Optimized Kinetic Data Structures Based Algorithms For Collision Detection In Robot Motion Planning

Neha Arora and Deepak Garg
Thapar University, Patiala

article

Abstract

This paper discusses about the collision detection algorithms used in Robotics. Classical methods were based on bounding volumes which were not suitable for real environment. Hence, new algorithms and technologies are implemented to improve the "Motion planning" process. Also a few optimizations have been made in classical methods like optimized spatial hashing technique and greedy pseudo triangulations. Latest projects and softwares used for this purpose have also been discussed. HAPTICS and GENETIC algorithms play an important role in this area. Moreover effort has been made to accelerate the performance of earlier methods using high speed graphic hardware and pipelines. And also a further research has been made on SEMIDEFINITE PROGRAMMING (A new form of optimization technique).

1. Introduction

Collision detection is a tool whose performance is very important in order to achieve efficiency in many robotics applications primarily "Motion Planning" which has arisen in context of obstacle avoidance and path planning. It has been listed as the major bottleneck towards real time virtual environment simulations of human intervention. Visual sensors have evolved but they have yet not been able to extract wealth information in visual scenes.

In order for Humanoid robots to become practical they must be able to operate safely. Even self collisions are harmful. These occur when one or more links of a Bot collide which results into damage. Hence, ageometric approach is followed.

A humanoid consists of a tree of connected links which can be viewed as a set of five serial chain manipulators (2 arms, 2 legs, and 1 neck head chain) all attached to a free floating trunk. Hypothetically, joint limits prevent collision between a given link and a parent link. Example Protocol is ARMAR with 23 degrees of freedom.

The main goal here is to develop optimized and efficient methods that do not require recomputations at fixed time intervals but detect collisions exactly when they happen. For this purpose we use "Kinetic Data Structures".

2. Kinetic Data Structures

The idea originates from a fact that computational geometry structures are built using predicates - functions on quantities defining geometric input which return a discrete set of values such as a convex hull of closest pair of points. While modeling motion, the data must be sampled and combinatorial structure of the data must be updated periodically. Dynamic data structures do not suffice this condition. Hence, a KDS framework was introduced. These are based on assertions about the world called certificates which are geometric relations among few features of moving objects. As long as certificates remain valid, this computation can be used to get value of relevant system attributes. At each certificate failure, certificate set being maintained, must be repaired, is the and essence of good kinetic structure. The certificate failure time must be either detected or predicted. The predicted failure times of kinetic structures are placed in an event queue i.e priority queue based on their time of expiration and the system clock can be advanced until either one of the objects changes its motion plan. The following four measures are used to describe quality of KDS as suggested by Guibas [7] :

Respon siveness	Effic- iency	Comp actness	Loc ality
O(nlogn)	O(logn)	O(n)	O(logn)

Table 1 Complexities

3. Application Of KDS

A KDS provides with event based algorithms. As we know, the roots of collision detection lies in computational geometry where the problem is to report intersections among static objects. However, in motion planning of Robots the problem is more difficult as they are moving. The practical approach is to study collision among polyhedral objects simulating robots in real world. Earlier moving polyhedral objects solved problem instances from scratch at every time step. These algorithms required running time that was quadratic in nature. They were too slow as they neglected the fact that objects in motion discretize the time axis and update the structures based on relative position. If all objects move continuously then in general their configuration does not change significantly between time steps. Time discretization can exploit spatial and temporal coherence. Temporal Coherence is necessary to detect precisely those points in time where there is an actual change in structure. Hence there are certain discrete moments in time when combinatorial structure of object changes and certificates constitute a property of interest as they assert facts like" point c lies on left of directed line through a and b". This property leads to development of constant time algorithms. The most efficient method uses "Pseudo Dynamic or two phase detection algorithms". Each possible state or pose of a robot is a point, and a path is a curve in space. Hence, it requires expensive precomputations.

4 Pseudo-Dynamic Collision Detection Algorithms(Probabilistic methods):
The methods use point cloud data as inputs. Firstly, the collision query is made between two point objects and then for polygonal objects. It is used to uncover uncertainity modeled as

$$x_i | S \sim p(x_i | S) = \int p(x_i | S) p(x_i | x_i'; \Sigma_i) dx_i$$

The outcome of which is a probability $P_{c1,c2}$ which estimates whether two point clouds are in collision. P is the separating surface. However, separating surface is not the sufficient condition for point data as comapared to polygons. Hence, it makes it difficult to use point cloud data sets. Though it helped in avoiding noise interference.

4.1. Broad Phase
The function of these algorithms is to quickly remove most of the intersection pairs from consideration. using "Bounding volumes" such as cuboids, spheres, polyhedrals. Boxes can be used to bound the region of space occupied by an object at a specific time interval. Consider a bounding box Bi, that encloses object i over time interval [to,to+t].Let ri be the maximum distance of object from its centre of mass. Bi is found by noting position of centre of mass at time to and to+t and possibly the apex of trajectory. A recent advancement over it is ORIENTED BOUNDED BOX as it yields the better outer approximation of the objects which lead the pathway to hierarchical structures known as "Strip trees". Commonly used data structure is R Trees(Rectangle trees).They provide exceptional improvement over K-d Trees(Binary space partition) because they partition obstacles rather than space and have potential for efficiently performing collision detection queries. Objects pairs with intersecting bounding boxes are passed to narrow phase collision detection.

4.1.1 Minimum Distance Detection Algorithm:
Quillan's Greedy Combine Method(Sphere tree hierarchy construction):In this approach an initial grid of leaf spheres with fixed size is blacketed over the surface of every polygon. Nearby spheres are grouped together to form a hierarchy bottom up. The radii of larger are calculated to entirely contain smaller ones

4.1.2.1 Intersection Detection Algorithm
4.1.3 Optimised Spatial Hashing Technique(Grid based Method):
Spatial hashing is a process by which a 3D or 2D domain space is projected into 1D hash table. Imagine a grid of 100 by 100 pixels and each cell is 25 by 25. Each cell is a bucket of game objects and a unique hash id. Any object in bucket [3] cannot collide with object in bucket [9]. This reduces the time we need to cycle the nearby objects. But what if it crosses a line and exists in more than one bucket. To resolve this, imagine a box around object and calculate hash id for each corner of object. Figure 1 depicts the Simulation done using MATLAB.

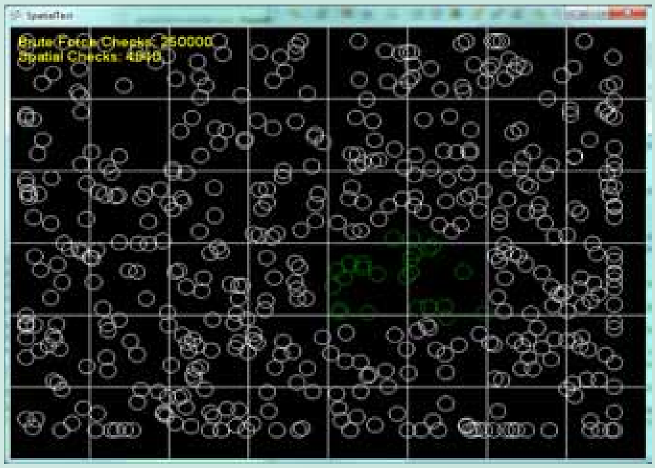


Figure 1. Spatial hashing technique

4.1.3 CoordinateSorting:
It is based on sorting the coordinates of bounding boxes along each of the three coordinate axes. The minimum and maximum of each of x-axis aligned are maintained in sorted list. Same is done for y and z coordinates. Two Boxes overlap if and only if their coordinates overlap in each of three directions.

Coherence is exploited when updating previous sorted lists. Coordinate sorting is based on dimension reduction. The coordinates may be clustered even when original boxes are not; the clustering becomes worst in higher dimesions. One way of handling clustering problems is to perform less drastic dimension reduction, projecting 3-Dimensional boxes into 2-Dimensional rectangles in plane.
Hashing does not suffer from clustering, however it does not cull as many intersection pairs as coordinate sorting. The system implemented here is the “RAPID” system implemented by Gottschalk, Lin and Manocha [3]. Vanecek used another approach i.e Modified back face culling technique. It removes all polygons whose normals face away from moving polygon.

4.2 Narrow Phase (space time bound algorithms):
Narrow phase collision detection analyze the pair of objects that are not easily culled by broad phase detector. These are based on fourth dimensional geometry. They employ more refined but slower algorithms to determine whether objects are penetrating or disjoint. When penetration is detected, a binary search method can be used to localize the time of transition from disjoint stat to the penetration state, within a desired tolerance

4.2.1 Lin Canny Closest Feature Algorithm:
It Operates on rigid convex polyhedra. It is an extremely fast method for tracking the closest features. The fundamental concept in this algorithm is of Vornoi region. The polygon has 8 features: four vertices and four edges. For each feature F, the set of points closer to F than to any other feature of polygon is called vornoi region of F, denoted by V(F). The shapes are easily deduced for polygon objects. From each vertex, extend two rays outward from the polygon, each perpendicular to one of the edges. The Vornoi region of a vertex is an infinite cone while that of an edge is an infinite rectangle. Collectively, it partitions the space outside polygon. The system used here is I-Collide. Recent improvement over it is Q-Collide system and V-Clip.

4.3 Analysis:
In case of two convex polytopes, the distance can be determined in O(log^2n)time, where n is the number of vertices, which takes O(n) time and space to construct. Generalizations have been done to provide sub quadratic time in worst case for a collection of polyhedra.

5. Latest Trends and Optimisations:
5.1 Greedy Pseudo Triangulations (Mesh-based method):
Pseudotriangle is a simply connected subset of the plane that lies between any three mutually tangent convex sets. A pseudo triangulation(Delaunay) is a partition of region of plane into pseudo triangles.

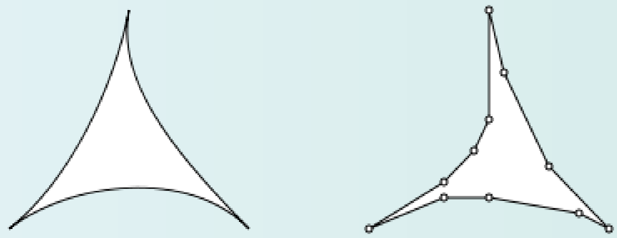


Figure 2. Pseudo triangulations

A pseudo triangulation is denoted by T(S) and each face is a pseudo triangle which lies in interior S pseudo triangles can flex as the objects move and have sufficiently simple shapes and therefore the self intersections are easy to detect. For a Low degree algeabraic motion, pseudo triangulation changes about n^2 times. Therfore, a greedy vertical pseudo triangulation is used which considers a minimum number of cells K. For any linear ordering on the free tangents, we can build a corresponding greedy triangulation, first we sort the list and then scan the set of tangents. At each step pick a tangent segment s and check if it intersects with any tangents in Sorted sequence S. Discard if it does not. The number of certificates needed to maintain correctness is propotional to minimum link subdivision. This method is particularly useful for continuously deforming objects. Therefore it is possible to obtain O(logn) query time with $O(n^4)$ space

Robots	dof	triangles
2 robots	12	8452
4 robots	24	16644
3 robots	18	14048
2 robots	12	9044
4 robots	24	14048
3 robots	18	30462
6 robots	36	41244

5.2 Dynamic Collision Detection Methods:
Pseudo dynamic collision methods suffer from temporal aliasing. However, Dynamic collision detection methods considers time and motion information in order to give conservative results without missing collisions. The dynamic collision does not suffer from temporal aliasing
The threshold where the accuracy of these algorithms depend is how far objects move per sample interval. This approach uses three stage pipeline:

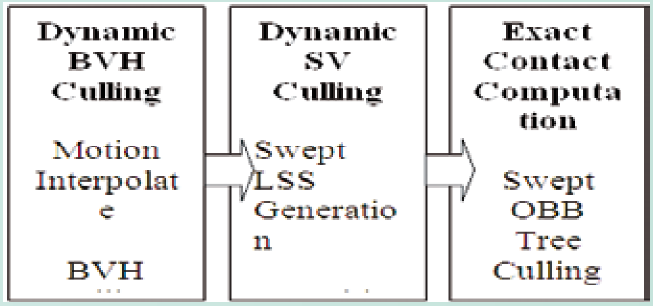


Figure 3. Pipelining

5.2.2 Graphics Hardware Method(Fast dynamic method)
Acceleration in hardware can be achieved through FPGA which acts as a co processor. It takes advantage of inherent parallism of collision detection algorithms

Due to recent advances in performance of graphic processors as well as increased support for programmability and lack of preprocessing, these methods are in great demand.

Figure 4. V-REP(Virtual Robot Experimentation Platform): Simulation using CMLABS

5.3. Haptic Interface
Haptic technology is a tactile feedback technology that takes advantage of users sense of touch by applying forces, vibrations or motions to the user. Recently, haptic devices have been used in the field of robotics. Haptic

interface is used to teleoperate a mobile robot to explore polygonal environments. It improves navigation time and operator perception. Robot motion is usually controlled by the camera mounted on robot. However, they require much of operator attention whereas haptic devices give additional sense of "feeling" the robot workspace. The human operator can simultaneously select two commands active and guarded motion type. Each receives a force feedback independently and force origin becomes clear. eM - Virtual Desktop is one of the software used for this purpose which provides powerful solutions. Prof. Gabriel Zachmann has coined a term Haptesha - a project being sponsored by grant avilus.

5.4. Evolutionary Algorithm:

It Is a multi objective optimization problem. It considers three objectives:

- Minimum Energy Consumption
- Stability
- Walking speed

Non-dominated sorted genetic algorithm is used to generate the generate the pare to set of robot gaits

5.5. Hybrid Genetic Algorithm

Implementation:

Genetic algorithm is particularly easy to implement and promises substantial gains in performance. It is used to search for optimal path from a population of feasible paths. It eliminates the problem of free space connectivity problem. The algorithm is further enhanced using traversibility vectors. The basic structure is given as:

1. Initialize population
2. Evaluate initial population
3. While not termination-condition
4. Assign fitness values to entire population
5. Select individuals for crossover
6. Crossover
7. Mutation
8. Offspring evaluation
9. Offspring reinsertion

6. Conclusion and Future Work:

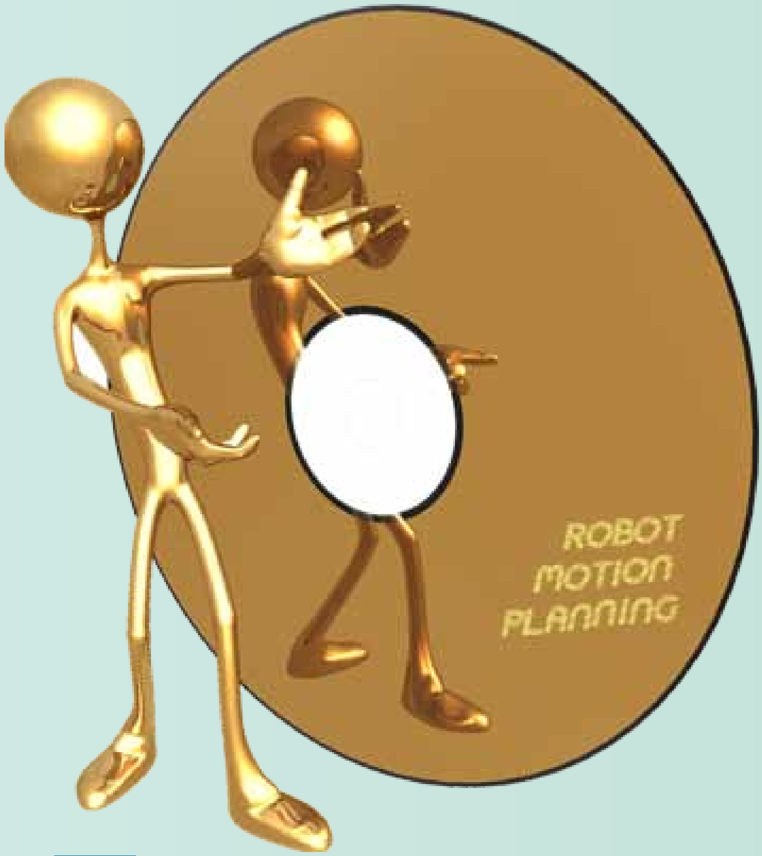
Most of the prior work on collision detection assumes an exact geometric description of the objects in the scene, typically represented as a polygon mesh. However, these methods may not work well for robots operating in real-world environments, where only partial observations of the environment are possible based on robot sensors. Hence, new methods and technology was implemented which increases the efficiency and time complexity reduces to exponential time. Also, it overcomes errors such as discretisation and noise errors. All methods are efficient, however, haptics needs much more exploration in this field.

There are many avenues for future work. The above analysis can be applied to various potential applications like virtual reality based training and dynamic system simulation. In fact, optimizations are being made through the use of new programming language "Semidefinite programming" which is commonly used in Gaming. It provides a virtual framework for optimization of complex systems. Hence, it would be helpful in resolving issues of contact points and penetration which are still under the veil.

REFERENCES

1. Abdel -Malek K., Othman S.: Multiple sweeping using the denavit-hartenberg representation method. Computer-Aided Design 31 (1999), 567–583.
2. Cameron S.: Collision detection by four-dimensional intersection testing. Proceedings of International Conference

on Robotics and Automation (1990), 291–302.
3. Larsen E., Gottschalk S., Lin M., Manocha D.: Distance queries with rectangular swept sphere volumes. Proc. of IEEE Int. Conference on Robotics and Automation(2000).
4. K.-H. Bae, D. Belton, and D. D. Lichti, “A closed-form expression of the positional uncertainty for 3d point clouds,” Transactions on Pattern Analysis and Machine Intelligence, vol. 31, pp. 577–590, 2009.
5. David Bara_. Dynamic Simulation of Non-Penetrating Rigid Bodies. PhD thesis, Department of Computer Science, Cornell University, March 1992.
6. Stephen Cameron. Enhancing GJK: Computing minimum penetration distances between con- vex polyhedra. In Proceedings of International Conference on Robotics and Automation. IEEE, April 1997.
7. J. Basch, L. Guibas, and J. Hershberger. Data structures for mobile data. Proc. 8th ACM- SIAM Sympos. Discrete Algorithms, pp. 747{756. 1997.
8. [Cameron97] Cameron, S., A comparison of two fast algorithms for computing the distance between convex polyhedra, IEEE Transactions on Robotics and Automation, 13(6):915--920, December 1997.
9. Dobkin, D. P. and Kirkpatrick, D. G., Determining the separation of preprocessed polyhedra -- A unified approach, in the Proceedings of the 17th Internat. Colloq. Automata Lang. Program., V. 443 Computer Sciences, pages 400--413. Springer-Verlag,1990.



AUTHORS

Neha Arora is a PG student at Thapar University, Patiala. She is a scholarship holder for Android Curriculum.

Dr Deepak Garg is Chair, ACM SIGACT North India and Secretary IEEE Computer Society, Delhi Section.

He has more than 83 publications in International Conferences and Journals to his credit.

His research interests are algorithms, Complexity theory and Theoretical Computer Science.