4 chip. Further improvements are possible by using more advanced collision detection techniques.

Keywords: Collision Detection, FPGA, Hardware

1 Introduction

In this paper, we present a collision detection chip based on FPGAs (Field Programmable Graphics Array). Our design takes advantage of inherent parallelism of collision detection algorithms and checks collisions in parallel. Since we hope to utilize our system in motion planning algorithms, we have defined our collision detection problem as the following: *"given an environment, a dynamic object within this environment, and a set of configurations of the dynamic object, decide if the dynamic object at the given configurations is in collision with the environment"*.

We have both the environment and the dynamic object represented as triangular meshes. Our models are general, non-convex objects. Each configuration of the dynamic object is represented with six parameters (three for the position and three for the orientation). For each configuration our chip finds the transformations of the dynamic object's triangles and do several parallel fast triangle intersection tests against the environment. Since it is highly parallel and use simpler operations, we have used a fast triangle-triangle intersection test to decide the collisions. We observed that at the limit of current FPGA technology, we can get speed-ups of around 36 with respect to a Pentium 4, 3 Ghz. The details of our system can be found in [1].

Although recently a new class of collision detection algorithms [2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13] is proposed to take the advantage of graphics hardware (GPUs), to the best of our knowledge, we are the first researchers implementing a collision detection chip on an FPGA. However, there has been a recent design on Application-Specific Integrated Circuit (ASIC) [14]. In addition to advantage of FPGAs over ASIC, such as more flexible design and low cost, we believe our design is more modular, i.e., the collision detection module can be replaced by any other collision detection algorithm and the further performance gain can be achieved.

2 System

The collision detection chip has four major modules (see Figure 1): (i) *I/O* is responsible from communication between the host computer and the chip, (ii) *memory* stores the object models, (iii) *transformation* transforms the dynamic object, and, (iv) *collision detection* checks the triangular intersection between the dynamic object and the environment.

Figure 2 shows internal design of our chip. Our collision detection chip communicates with the host through serial port. The chip has a local memory to store object representations as well as the configurations of the

*Authors' address: Dept. of Computer Science and Engineering, Washington University, St. Louis, MO 63130.

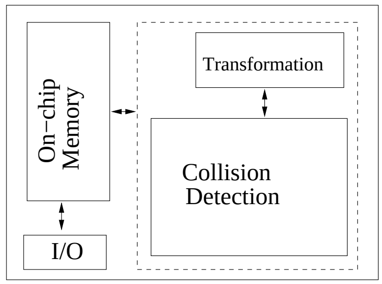

Figure 1: The collision detection chip.

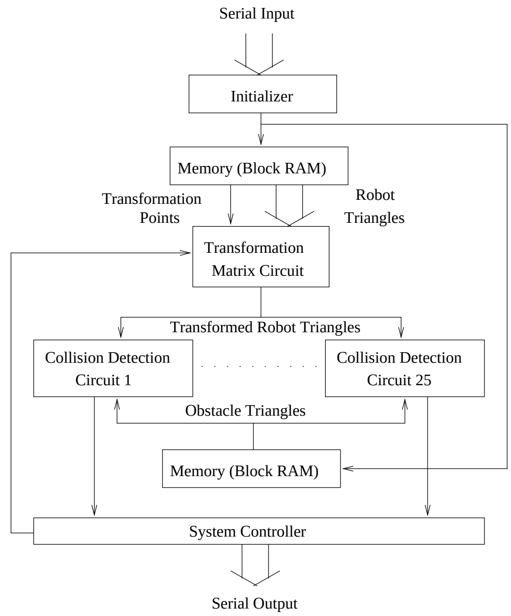

Figure 2: System overview.

dynamic object. A transformation circuit translates the dynamic object to given position and orientation. There are n parallel collision detection circuits each of which get a transformed triangle of dynamic object and check it against the environment's in parallel. The results of collision detection at each configuration is then sent back to the host. Note that this system is very flexible, by replacing the collision detection circuit, we can implement different collision detection algorithms.

Our triangle-triangle intersection test is based on the fast triangle-triangle intersection test described in [11]. Briefly, this algorithm considers three cases: (i) triangles lie in the half-planes of each other, (ii) the triangles are coplanar, (iii) the triangles are not coplanar. We have implemented this algorithm in hardware for the collision detection module.

3 Experiments

Our target FPGA chip is Xilinx Virtex-4 XC4VLX200. This chip allows us to create up to 25 collision detection circuits in parallel and can run our design at the clock rate of 50 MHz. We compared our solution to the sequential execution running on a workstation with Pentium-4 processor at 3 GHz with 1 GB memory. We have both simulation and real hardware results. In our simulation experiments, we have used ModelSim XE II/Starter Edition 5.8c by Mentor Graphics which runs a circuit in real-clock time so timing is accurate, i.e. same results are obtained when design is loaded to FPGA chip.

In order to test our chip, we designed an experiment where a dynamic object is placed at different configurations in an environment. The object is made-up of 12 random triangles. Next we have randomly placed 1600 obstacle triangles in a workspace which has 10 times length, width and height of the bounding box covering the dynamic object. We have checked the collision of the dynamic object at 20 different configurations.

In our experiments, we have executed a functionally similar algorithm in a workstation. In the worst case, each collision check between the moving object and the environment takes 19200 triangular intersection checks. For 20 different configurations of the dynamic object, we have to perform 384,000 triangle comparisons. Next, we run the same experiment using our design in ModelSim. We have tried, 1, 12 and 25 collision detection circuits. Figure 3 summarizes our findings. These results show that even without any parallel collision detection circuit, our chip performs faster than a Pentium 4. Its performance can be explained by the parallelism in individual circuits. The speed-up reaches 36 when we add more collision detection circuits run them concurrently.

To verify that our system is working in real hardware, we have also tested it on Virtex-4 ML401 evaluation board. Unfortunately, this board contains a limited capacity FPGA (Virtex-4 XC4VLX25). Hence, we could test only one collision detection circuit in the real FGPA. We were successfully run the same experiment on this board and found similar results to the simulation.

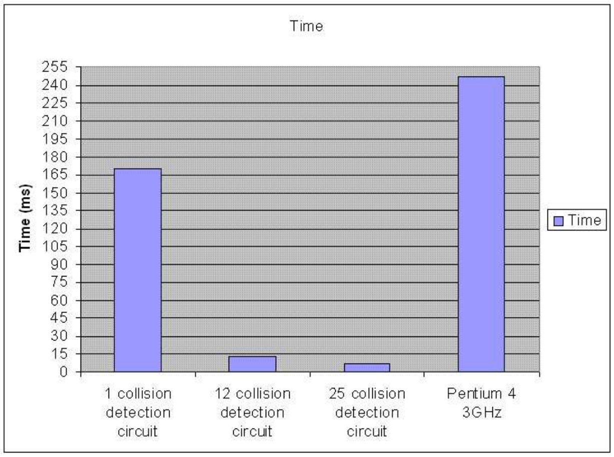

Figure 3: Collision detection chip with 1, 12 and 25 collision detection circuit vs. Pentium 4 for 384K triangle intersection test.

4 Conclusion

We have presented a collision detection chip based on an FPGA. Our chip takes the advantage of inherited parallelism of collision detection algorithms and can compute the collision detection up to 36 times faster than a Pentium-4, 3Ghz CPU. Our future work includes implementing more advanced collision detection algorithms on the FPGA and introducing pipelining to our chip.

References

[1] N. Atay, J. W. Lockwood, and B. Bayazit. A collision detection chip on reconfigurable hardware. Technical Report WUCSE-2005-33, Washington University in St. Louis, St. Louis, MO, 2005.

[2] G. Baciu and S. K. Wong. Image-based techniques in a hybrid collision detector. In *IEEE Trans. on Visualization and Computer Graphics*, 2002.

[3] N. K. Govindaraju, M. C. Lin, and D. Manocha. Fast and reliable collision culling using graphics hardware. In *Virtual Reality Software and Technology (VRST)*, 2004.

[4] N. K. Govindaraju, S. Redon, M. C. Lin, and D. Manocha. Cullide: Interactive collision detection between complex models in large environments using graphics hardware. In *Graphics Hardware*, 2003.

[5] A. Gress and G. Zachman. Object-space interference detection on programmable graphics hardware. In *In SIAM Conf. on Geometric Design and Computing*, 2003.

[6] B. Heidelberger, M. Teschner, and M. Gross. Real-time volumetric intersections of deforming objects. In *Proc. of Vision, Modeling and Visualization*, 2003.

[7] K. Hoff, A. Zaferakis, M. Lin, and D. Manocha. Fast and simple 2d geometric proximity queries using graphics hardware. In *Proc. of ACM Symposium on Interactive 3D Graphics*, pages 145–148, 2001.

[8] P. M. Hubbard. Collision detection for interactive graphics applications. In *IEEE Transactions on Visualization and Computer Graphics*, volume 1, pages 218–230, 1995.

[9] Y. J. Kim, M. A. Otaduy, M. C. Lin, and D. Manocha. Fast penetration depth computation for physically-based animation. In *Proc. of ACM Symposium on Computer Animation*, 2002.

[10] D. Knott and D. K. Pai. Cinder: Collision and interference detection in real-time using graphics hardware. In *Proc. of Graphics Interface*, 2003.

[11] Tomas Moller. A fast triangle-triangle intersection test. *J. Graph. Tools*, 2(2):25–30, 1997.

[12] K. Myszkowski, O. G. Okunev, and T. L. Kunii. Fast collision detection between complex solids using rasterizing graphics hardware. In *The Visual Computer*, volume 11, pages 497–512, 1995.

[13] J. Rossignac, A. Megahed, and B. D. Schneider. Interactive inspection of solids: cross-sections and interferences. In *Proceedings of ACM SIGGRAPH*, 1992.

[14] G. Zachman and G. Knittel. An architecture for hierarchical collision detection. In *The 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2003*, pages 149–156, 2003.