Towards Numerical Computational Geometry

Chee K. Yap

Courant Institute, NYU

July 30, 2014

Beihang University – NODI Center

ABSTRACT

Exact Geometric Computation (EGC) has provided a powerful paradigm for correct implementation of geometric algorithms for two decades. It is the most successful technique in Computational Geometry, and has led to successful libraries (CGAL, LEDA, Core Library) and numerous practical robust algorithms. We have fairly good understanding of the key techniques and what can be achieved under this paradigm. We review some strong reasons to extend this paradigm:
* The existence of exact solutions may not be known (e.g., analytic roots)
* Exact methods may be too inefficient (e.g., shortest path amidst discs)
* Exact algorithms may not be known (e.g., Voronoi diagram of polyhedra)
* Exactness is inappropriate in many practical problems (e.g., robot motion planning)
To get around this, we can use the well-known idea of computing up to some epsilon resolution. But we will point out some problems with the standard use of the epsilon parameter. Our proposed solution is surprisingly new and opens up new classes of algorithms. The correct viewpoint is to develop ”purely numerical” approaches. It is an exciting new direction for Computational Geometry: it will allow us to solve old problems more efficiently, and allow us to solve previously untouchable problems. We will illustrate with examples from several of our recent papers, including analytic root isolation, isotropic approximation of surfaces, robot motion planning.

Overview of Talk

A. The Exact Computation Paradigm
B. Beyond Exactness?
C. Towards Numerical CG
D. Conclusion

Next...

I. The Exact Computation Paradigm

II. Beyond Exactness?

III. Towards Numerical Computational Geometry?

IV. Issues of Subdivision

IV. Conclusion

I. The Exact Computation Paradigm

Exact Geometric Computation

(A) Exact Geometric Computation (EGC)

- Most successful general paradigm in the last 20 years

(B) The EGC Prescription:

(C) How to Implement EGC?

Exact Geometric Computation

(A) Exact Geometric Computation (EGC)

- Most successful general paradigm in the last 20 years

(B) The EGC Prescription:

- 1. Geometric algorithms branch by evaluating predicates

(C) How to Implement EGC?

Exact Geometric Computation

(A) Exact Geometric Computation (EGC)

- Most successful general paradigm in the last 20 years

(B) The EGC Prescription:

- 2. These branches determine the geometry

(C) How to Implement EGC?

Exact Geometric Computation

(A) Exact Geometric Computation (EGC)

- Most successful general paradigm in the last 20 years

(B) The EGC Prescription:

- 3. Just ensure that each branch are correctly taken.

(C) How to Implement EGC?

Exact Geometric Computation

(A) Exact Geometric Computation (EGC)

- Most successful general paradigm in the last 20 years

(B) The EGC Prescription:

- THAT IS IT!

(C) How to Implement EGC?

Exact Geometric Computation

(A) Exact Geometric Computation (EGC)

- Most successful general paradigm in the last 20 years

(B) The EGC Prescription:

- THAT IS IT!

(C) How to Implement EGC?

- 1. Must use iterative numerical approximation

Exact Geometric Computation

(A) Exact Geometric Computation (EGC)

- Most successful general paradigm in the last 20 years

(B) The EGC Prescription:

- THAT IS IT!

(C) How to Implement EGC?

- 2. Need zero bounds

Exact Geometric Computation

(A) Exact Geometric Computation (EGC)

- Most successful general paradigm in the last 20 years

(B) The EGC Prescription:

- THAT IS IT!

(C) How to Implement EGC?

- 3. Filters makes EGC practical

Exact Geometric Computation

(A) Exact Geometric Computation (EGC)

- Most successful general paradigm in the last 20 years

(B) The EGC Prescription:

- THAT IS IT!

(C) How to Implement EGC?

- 4. INSIGHT: it is possible to automate all these

Exact Geometric Computation

(A) Exact Geometric Computation (EGC)
- Most successful general paradigm in the last 20 years

(B) The EGC Prescription:
- THAT IS IT!

(C) How to Implement EGC?
- Hence, Core Library, LEDA, CGAL,...

Next...

II. Beyond Exactness?

The Trouble with Exactness, I.

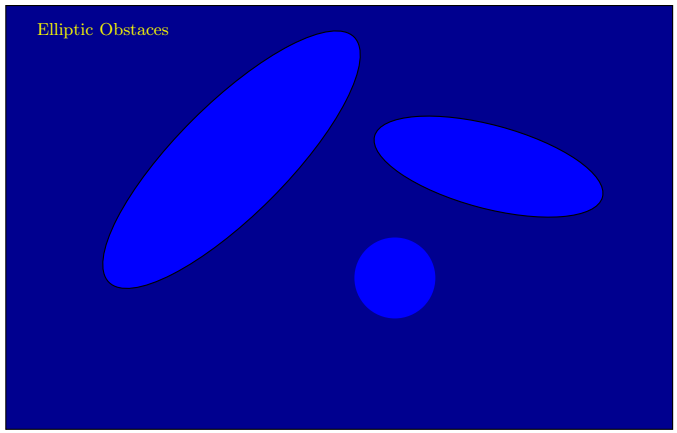
EGC Solution is unknown:

Shortest path from points p and q in the plane, avoiding elliptic obstacles.

The Trouble with Exactness, I.

EGC Solution is unknown:

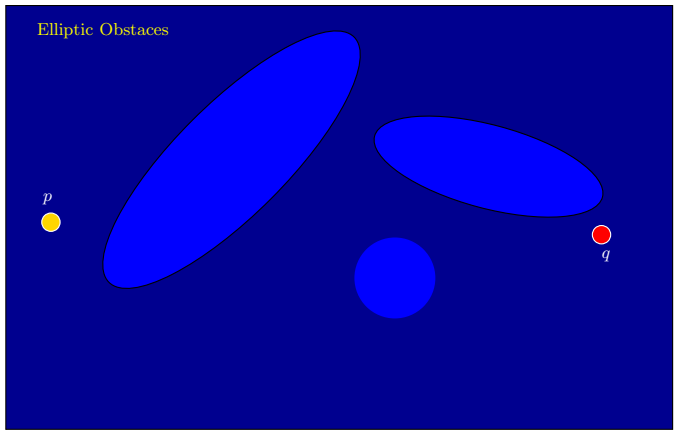
Shortest path from points p and q in the plane, avoiding elliptic obstacles.



The Trouble with Exactness, I.

EGC Solution is unknown:

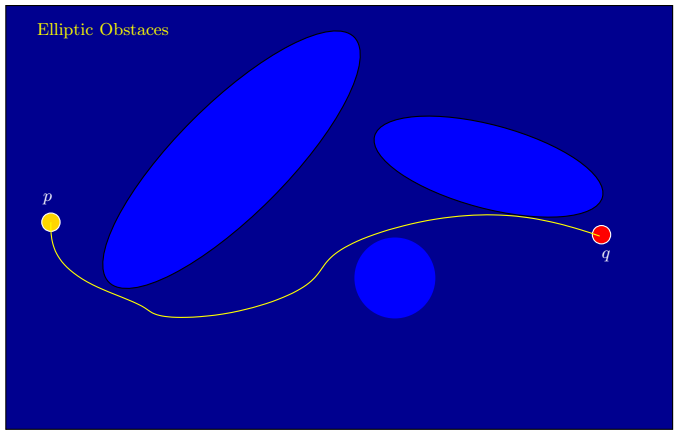
Shortest path from points p and q in the plane, avoiding elliptic obstacles.



The Trouble with Exactness, I.

EGC Solution is unknown:

Shortest path from points p and q in the plane, avoiding elliptic obstacles.



The Trouble with Exactness, I.

EGC Solution is unknown:

Shortest path from points p and q in the plane, avoiding elliptic obstacles.

- Elliptic arc length is transcendental

The Trouble with Exactness, I.

EGC Solution is unknown:

Shortest path from points p and q in the plane, avoiding elliptic obstacles.

- Most transcendental problems defy exact computation

The Trouble with Exactness, I.

EGC Solution is unknown:

Shortest path from points p and q in the plane, avoiding elliptic obstacles.

- E.g., non-holonomic or kino-dynamic motion planning

The Trouble with Exactness, II.

But, suppose ellipses are replaced by discs:

Theorem [CCKYP 2006]

Shortest Path amidst rational discs is in single-exponential time.

The Trouble with Exactness, II.

But, suppose ellipses are replaced by discs:

Theorem [CCKYP 2006]

Shortest Path amidst rational discs is in single-exponential time.

- Baker's Theory of Linear Form in Logarithms

The Trouble with Exactness, II.

But, suppose ellipses are replaced by discs:

Theorem [CCKYP 2006]

Shortest Path amidst rational discs is in single-exponential time.

- Rare computable transcendental problem

The Trouble with Exactness, II.

But, suppose ellipses are replaced by discs:

Theorem [CCKYP 2006]

Shortest Path amidst rational discs is in single-exponential time.

- BUT complexity is too high!

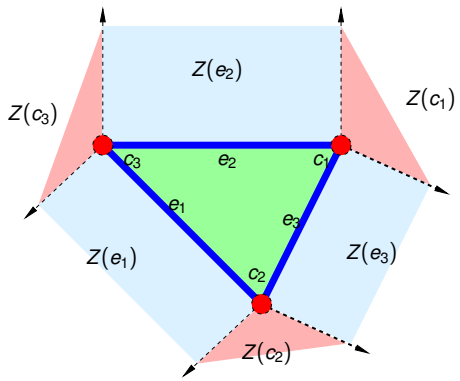
20-Year Old Voronoi Quest

Wanted: Vor Diagram of a Polyhedral set in 3-D

- 2-D is well-known
  - Kirkpatrick (1980), Y. (1984), Fortune (1986)
  - Exact Implementation: Burnikel, Mehlhorn, Schirra (1994)
- 3-D “Milestone” of Hemmer, Setter, Halperin (2010)
  - “Constructing the exact Voronoi diagram of arbitrary lines in 3-D space” (ESA 2010)

What is the Difficulty?

Classification of Voronoi Cells



- Based on (boundary) Features
- 3 Types of Features (+ their zones)
  - C: corners, E: edges, W: walls
- A set S of features defines a Voronoi cell $V(S)$ of some dimension $k = 0, 1, 2, 3$

- 10 Types of Voronoi curves ($k = 1$):
  - CCC, CCE, CCW, CEE, CEW, CWW, *EEE*, EEW, EWW, WWW

Theorem of Everett et al

Everett, Lazard, Lazard, Safey el Din, Gillot, Pouget (2007–9)

The Voronoi curve determined by three lines in space are connected components of these curves:

(a) a non-singular quartic if the 3 lines are pairwise skew but not all parallel to a common plane nor on the surface of a hyperboloid of revolution;

(b) a cubic and a line if the 3 lines are pairwise skew and lies on the surface of a hyperboloid of revolution;

(c) a nodal quartic if ...

(d) one or two parabolas or hyperbolas if ...

(e) Between 0 and 4 lines if ...

Algorithm for EEE CASE

Theorem [Everett et al (2009)]

There is a rational linear semi-algebraic test for:

(i) Given a point on a 2D cell, deciding on which of the connected components of the cell it lies;

(ii) Given a point on the trisector, deciding on which of the branches of the trisector it lies;

(iii) Ordering points on each branch of the trisector, knowing that they lie on the trisector.

- Bad news: there are 9 other CASES for Vor Curves
- Good news: the hardest CASE (EEE) is solved.

The Trouble with Exactness, III.

Why is an explicit exact algorithm so hard?

- Exactness requires knowing every type of degeneracy
- Exact primitives reduce to non-trivial tasks in algebra and algebraic geometry

"Algebra is generous,
she often gives more than is asked of her."

— JEAN LE ROND D'ALEMBERT (1717-83)

The Trouble with Exactness, IV: The World is Inexact

Physical Constants

The least accurately-known of physical constant is the gravitational constant G, with relative standard uncertainty of 0.12%.

— Guinness Book of World Records

What about Engineering accuracy? (Herb Volcker)

The Trouble with Exactness, IV: The World is Inexact

Physical Constants

Constant	Value	Std Uncertainty
Avogadro's constant N_A:	$6.02214129(27)x10^{23}mol^{-1}$	$4.4x10^{-8}$
Elementary charge e:	$1.602176565(35)x10^{-19}C$	$2.2x10^{-8}$
Constant of Gravity G:	$6.67384(80)x10^{-11}m^3kg^{-1}s^{-2}$	$1.2x10^{-4}$
Speed of Light c	$299,792,458ms^{-1}$	0

What about Engineering accuracy? (Herb Volcker)

The Trouble with Exactness, IV (contd.)

Exact motion planning is Dangerous

- Inexact robot dimensions
- Cannot control mobile robots accurately
- Knowledge of environment is limited by sensor accuracy
- Should never navigate a path that touches the obstacles

Instead: find paths with clearance $> \varepsilon$.

Next...

III. Towards Numerical Computational Geometry?

Can we really avoid Exact Computation?

What if the input is inexact? Or, you only want an approximation?

Quoting a 2004 paper

"... formulate a related problem in which the inexact input can be treated as exact...

So the convex hull of a set of points becomes the convex hull of a set of balls.

...

[But we] still end up with exact inputs for a well-defined computational problem."

Can we really avoid Exact Computation?

What if the input is inexact? Or, you only want an approximation?

Quoting a 2004 paper

"... formulate a related problem in which the inexact input can be treated as exact...

So the convex hull of a set of points becomes the convex hull of a set of balls.

...

[But we] still end up with exact inputs for a well-defined computational problem."

Can we really avoid Exact Computation?

What if the input is inexact? Or, you only want an approximation?

Quoting a 2004 paper

"... formulate a related problem in which the inexact input can be treated as exact...

So the convex hull of a set of points becomes the convex hull of a set of balls.

...

[But we] still end up with exact inputs for a well-defined computational problem."

— Robust Geometric Computation.
In CRC Handbook of Comp. and Discrete Geometry

When is Approximate Better than Exact?

- Which is better:

  Exact or approximate answers?

  - $\sqrt{3}$ or 1.732?
  - $\sqrt{19}-\sqrt{7}$ or 1.713?
- Depends on whom you ask: engineer or mathematician
- Joke: 100% correct, but totally useless.
- "Explicitization Problem": $\sqrt{3} \to 1.732$
- The approximation 1.732 "locates" $\sqrt{3}$ in the real line
  - This remark generalizes: we often prefer numerical to algebraic solutions

How Numbers Come into CG

Geometric problems as “Explicitization”:

- E.g., approximating a surface by a triangulation

Object *G* of Geometric Computation =

combinatorial cell complex + their embedding in space !!

- Root isolation (0-dimensional complex)
- Voronoi diagrams
- Curve/Surface Approximation
- Motion planning (configuration space)

Towards Numerical Computational Geometry?

Outline

Towards Numerical Computational Geometry?

Outline

Numerical computation is necessary
for “Explicitization Problems”

Towards Numerical Computational Geometry?

Outline

Like the EGC prescription,

we focus on the predicates !

Towards Numerical Computational Geometry?

Outline

Introduce a new $\varepsilon > 0$ (resolution parameter).

Intuitively, G is approximated to within ε
(varied interpretations)

Towards Numerical Computational Geometry?

Outline

Unlike numerical analysis,

we adjust numerical accuracy during the algorithm

Towards Numerical Computational Geometry?

Outline

"Ansatz" of Numerical CG

(A) Localize the computation

(B) Approximate (use "soft predicates")

Towards Numerical Computational Geometry?

Outline

How to integrate “Ansatz” into a global algorithm?

- Use iteration (from numerical computation)
- Use dovetailing (from computability theory)

Combined in Subdivision Framework!

Subdivision Framework

Assume a box predicate $\widetilde{C}(B)$:

INPUT: Usual input, $\varepsilon > 0$ and B_0

OUTPUT: the geometric object G restricted to B_0

0. $Q_0 \leftarrow \{B_0\}$ ..... [initialize]

I. $Q_I \leftarrow SUBDIVIDE(Q_0)$ ..... [until $\widetilde{C}(B)$ holds]

II. $Q_{II} \leftarrow SMOOTH(Q_I)$ ..... [balancing]

III. $G \leftarrow CONSTRUCT(Q_{II})$ ..... [combinatorial part]

IV. $Q_{out} \leftarrow REFINE(G, \varepsilon)$ ..... [ensure accuracy]

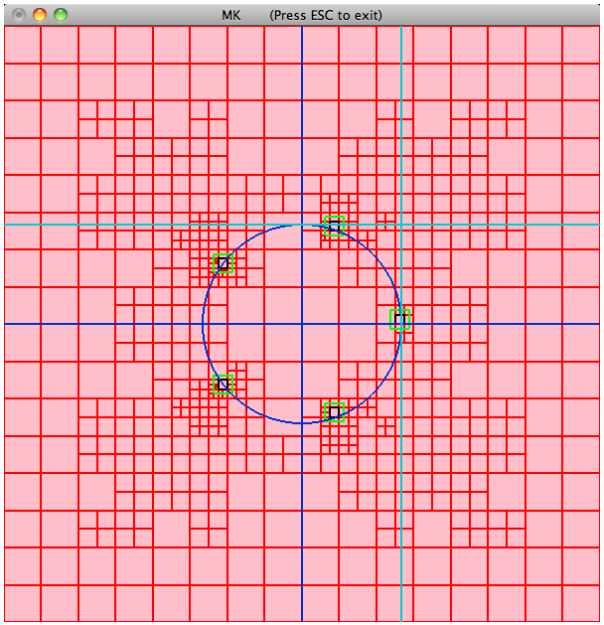
This is an "algorithm framework" (cf. PRM).

Simplify: assume $\varepsilon = \infty$.

Instantiations

1. Root Isolation (Real/Complex/Analytic)
2. Voronoi diagrams (2D)
3. Voronoi diagrams (3D)
4. Isotopic Surface Approximation
5. Motion Planning

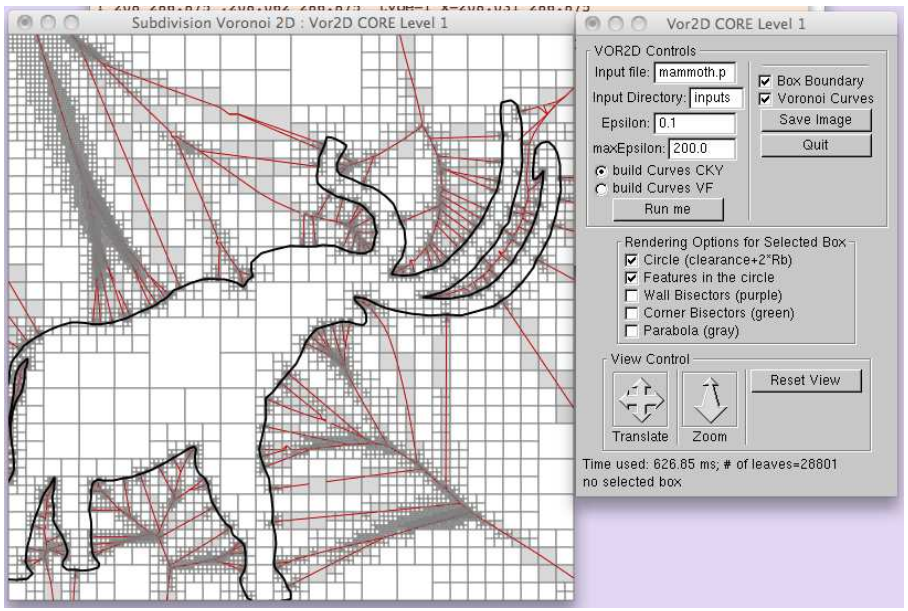
Complex Roots [ISSAC'11, SNC'12]



Instantiations

1. Root Isolation (Real/Complex/Analytic)
2. Voronoi diagrams (2D)
3. Voronoi diagrams (3D)
4. Isotopic Surface Approximation
5. Motion Planning

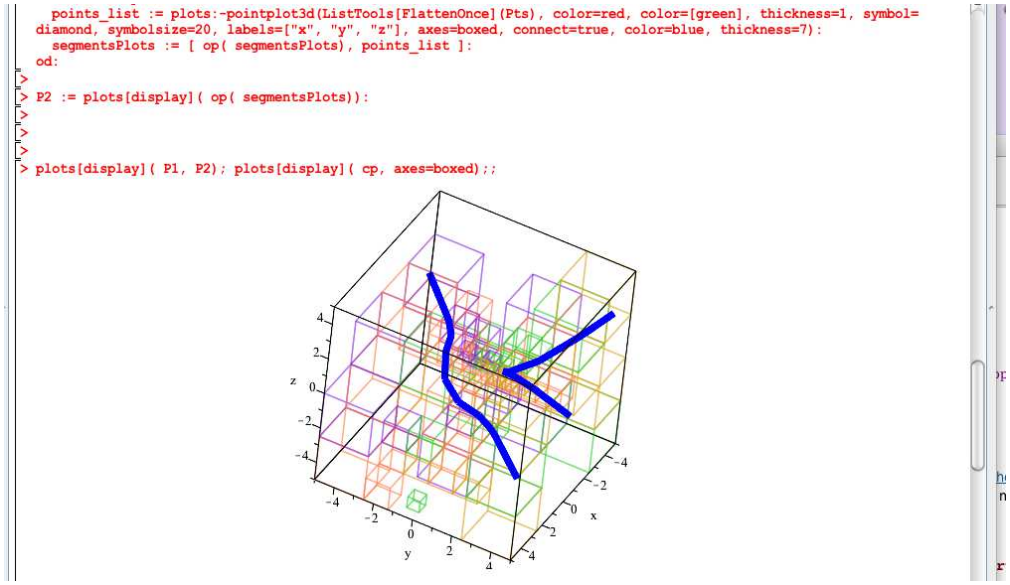
[ISVD'12]



Instantiations

1. Root Isolation (Real/Complex/Analytic)
2. Voronoi diagrams (2D)
3. Voronoi diagrams (3D)
4. Isotopic Surface Approximation
5. Motion Planning

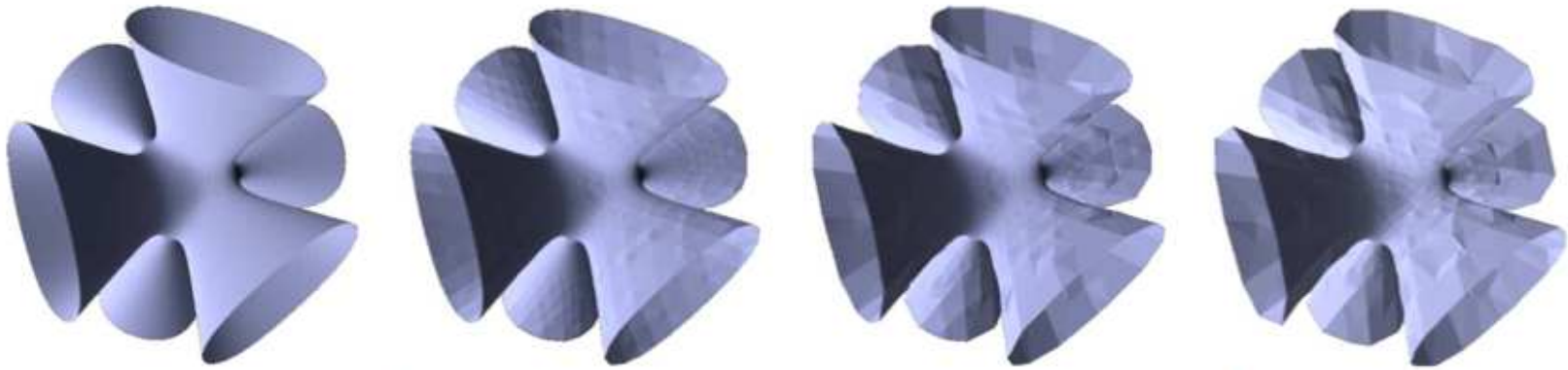
[Ongoing]



Instantiations

1. Root Isolation (Real/Complex/Analytic)
2. Voronoi diagrams (2D)
3. Voronoi diagrams (3D)
4. Isotopic Surface Approximation
5. Motion Planning

[SoCG'09, SPM'12]



(a) Cxyze. (b) PV. (c) Cxyz. (d) Rect-2.

Fig. 12. Eg6: $-x^4 - y^4 - z^4 + 4(x^2 + y^2z^2 + y^2 + z^2x^2 + z^2 + x^2y^2) - 20.7846xyz - 10.$

Instantiations

1. Root Isolation (Real/Complex/Analytic)
2. Voronoi diagrams (2D)
3. Voronoi diagrams (3D)
4. Isotopic Surface Approximation
5. Motion Planning

[SoCG'13, RCV'13]

SEE DEMO

Next...

I. The Exact Computation Paradigm

II. Beyond Exactness?

III. Towards Numerical Computational Geometry?

IV. Issues of Subdivision

IV. Conclusion

IV. Issues of Subdivision

Issues

- Correct use of ε
- Correct use of boxes (over-computing)
- Filters
- Singularity

Basic Robot Motion Planning

Fix a robot R_0 in $\mathbb{R}^k, (k = 2, 3)$

GIVEN:

Polyhedral obstacle set $\Omega \subseteq \mathbb{R}^k$

Start and Goals $\alpha, \beta \in C_{space}(R_0)$

FIND:

An Ω-avoiding path from α to β, if it exists.

Else report "NO PATH".

Dominant paradigm in the last 20 years:

- Probabilistic Road Map (PRM)
- Main Open Problem (Latombe 2011):

  Halting Problem (Climber's dilemma)

Resolution Exactness in Motion Planning

- On Resolution-based Methods
  - Grid versus Box method
- What is “resolution completeness”?
  - “(P) If exists path, will find one when resolution is fine enough”
  - Where is the converse? Nonhalting!
- Typical solution: invoke $\varepsilon > 0$
  - “(P) If exists path with clearance $\varepsilon > 0$, will find one;
    (N) If no path with clearance ε, report NO-PATH”
  - Need exact computation! Isn’t ε suppose to avoid it?

Resolution Exactness in Motion Planning

- On Resolution-based Methods
  - Grid versus Box method
- What is “resolution completeness”?
  - “(P) If exists path, will find one when resolution is fine enough”
  - Where is the converse? Nonhalting!
- Typical solution: invoke $\varepsilon > 0$
  - “(P) If exists path with clearance $\varepsilon > 0$, will find one;
    (N) If no path with clearance ε, report NO-PATH”
  - Need exact computation! Isn’t ε suppose to avoid it?

Resolution Exactness in Motion Planning

- On Resolution-based Methods
  - Grid versus Box method
- What is “resolution completeness”?
  - “(P) If exists path, will find one when resolution is fine enough”
  - Where is the converse? Nonhalting!
- Typical solution: invoke $\varepsilon > 0$
  - “(P) If exists path with clearance $\varepsilon > 0$, will find one;
    (N) If no path with clearance ε, report NO-PATH”
  - Need exact computation! Isn’t ε suppose to avoid it?

Resolution Exactness in Motion Planning

- On Resolution-based Methods
  - Grid versus Box method
- What is “resolution completeness”?
  - “(P) If exists path, will find one when resolution is fine enough”
  - Where is the converse? Nonhalting!
- Typical solution: invoke $\varepsilon > 0$
  - “(P) If exists path with clearance $\varepsilon > 0$, will find one;
    (N) If no path with clearance ε, report NO-PATH”
  - Need exact computation! Isn’t ε suppose to avoid it?

Resolution Exactness in Motion Planning

- On Resolution-based Methods
  - Grid versus Box method
- What is “resolution completeness”?
  - “(P) If exists path, will find one when resolution is fine enough”
  - Where is the converse? Nonhalting!
- Typical solution: invoke $\varepsilon > 0$
  - “(P) If exists path with clearance $\varepsilon > 0$, will find one;
    (N) If no path with clearance ε, report NO-PATH”
  - Need exact computation! Isn’t ε suppose to avoid it?

Resolution Exactness in Motion Planning

- On Resolution-based Methods
  - Grid versus Box method
- What is “resolution completeness”?
  - “(P) If exists path, will find one when resolution is fine enough”
  - Where is the converse? Nonhalting!
- Typical solution: invoke $\varepsilon > 0$
  - “(P) If exists path with clearance $\varepsilon > 0$, will find one;
    (N) If no path with clearance ε, report NO-PATH”
  - Need exact computation! Isn’t ε suppose to avoid it?

Resolution Exactness in Motion Planning

- On Resolution-based Methods
  - Grid versus Box method
- What is “resolution completeness”?
  - “(P) If exists path, will find one when resolution is fine enough”
  - Where is the converse? Nonhalting!
- Typical solution: invoke $\varepsilon > 0$
  - “(P) If exists path with clearance $\varepsilon > 0$, will find one;
    (N) If no path with clearance ε, report NO-PATH”
  - Need exact computation! Isn’t ε suppose to avoid it?

Resolution Exactness in Motion Planning

- On Resolution-based Methods
  - Grid versus Box method
- What is “resolution completeness”?
  - “(P) If exists path, will find one when resolution is fine enough”
  - Where is the converse? Nonhalting!
- Typical solution: invoke $\varepsilon > 0$
  - “(P) If exists path with clearance $\varepsilon > 0$, will find one;
    (N) If no path with clearance ε, report NO-PATH”
  - Need exact computation! Isn’t ε suppose to avoid it?

Planner is resolution-exact if:

- (P) If exists path with clearance 2ε, will find one
- (N) If no path with clearance $\varepsilon/2$, will report NO-PATH

Indeterminacy: what if the max clearance is between $\varepsilon/2$ and 2ε?

- But we have avoided the zero problem!

Next...

IV. Conclusion

Remark I

But you have basically ONE algorithm!

- Machine Learning: SVM
- Motion Planning in Robotics:
  - What we can learn from PRM
  - Soft Subdivision Search (SSS) framework

Remark II

My “Physics Envy” Problem

- Physical simulation:

  fluids, force, fields, etc.
- Not algebraic solution, since these are “explicitization problems”:

  Rigorous solution depends on using ε properly!

Directions

- EGC is powerful but also limiting
  Need for alternative computational models
- Resolution-exact computation
  Opens up previously untouchable problems
  - analytic roots, Voronoi diagram of nonlinear objects
- Research Topics:
  - Design of Soft Predicates, Design Global Strategies, Theoretical Foundations, Implementation
- Theoretical challenge: Complexity analysis of iterative algorithms
  - E.g., Continuous Amortization
    [BKY'2011, BK'2012, SY'2012]

Directions

- EGC is powerful but also limiting
  Need for alternative computational models
- Resolution-exact computation
  Opens up previously untouchable problems
  - analytic roots, Voronoi diagram of nonlinear objects
- Research Topics:
  - Design of Soft Predicates, Design Global Strategies, Theoretical Foundations, Implementation
- Theoretical challenge: Complexity analysis of iterative algorithms
  - E.g., Continuous Amortization
    [BKY'2011, BK'2012, SY'2012]

Directions

- EGC is powerful but also limiting
  Need for alternative computational models
- Resolution-exact computation
  Opens up previously untouchable problems
  - analytic roots, Voronoi diagram of nonlinear objects
- Research Topics:
  - Design of Soft Predicates, Design Global Strategies, Theoretical Foundations, Implementation
- Theoretical challenge: Complexity analysis of iterative algorithms
  - E.g., Continuous Amortization
    [BKY'2011, BK'2012, SY'2012]

Directions

- EGC is powerful but also limiting
  Need for alternative computational models
- Resolution-exact computation
  Opens up previously untouchable problems
  - analytic roots, Voronoi diagram of nonlinear objects
- Research Topics:
  - Design of Soft Predicates, Design Global Strategies, Theoretical Foundations, Implementation
- Theoretical challenge: Complexity analysis of iterative algorithms
  - E.g., Continuous Amortization
    [BKY'2011, BK'2012, SY'2012]

Directions

- EGC is powerful but also limiting
  Need for alternative computational models
- Resolution-exact computation
  Opens up previously untouchable problems
  - analytic roots, Voronoi diagram of nonlinear objects
- Research Topics:
  - Design of Soft Predicates, Design Global Strategies, Theoretical Foundations, Implementation
- Theoretical challenge: Complexity analysis of iterative algorithms
  - E.g., Continuous Amortization
    [BKY'2011, BK'2012, SY'2012]

Directions

- EGC is powerful but also limiting
  Need for alternative computational models
- Resolution-exact computation
  Opens up previously untouchable problems
  - analytic roots, Voronoi diagram of nonlinear objects
- Research Topics:
  - Design of Soft Predicates, Design Global Strategies, Theoretical Foundations, Implementation
- Theoretical challenge: Complexity analysis of iterative algorithms
  - E.g., Continuous Amortization
    [BKY'2011, BK'2012, SY'2012]

Directions

- EGC is powerful but also limiting
  Need for alternative computational models
- Resolution-exact computation
  Opens up previously untouchable problems
  - analytic roots, Voronoi diagram of nonlinear objects
- Research Topics:
  - Design of Soft Predicates, Design Global Strategies, Theoretical Foundations, Implementation
- Theoretical challenge: Complexity analysis of iterative algorithms
  - E.g., Continuous Amortization
    [BKY'2011, BK'2012, SY'2012]

Thanks for Listening!

“Algebra is generous,
she often gives more than is asked of her.”

— JEAN LE ROND D’ALEMBERT (1717-83)