



KTH Industriell teknik
och management

Examensarbete MMK [TRITA-ITM-EX 2019:394]

Raskt-utforskande Slumpmässiga Träd med N:tegradsytor

Yared Efrem Afework

Godkänt 2019-06-03	Examinator DeJiu Chen	Handledare Pouya Mahdavi pour Vahdati
	Uppdragsgivare N/A	Kontaktperson Rickard Svensson

Sammanfattning

Nyckelord: Superquadrics, RRT, GJK, Kollisionsdetektering

Detta examensarbete undersökte fördelarna och nackdelarna med att använda n:tegradsytor (NY) för att utföra kollisionsdetektion i algoritmen Raskt-utforskande Slumpmässiga Träd (RST). RST används typiskt för planeringen av system med relativt många frihetsgrader. En etablerad metod för kollisionsdetektion, Gilbert-Johnson-Keerthi-algoritmen (GJK), implementerades även i jämförelsesyfte. Då GJK-algoritmens största flaskhals ligger i kollisionsdetektionen är detta ett intressant ämne att efterforska.

Den NY-baserade kollisionsdetektionsmetoden jämfördes med den GJK-baserade metoden både kvantitativt och kvalitativt. Kvalitativt jämfördes beräkningshastighet och minnesåtagande, medan de kvalitativt jämfördes i deras förmåga att representera godtyckliga geometriska former. På ett högre plan diskuterades det även hur lämpliga de är för att modellera en robotarm med 6 stycken frihetsgrader. RST-algoritmen jämfördes även med en annan planeringsalgoritm, A*. Framförallt fokuserade diskussionen kring planering av system med relativt många frihetsgrader.

I det fall inga kollisioner fanns presterade GJK-algoritmen ungefär lika bra som NY algoritmerna i att fastslå detta, utifrån beräkningshastighet. Men när det kom till att upptäcka existerande kollisioner presterade GJK-algoritmen sämre. Minnesmässigt använder GJK-algoritmen mer minne, då den kräver att båda objekten är explicitrepresenterade (dvs, som ett punktmoln), medan man med en NY-metod endast behöver representera ena objektet explicit och den andra implicit.

Gällande förmågan att representera godtyckliga geometriska former är NY-baserade metoder bättre. Till skillnad från GJK som är begränsad till konvexa mängder kan NY uppta icke-konvexa former, exempelvis flottirmunkformade supertoroider. En metod som använder GJK-algoritmen skulle behöva bygga upp icke-konvexa former med flera mindre konvexa komponenter. NY-metoden är således bättre för att modellera robotarmar med 6 frihetsgrader. Det är dock i praktiken lättare att implementera GJK-metoden då den endast kräver punktmoln, medan NY kräver parametrar som måste bestämmas eller finjusteras.

RST-algoritmen implementerades sist, utformad så att kollisionsdetektionsmetoderna är utbytbara. Det var dock inte möjligt att dra slutsatser utifrån det testdata som erhöles, ty RST-algoritmens slumpmässiga karaktär. RST-algoritmen kan ses som en multiplikator som endast förstör de inneboende egenskaperna hos kollisionsdetektionsmetoderna.



KTH Industrial Engineering
and Management

M.Sc. Thesis MMK [TRITA-ITM-EX 2019:394]

Superquadrics-augmented Rapidly-exploring Random Trees

Yared Efrem Afework

Approved 2019-06-03	Examiner DeJiu Chen	Supervisor Pouya Mahdavi pour Vahdati
	Commissioner N/A	Contact person Rickard Svensson

Abstract

Keywords: Superquadrics, RRT, GJK, Collision Detection

This thesis work investigated the advantages and disadvantages of using superquadrics (SQ) to do the collision-checking part of the Rapidly-exploring Random Trees (RRT) motion planning algorithm for higher Degree of Freedom (DoF) motion planning, comparing it with an established proximity querying method known as the Gilbert-Johnson-Keerthi (GJK) algorithm. In the RRT algorithm, collision detection is the main bottleneck, making this topic interesting to research.

The SQ-based collision detection method was compared to the GJK algorithm both qualitatively and quantitatively, comparing computational speed, memory requirements, as well as the ability to handle arbitrary shapes. Furthermore, how appropriate they are in modelling a 6 DoF arm was analyzed. A qualitative comparison between the RRT algorithm and the A* algorithm was also provided, comparing their suitability for searching in higher dimensional spaces.

When there were no collisions the SQ-based algorithms performed roughly at parity with the GJK algorithm in terms of computational speed. However, when a collision had occurred, the SQ-based algorithms were able to return a positive faster than the GJK algorithm, outperforming it. From a memory standpoint the SQ-based algorithms required less memory as they could leverage the explicit and implicit representations of the SQ objects, whereas the GJK algorithm requires both objects being checked for collision to be explicitly represented as convex sets of points.

Regarding handling arbitrary shapes, the SQ-based algorithms have an advantage in that they can allow for certain non-convex shapes to be. Conversely, the GJK algorithm is limited to convex shapes. The GJK algorithm would thus require more geometric primitives to accurately capture the same non-convex shape. Thus, it can be concluded that the SQ-based method is more suitable for modelling a 6 DoF arm. However, a GJK-based collision detection module would in most cases be a lot more straightforward than the alternative to set up, as it is very simple to collect a set of points.

Finally, both collision detection method types were implemented with the RRT algorithm. Due to the inherently random nature of the RRT algorithm the results of this set of tests could not be used to make any further conclusions beyond showing that it is possible to combine the SQ-based algorithm with the RRT algorithm. Instead, one should see the RRT algorithm as a multiplicative factor applied to the inherent properties of the previously examined collision detection methods.