Legs, Hands, and Wheels: Bridging the Gap Between High-Level Planning and Low-Level Control

James Kuffner

The Robotics Institute
Carnegie Mellon University

(calledly full-time research at Google)
Humanoid Motion Planning (1995-2011)

- Stanford University
  1995-1999
- University of Tokyo
  JSK Lab
  1999-2001
- Carnegie Mellon University
  The Robotics Institute
  2001-present
- Digital Human Research Center (AIST)
  2001-present
Self-driving Cars

Google

SMATER THAN YOU THINK

Google Cars Drive Themselves, in Traffic

Dmitri Dolgov, a Google engineer, in a self-driving car parked in Silicon Valley after a road test.

By JOHN MARKOFF
Published: October 8, 2010

MOUNTAIN VIEW, Calif. — Anyone driving the twists of Highway 1 between San Francisco and Los Angeles recently may have glimpsed a Toyota Prius with a curious funnel-like cylinder on the roof. Harder to notice was that the person at the wheel was not actually driving.
Challenges for Motion Planning in the “Real World”

- Uncertainty
  - Prior models
  - Perception
  - Control

- Search Space
  - Continuous
  - High-dimensional

- Hard, real-time constraints
Footstep Placement Planning

"Footstep planning among obstacles for biped robots"
[ Kuffner, Nishiwaki, Kagami, Inaba, Inoue, IROS2001 ]

[Image of a robot with planned footstep sequence and obstacles]

James Kuffner  (CMU/Google)  
ICRA2011 : Workshop on Motion Planning for Physical Robots
Search Over Possible Footstep Placements

Joel Chestnut
RI PhD student
2001-2007

Initial Configuration

DISCRETE PLACEMENTS

INTERMEDIATE POSTURE
Planning approaches

Plan for all degrees of freedom

Computationally expensive
Uses the full capabilities of the robot

Footstep Planning

Abstract away all leg details

Fast
Ignores leg capabilities
Action model based on potential footstep motions

- $(x,y,\theta)$ footstep locations relative to stance foot

- Fixed sampling of possible footsteps
Discrete Planning

- **Input**
  \[ x_{\text{init}}, X_{\text{goal}}, e, A \]

- **Successor function**
  \[ x' = \text{Succ}(x, a, e) \]

- **How do you choose the set of actions?**
State Space Categorization

Single Support

Double Support
Planner state describes contact configuration

- State Representation:
  - \((x, y, \theta, \text{leg})\) of current stance foot
  - roll, pitch, and height determined by terrain shape
- Height map terrain
# Location Cost

<table>
<thead>
<tr>
<th>Metric Evaluation</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle</td>
<td>$\cos^{-1}(n_z)$</td>
</tr>
<tr>
<td>Roughness</td>
<td>$\frac{1}{N} \sum_{c \in \text{Cells}}</td>
</tr>
<tr>
<td>Largest Bump</td>
<td>$\max_{c \in \text{Cells}} (h_c)$</td>
</tr>
<tr>
<td>Stability</td>
<td>$\frac{1}{N} \sum_{c \in \text{Cells}} [(h_c - h_p)w_c]$</td>
</tr>
<tr>
<td>Safety / Surroundings</td>
<td>$\max_{c \in \text{SurrCells}} (h_c)$</td>
</tr>
</tbody>
</table>

**Input Terrain**

**Metric Evaluation**
Location Metrics

Angle

Roughness

Largest Bump

Stability

Safety

All
Search for a Global Footstep Path

\[ \text{PlanPath}(s_{\text{init}}, s_{\text{goal}}, A, e) \]

// Init search (state, cost, expected, parent)
Q.Insert(s_{\text{init}}, 0, 0, NULL);
while running_time < t_{\text{max}} do
    s_{\text{best}} \leftarrow Q.ExtractMin();
    if GoalReached(s_{\text{best}}, s_{\text{goal}}) then
        return s_{\text{best}};
    end
    foreach a \in A do
        s_{\text{next}} \leftarrow s_{\text{best}} + a;
        c_l \leftarrow \text{LocationCost}(e, s_{\text{next}});
        c_s \leftarrow \text{StepCost}(e, a);
        c_e \leftarrow \text{ExpectedCost}(e, s_{\text{next}});
        Q.Insert(s_{\text{next}}, s_{\text{best}}.cost + c_l + c_s, c_e,
                  s_{\text{best}});
    end
end
Online Footstep Planning

[Kuffner, Nishiwaki, Kagami, Inaba & Inoue, ICRA 2003]
Online Experiments

[Chestnutt, Kuffner, Nishiwaki, Kagami, Inaba & Inoue, 2003]
Honda ASIMO at CMU
(2004 – 2008)

[ Chestnutt, Michel, Kuffner, Kanade, IROS 2007 ]
Planning Dynamic Actions
Key ideas of footstep planning

• Plan in the low-dimensional space of contact configurations (stances)
• Approximate path existence between stances by describing the limits of the robot and its controller
• Evaluate stances for stability and properties needed by the controller
Limitations

- Does not know anything about the physical makeup (softness, friction, strength) of the world.
Autonomous Grasping & Manipulation (2000-2010)
Sampling-Based Planning with Rapidly-exploring Random Trees (RRTs)

“RRT-Connect” [Kuffner, LaValle ICRA ‘00]
RAVE: Online Manipulation Planning (2001)
Stable Grasp Generation

1. Approach Target
2. Close Fingers
3. Compute Contacts

CMU PhD thesis: Rosen Diankov
Feasible Grasp Generation
Automatic Regrasping (2006)

Object-Specific 6D Pose Extraction

- Modeling Object Pose Error

CMU PhD thesis: Rosen Diankov
Pose Sets due to a Curve
Mean Images of Induced Pose Sets

CMU PhD thesis: Rosen Diankov
“HERB” : Home-Exploring Robot Butler
(2008 – 2010)
Planning With Constraints
Whole-body Constrained Planning

Simultaneous Constraints and Goal Sampling
Using TSR chains

[ Berenson, Chestnutt, Srinivasa, Kagami, Kuffner, *Humanoids 2009* ]
Self-driving Cars

Google Cars Drive Themselves, in Traffic

Dmitri Dolgov, a Google engineer, in a self-driving car parked in Silicon Valley after a road test.

By JOHN MARKOFF
Published: October 9, 2010

MOUNTAIN VIEW, Calif. — Anyone driving the twists of Highway 1 between San Francisco and Los Angeles recently may have glimpsed a Toyota Prius with a curious funnel-like cylinder on the roof. Harder to notice was that the person at the wheel was not actually driving.
Accomplishments

- A total of more than 145,000 autonomous miles
- 10 high-complexity routes of roughly 100 miles each without human intervention.
Vehicle Pose + Derivatives
(position, velocity, acceleration, orientation, angular vel, angular accel)

- High-dimensional in principle
- **Driving model abstraction** greatly reduces the dimensionality (i.e. speed up, slow down, change lane)
Driving Model Abstraction

Localized Pose
Correction to global Lat/Lng coordinates

Mike Montemerlo, Andrew Chatham
Driving Model Abstraction

Road Graph
Lanes, Traffic Lights, Crosswalks, etc.
Driving Model Abstraction

Perception
Static objects, cars, pedestrians, traffic lights

Jiajun Zhu, Nathaniel Fairfield, Russell Smith, Hector Yee, Dirk Haehnel
Driving Model Abstraction

Output: Trajectory
Curve in x,y with speed profile, stop line, ACC targets.

Dmitri Dolgov, Chris Urmson
Trajectory Planning

Trajectory
Curve in x,y with speed profile, stop line, ACC targets.
Route Planning

• Minimize overall route complexity (lane changes, unprotected left turns, no U-turns)
• Route duration used as additional metric
A Self-Driving Prius

- Laser
- GPS
- Wheel Encoder
- Radars
- Safety Driver
Summary

• Motion planning techniques can be made practical, useful, and even essential for physical robots.

• Model reduction is a powerful tool for making planning tractable.

• Need to think carefully about proper abstractions
  – Discretization of state and actions
  – Representing prior models