# RTSA: An RRAM-TCAM based In-Memory-Search Accelerator for Sub-100 µs Collision Detection

Jiahao Sun[1], Fangxin Liu[2,*], Yijian Zhang[1], Li Jiang[2,3,*], and Rui Yang[1,2,4,*]

[1]*University of Michigan – Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai, China*
[2]*School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China*
[3]*Shanghai Qi Zhi Institute, Shanghai, China*
[4]*State Key Laboratory of Radio Frequency Heterogeneous Integration, Shanghai Jiao Tong University, Shanghai, China*
[*]*Corresponding authors. Emails: liufangxin@sjtu.edu.cn, ljiang_cs@sjtu.edu.cn, rui.yang@sjtu.edu.cn*

***Abstract*—Collision detection is a highly timing-consuming task in motion planning, which accounts for over 90% of the total calculation time. Previous hardware accelerators can hardly maintain fast computation speed in real time while supporting a large roadmap. In this work, we present RTSA, a novel in-memory-search collision detection accelerator, which achieves an impressive sub-100 µs response time for collision detection in 800 MB scale roadmaps. Such accelerator leverages an in-situ-search-enabled memory architecture, enabling massively parallel search operations. RTSA is powered by ternary content-addressable memories (TCAMs) based on large-scale non-volatile resistive random-access memory (RRAM) arrays. TCAM eliminates the need for extensive data transfer between memory and computing units, leading to significant energy and delay saving. Such accelerator well exceeds the speed requirement for collision detection (<1 ms), making it highly suitable for various applications, including robot motion planning in dynamic environment, manufacturing, and physical simulation.**

## I. Introduction

Motion planning is one of the key technologies in robot motion navigation. Roadmap construction, collision detection, and path search are the three major components in motion planning. Among them, collision detection can consume over 90% of the total motion planning time [1]. To ensure real-time collision detection in a dynamic environment, it is desirable to finish the collision detection in milliseconds [2]. Since a robot may have several degrees of freedom (DOF), probabilistic roadmaps (PRMs) have been used to find possible solutions in a high-dimensional space [3]. However, due to the space complexity of high-DOF robotic arms, CPU-based motion planning can take a few seconds [4], which cannot meet the requirements for real-time collision detection in a dynamic environment. In addition, many applications involving collision detection have limited power supply. Therefore, both high energy efficiency and fast computing speed are important for collision detection.

Various accelerator designs for improving the performance of collision detection have been proposed. High-performance GPUs with parallel processing require hundreds of milliseconds to compute the collision-free path, which are still far from the requirement for real-time processing in a dynamic environment, and they usually consume very high power [5]. An FPGA-based accelerator can take less than a millisecond to finish the collision detection, but because of the limited storage capacity, it processes a roadmap with only 2,500 edges [2]. Therefore, it is not applicable for a large roadmap.

In this manuscript, we introduce a highly efficient RRAM-TCAM-based accelerator for collision detection applications in a large roadmap. In contrast to the existing architectures, the accelerator eliminates the massive data movement between memory and computing units, to perform collision detection with high energy efficiency and delay below 100 µs.

## II. Motivation

The collision detection process can be regarded as a comparison between the environment and all the edges within the same volume. Edges in a roadmap refer to the possible moving trajectories of robots, whose number is very large (up to 100,000). Given such immense volume, the deployment of large-capacity, on-chip memory array is essential for storing all the information of the edges. Meanwhile, massive data transfer between the memory and computing units consumes a lot of energy and delay due to the large data capacity [6]. Therefore, an architecture that can accelerate the comparison functionality in a highly parallel scheme and can directly process large amounts of data inside the memory is highly desirable for decreasing the energy and delay.

Application-specific integrated circuit (ASIC) has been proposed to accelerate the collision detection, but the speed and energy efficiency are still limited [6,7]. In-memory computing avoids the frequent data transfer between the memory and computing units, which can achieve high energy efficiency [8]. Processing-in-memory hardware accelerator for collision detection has been demonstrated, which can store and process the information on a 16 GB DRAM [9]. We summarize the existing ASIC-based designs for collision detection in Table I. From this table, we find that current accelerators still suffer from high computing latency and low energy efficiency.

To further improve the speed and energy efficiency for collision detection, it is imperative to optimize both memory devices and architecture. Resistive random-access memory (RRAM) is a nonvolatile memory with high potential for in-memory computing due to its high density and low power consumption [10]. Ternary content-addressable memories (TCAMs) can perform parallel in-memory searches. Notably, RRAM-based TCAMs offer advantages of reduced size, fast search speed, and high energy efficiency [11]. In this work, we design the first RRAM-TCAM based in-memory-Search Accelerator (RTSA). RTSA *leverages the TCAM structure to implement the collision detection logic, and allows fully parallel search operation with an octree representation, enabling fast speed (below 100 µs) and high energy efficiency for collision detection of a large roadmap with 100,000 edges.*

979-8-3503-4859-0/DATE24/© 2024 EDAA

TABLE I. ASIC ACCELERATORS FOR COLLISION DETECTION

| | MICRO 16 [2] | DADU-P [6] | DADU-CD [9] | **This work** |
|---|---|---|---|---|
| Storage cell | SRAM | SRAM | DRAM | **RRAM** |
| Energy | High | High | Medium | **Low** |
| Speed | Fast | High | Medium | **Fast** |
| Supporting Large Roadmaps? | No | Yes | Yes | **Yes** |

TABLE III. TWO-BIT ENCODING OF AN ELEMENT

| **Two-Bit Encoding** | **Environment (SRAM)** | **Edge (RRAM)** |
|---|---|---|
| Totally Occupied | 1/1 | 1/1 (LRS/LRS) |
| Partially Occupied | 1/0 | 1/0 (LRS/HRS) |
| Free Space | 0/0 | 0/0 (HRS/HRS) |

## III. RTSA ARCHITECTURE

We show the octree encoding scheme in Fig. 1. Octrees recursively divide the whole volume cube into eight small even cubes until reaching fully occupied or free cubes. Then, with proper digital encoding of the space, the 3D models can be stored in memory arrays. In our design, to fully leverage the parallelism of TCAMs, all the sub-cubes are divided into the finest resolution (6$^{th}$ level). Each finest-resolution sub-cube is named an element. Two bits are used to encode three states of an element: occupied, partially occupied, and free space, for both the environment and the edge (Table II).

After using two bits to encode an element, we perform TCAM search operation to determine whether collision happens by detecting the ML voltage. As shown in Fig. 2, the edge information of an element is stored in a two-transistor-two-resistor (2T2R) TCAM cell, while data on the corresponding search lines (SL/SLR) are the 2-bit environment encoding. All the match line (ML) voltages are pre-charged to high. If the ML shows high resistance, the ML voltage remains high or slowly discharge. If there are low-resistance TCAM cells, the corresponding ML voltage will be pulled down more quickly. ML sense amplifiers (MLSAs) can capture the output of all the rows as the collision detection signal, which is used for further processing. In our design, ML kept high indicates no collision, while ML pulled down implies collision.

The dynamic obstacles in the environment require the encoding of environment to be changed in real time. Although RRAM costs less area and energy compared with SRAM, it has relatively limited endurance [10]. Therefore, we use SRAM cells to store the information of the environment in MA1 (Fig. 2), which will be frequently modified. To utilize the fast read access of RRAM and massive parallelism of TCAM, MA2 is used to store all the edges of a roadmap with 6$^{th}$-level octree encoding, which does not need frequent update. If there exists element collision, the corresponding ML voltage will be pulled down. Only when all the elements are collision-free, ML voltage keeps high. That's consistent with the collision detection logic. The edge is determined to have collision with the whole environment even if a tiny collision happens for an element. Multiple edges are compared with the environment at the same time thanks to the high parallelism of the TCAM, which greatly saves the computation time.

## IV. CONCLUSIONS

In summary, we propose the RTSA, an efficient hardware accelerator for collision detection. We use TCAMs to perform massively parallel in-memory search, which accelerates the collision detection. In contrast to previous collision detection accelerators that suffer from high power and long computation time under a large roadmap, the RTSA can achieve collision detection within 100 μs for a large roadmap with 100,000 edges. With a set of innovative encoding and circuit architecture design, the RTSA accelerator outperforms other ASIC designs in terms of detection speed, energy consumption, and area overhead, and far exceeds those metrics for CPUs and GPUs.

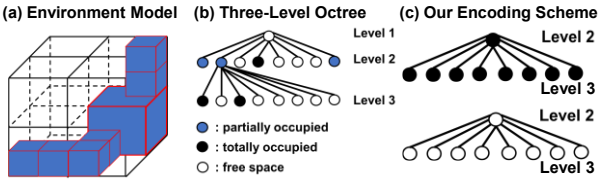


Fig. 1. Illustration of octree representation.

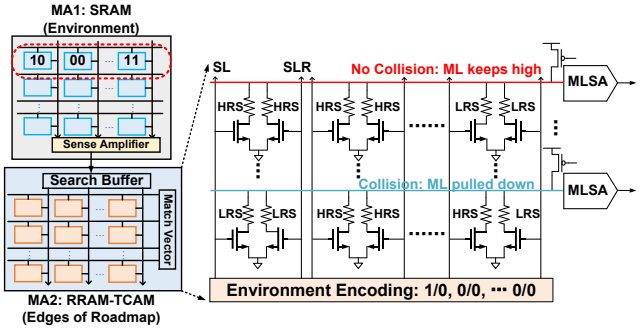


Fig. 1. Workflow mapping scheme and configuration of a TCAM.


## ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China (NSFC) (Grants 92364107, 62104140, 62250073), Science and Technology Commission of Shanghai Municipality (STCSM) Rising-Star Program (Grant 23QA1405300), Natural Science Foundation of Chongqing (CSTB2022NSCQ-MSX1095), and Lingang Laboratory Open Research Fund (Grant LG-QS-202202-11).


## REFERENCES


[1] J. Bialkowski et al., "Massively parallelizing the RRT and the RRT*," in *IROS*, 2011, pp. 3513–3518.

[2] S. Murray et al., "The microarchitecture of a real-time robot motion planning accelerator," in *MICRO*, 2016, pp. 1–12.

[3] L.E. Kavraki et al., "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *TRA*, 1996, vol. 12, pp. 566–580.

[4] K. Hauser, "Lazy collision checking in asymptotically-optimal motion planning," in *ICRA*, 2015, pp. 2951–2957.

[5] A. Hermann et al., "Unified GPU voxel collision detection for mobile manipulation planning," in *IROS*, 2014, pp. 4154–4160.

[6] S. Lian et al., "Dadu-P: A scalable accelerator for robot motion planning in a dynamic environment," in *DAC*, 2018, pp. 1–6.

[7] S. Murray et al., "A programmable architecture for robot motion planning acceleration," in *ASAP*, 2019, pp. 185–188.

[8] S. Yu et al., "Compute-in-memory with emerging nonvolatile-memories: Challenges and prospects." in *CICC*, 2020, pp. 1–4.

[9] Y. Yang et al., "Dadu-CD: Fast and efficient processing-in-memory accelerator for collision detection." in *DAC*, 2020, pp. 1–6.

[10] Z. Wang et al., "Resistive switching materials for information processing." *Nature Reviews Materials*, 2020, vol. 5, pp. 173–195.

[11] R. Yang et al., "Ternary content-addressable memory with $MoS_2$ transistors for massively parallel data search." *Nature Electronics,* 2019, vol. 2, pp. 108–114.