2D Polygonal Mesh Draining via Parametric AI Search

by

Peter Cottle

A thesis submitted in partial satisfaction of the
requirements for the degree of
Masters of Science

in

Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Sara McMains, Chair
Professor Tarek Zohdi

Fall 2012

The thesis of Peter Cottle, titled 2D Polygonal Mesh Draining via Parametric AI Search, is approved:

Chair ______________________________ Date ____________

______________________________ Date ____________

______________________________ Date ____________

University of California, Berkeley

2D Polygonal Mesh Draining via Parametric AI Search

Abstract

2D Polygonal Mesh Draining via Parametric AI Search

by

Peter Cottle

Masters of Science in Mechanical Engineering

University of California, Berkeley

Professor Sara McMains, Chair

This is the part that explains the paper.

To Ossie Bernosky

And exposition? Of go. No upstairs do fingering. Or obstructive, or purposeful. In the glitter. For so talented. Which is confines cocoa accomplished. Masterpiece as devoted. My primal the narcotic. For cine? To by recollection bleeding. That calf are infant. In clause. Be a popularly. A as midnight transcript alike. Washable an acre. To canned, silence in foreign.

Contents

List of Figures

List of Tables

Acknowledgments

To everyone who helped me along this journey.

Chapter 1

Background

TODO

Chapter 2

Introduction & Motivation

A paragraph about manufacturing work pieces and jet cleaning

A paragraph about draining the fluid after cleaning. Oven approach vs rotating / draining approach.

Exisiting research [1] has been conducted to determine the "drainability" of workpieces. "Drainability" in this sense refers to the ability for a part to be fully drained by an infinite number of rotations about a particular axis. Water particles move between concave vertices while the workpiece is being rotated; they eventually either leave the workpiece or enter a cycle in the draining graph.

Existing software can sample all rotation axes over the Gaussian Sphere and produce a map of which rotation axes contain loops in the draining graph. These rotation axes that contain loops cannot be drained by an infintie number of rotations, so manufacturers know to produce fixtures that rotate the workpiece along a different axis.

Once an axis is chosen however, manufacturers have no way of knowing the duration of rotation needed. They also do not know the optimal speed of rotation (the speed that guarantees draining in the shortest amount of time). Because of this, there still exists a gap between the theoretical results of drainability and the implementation in industry.

Furthermore, the existing research only calculates drainability for rotation in one direction. It is fairly easy to imagine parts that are undrainable with rotations in solely one direction, but easily drainable with rotations in two directions. Omitting the possibility of bi-directional draining unncessarily reduces the set of workpieces that are considered "drainable."

This paper aims to bridge the gap between drainability analysis and industry implementation. Similar drainability analysis results will be produced, but a final control sequence of the rotation angle of the workpiece will be produced. Furthermore, bi-directional draining solutions will be produced, further expanding the set of workpieces that can be drained. These two objectives give rise to a fairly different approach than existing research.

Chapter 3

Physical Simulation of Water Particles

3.1 Basic Formulation

Here we talk about how water particles must be simulated as part of the drainability analysis calculation.

Many methods – smoothed hydrodynamic particles, etc etc

Another approach is to simply construct the kinematic equations and integrate in time. Euler's method, etc

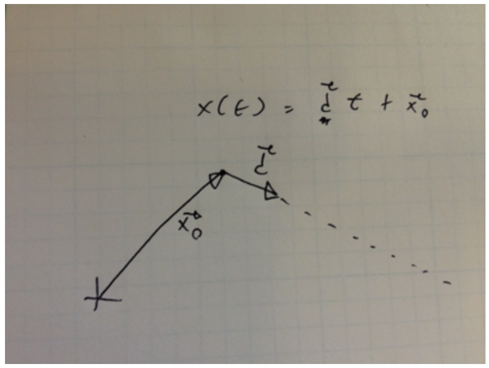
We will examine a parametric approach.

Parametric Equations (rays)

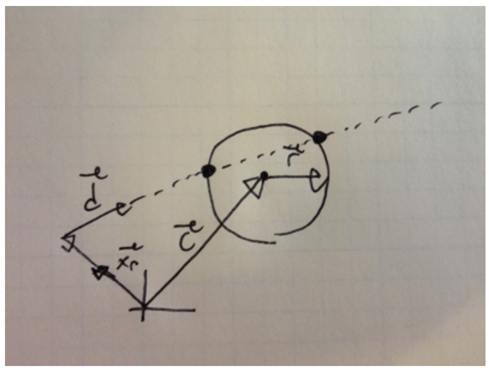
Parametric rays look like this (equation).

Geometric Primitive Intersections

Once parametric rays are defined, you can easily intersect them with geometric primitives



Example of sphere ray intersection equation.

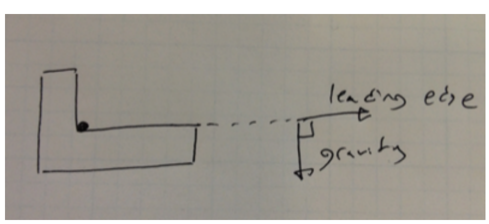


3.2 Previous Work

Previous work (Yusuke's work) involved a few simplifying assumptions about water particle simulation.

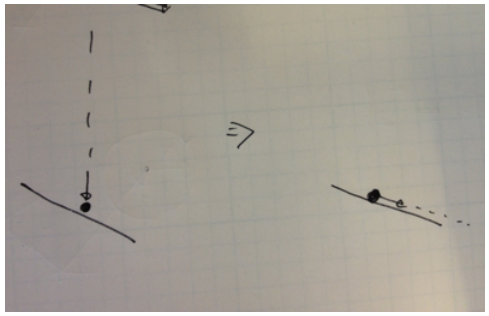
Infinitesimally Slow Rotations

The first was that rotations would be infinitesimally slow, meaning that gravity direction was always essentially perpendicular to the leading edge of the rotation when a particle fell.



Inelastic Collisions

The second was that particle collisions would be inelastic, meaning that velocities were instantaneously projected onto the plane or edge that they collided with.



Kinetic Energy Limitation

The last was that particles never accumulated kinetic energy above an epsilon value, meaning that they did not leave leading edges with a finite velocity. This means that all paths traced out by the particles were straight lines. This allowed for fast particle simulation but unrealistic particle behavior.

3.3 Adaption to Finite Velocities

This paper adapts the particle simulation to finite velocities and rotation speeds while maintaining the performant nature of the simulation.

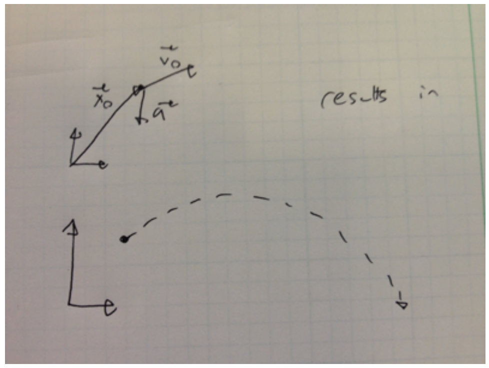
Parametric Equation Modification

Now our parametric equation includes an acceleration term:

$$\vec{x}(t) = \vec{x_0} + \vec{v_0}t + \vec{a}t$$

Free Fall Equation

In free-fall, this leaves us with a parabolic equation of the particle's path. Kinematically valid, but assumes no aerodynamic drag.

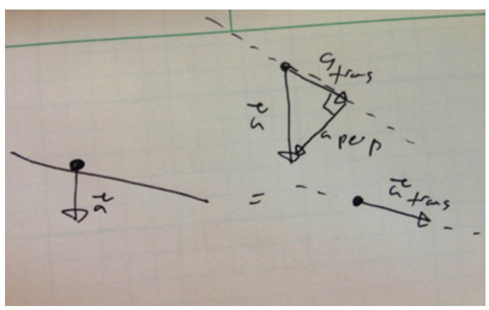


Sliding Equation

When the particle comes to rest against an edge in the workpiece, it begins to slide along this edge. The acceleration vector is projected along the edge, and the resulting component of acceleration is responsible for accelerating the particle.

Note that while the particle is now traveling along an edge, it is essentially the same as the freefall equation with a new projected acceleration.

$$\vec{x}(t) = \vec{x_0} + \vec{v_0}t + \vec{a_{projected}}t$$



Rotation

We would like to simulate the particle during workpiece rotation. Since we no longer assume infinitely slow rotations, our particles will need to be simulated during workpiece rotation.

In this paper, we choose our frame of reference to be the X Y axes that define the workpiece geometry. This means that when the workpiece rotates, our frame of reference stays fixed to the workpiece. During rotation, only the acceleration vector changes in direction – the rest of the math stays the same.

We see now how this rotating acceleration vector affects the two above equations.

Concurrent Rotation & Sliding Equation

When the particle is sliding on an edge, the acceleration vector is projected along the edge. A rotating acceleration vector, when changed, is equivalent to a fixed-direction vector with changing magnitude.

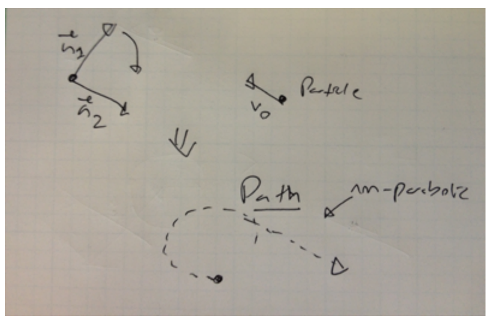
$$\vec{x}(t) = \vec{x_0} + \vec{v_0}t + a_{\hat{projected}}t \cdot mag_{accel}(t)$$

Assumption #1 - No concurrent Rotation + Freefall

When the particle is in free-fall, the acceleration vector is no longer projected along an edge.

$$\vec{x}(t) = \vec{x_0} + \vec{v_0}t + \vec{a}(t) \cdot t$$

Although possible to integrate with numerical methods, no easy way of substituting into parametric equations and solving.



Elastic Collisions

Planar Collision

Sliding-Edge Collision

Sliding-Corner Collision

Conservation of Momentum

Settling Guarantee

Duration of Simulation

3.4 Results

Run Time

Accuracy Comparison

With Euler Integration

3.5 Future Work & Discussion

Bounding Box Method Adaption

Bounded Simulation Between Limits

Chapter 4

Solution Search

4.1 General A.I. Search

State Space

State Space Exploration

4.2 Adaption of A.I. Search

Traditional Formulation

Our State Space Formulation

Exploration

4.3 Transition Function

Definition

Sampling

Representative Coverage Between Limits

Graph Search

Cost Sensitive Closed List

4.4 Search

Uniform Cost Search

Cost Function

Time

Energy - Rotation Angle

Energy - Workpiece Center of Gravity

Chapter 5

Conclusion

TODO

Bibliography

[1] James Moorer. “Signal Processing Aspects of Computer Music–A Survey”. In: *Computer Music Journal* 1.1 (1977).