

Interactive Visualization of 3D Configuration Spaces

Justin Stoecker
Department of Computer Science
University of Miami
justin@cs.miami.edu

Victor Milenkovic
Department of Computer Science
University of Miami
vjm@cs.miami.edu

ABSTRACT

We present a tool that enables interactive exploration of the configuration space of a planar robot that navigates static obstacles. The robot and obstacle boundaries consist of circular edges, and our visualization provides both 2D and 3D views of the contact surfaces of these objects. We work from an exact parametric representation of the free space boundary and construct a triangle mesh approximation that is used for interactive rendering.

Categories and Subject Descriptors

I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*boundary representations*

Keywords

Configuration space; visualization; interactive; triangulation

1. INTRODUCTION

Our video covers the main features of software we have developed to visualize 3D configuration spaces of 2D robots. The input for the software is an exact parametric representation of the free space boundary; this representation can be generated by the techniques and program described in [2]. From the exact geometry, we can visualize a 2D cross-sectional view of the configuration space and a 3D triangle mesh that closely approximates the exact boundary of the free space. The triangle mesh is a compact representation that is suitable for real-time rendering, yet we preserve the fine details of the exact surface.

A key feature of our visualization is the ability to manipulate the robot configuration and interactively explore the 2D and 3D configuration space views. This utility is convenient for introducing configuration spaces to students, for debugging path planning algorithms, and for illustrating how path planning can be done in the configuration space.

1.1 Related Work

Atariah and Rote [1] presented a visualization that shows both the workspace and configuration space using simple polygonal geometry. This work highlighted contact types that were used to produce a 3D view of obstacles in the configuration space, and animations were ray traced offline. It is also common to visualize a 3D configuration space by

sampling the rotation dimension to get cross sections, although this is time-consuming and can miss subtle changes in the surface without a fine sampling. For interactive visualization, some online applets demonstrate 2D configuration spaces for robotic manipulators^{1,2}.

2. TRIANGLE MESH CONSTRUCTION

We build a triangulated surface that approximates the exact parametric representation of the free space boundary for the purpose of interactive rendering: triangle meshes are the standard form of input for the graphics pipeline used for real-time rendering.

The robot and obstacle boundaries are defined by circular arc edges. Arcs are parameterized by normal angle α in the standard manner. For a fixed orientation, the free space is the set of translations such that the robot does not overlap the obstacle. It is bounded by arcs called sub edges, also parameterized by α . Each sub edge corresponds to a specific contact pair: an edge of the robot in contact with an edge of the obstacle (Fig. 2). Sub edges arising from different orientations of the robot are identified if they correspond to the same contact pair and if the two neighboring sub edges (along the boundaries of the free spaces) also have corresponding contact pairs. Robot orientation is parameterized by θ . A collection of identified sub edges is called a sub, which is parameterized by α and θ .

We display the subs in 3D. The x and y correspond to translation and z corresponds to rotation (θ). The subs bound the free space of the robot in this 3D configuration space (Fig. 1). In configuration space, the sub is curved. However, in (α, θ) parameter space, the sub is a trapezoid with parallel top and bottom of fixed θ . Two subs can share a boundary on the left or right along a curve corresponding to two contact pairs. We sample such a double contact curve at maximum or minimum θ values of subs which it bounds. We add extra sample points to obtain a good polygonal approximation to the curve. The triangulated approximation to a sub is bounded on the left and right by the appropriate subset of the polygonal approximation of its left and right double contact curves. The top and bottom of a sub are circular arcs, and the top of one sub can overlap the bottom of another if the maximum θ of one equals the minimum θ of the other. We sample the entire circle containing these arcs at the left and right α of each sub it bounds and then subsample to obtain a good approximation. This process

¹<http://ford.ieor.berkeley.edu/cspace/>

²<http://www.cs.unc.edu/~jeffi/c-space/robot.xhtml>

determines four polygonal chains bounding the sub triangulation: left, right, bottom, top. For each sub, we consider the chains in (α, θ) space. In this space, the chains are straight sides of the trapezoid. We triangulate the interior in a standard manner respecting the boundary vertices and then transform back into 3D. There is no duplications of vertex points: triangulation vertices that are shared by two subs are actually indices into a common point array. This way, the set of triangles from all subs forms a triangulation of the 3D free space boundary with no gaps.

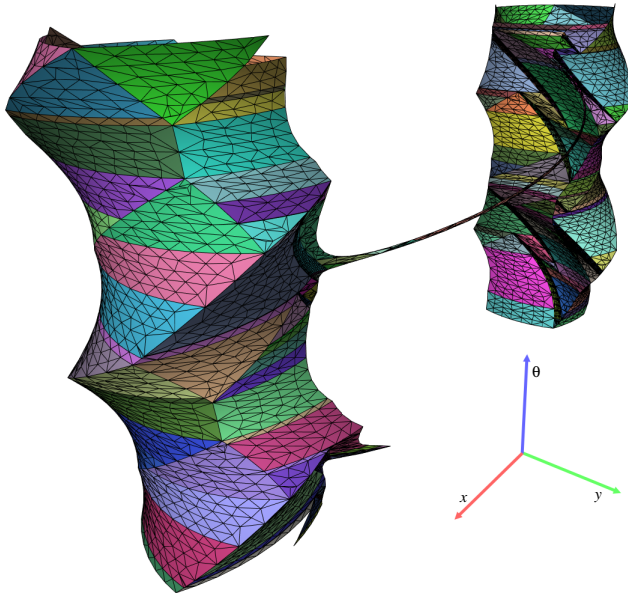


Figure 1: A triangulated approximation to a free space boundary viewed from outside. Each sub is drawn in a unique color, and a black wireframe is overlaid to show the triangles. The two vertical shafts indicate larger spaces the robot can translate and rotate within; the narrow tube connecting the shafts illustrates that the robot can move between these areas with restricted translation and rotation.

3. VIDEO OVERVIEW

Our video begins by introducing the 2D scene with the robot and obstacle, and demonstrates how the robot can be translated and rotated by the user. We then describe the parts that make up the complete representation used to model the configuration space. The 3D visualization follows, and a side-by-side comparison illustrates the relation of the 2D and 3D geometry.

We highlight how the user can configure the robot and see its corresponding 3D point in the configuration space. A brief summary of the triangulation algorithm is provided, and we conclude by showing path planning in the configuration space. Some additional scenes demonstrate the robustness of the software with narrow configuration space channels. We also show how linear robot and obstacle edges can be approximated in practice using arcs with very large radii (10^6).

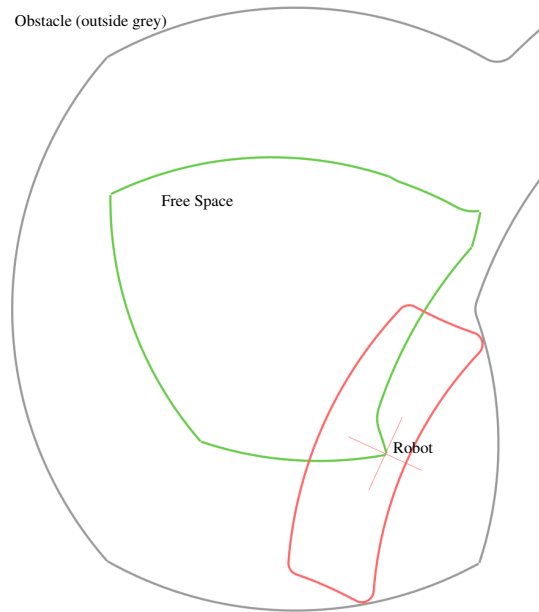


Figure 2: Visualization of the cross section of the free space for a fixed orientation of the robot. Since the robot (red) is at the junction of two sub edges (green), it has two contacts with the obstacle (gray).

4. SOFTWARE DETAILS

The software in this video was written in Java using the OpenGL graphics API through JOGL³. GLSL shaders are used for rendering effects, and YAML is used for configuration files. The visualization software⁴ can be experimented with and is bundled with some sample scenes; the software that produces the input, described in [2], can be downloaded⁵ as well.

5. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0904707. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

6. REFERENCES

- [1] D. Atarhah and G. Rote. Configuration space visualization. In *Proceedings of the 2012 Symposium on Computational Geometry*, pages 415–416, 2012.
- [2] Victor Milenkovic, Elisha Sacks, and Steven Trac. Robust complete path planning in the plane. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*. WAFR, 2012. Invited version to appear in IEEE T-ASE.

³<http://jogamp.org/jogl/www/>

⁴<http://jstoecker.github.com/cspace>

⁵<http://www.cs.miami.edu/~vjm/robust/>