

Path Planning with Incremental Roadmap Update for Large Environments

Tsai-Yen Li and Chih-Ching Chang

Computer Science Department
National Chengchi University
Taipei, Taiwan, R.O.C.

Email: {li, s8522}@cs.nccu.edu.tw

ICRA2001

May 24, 2001

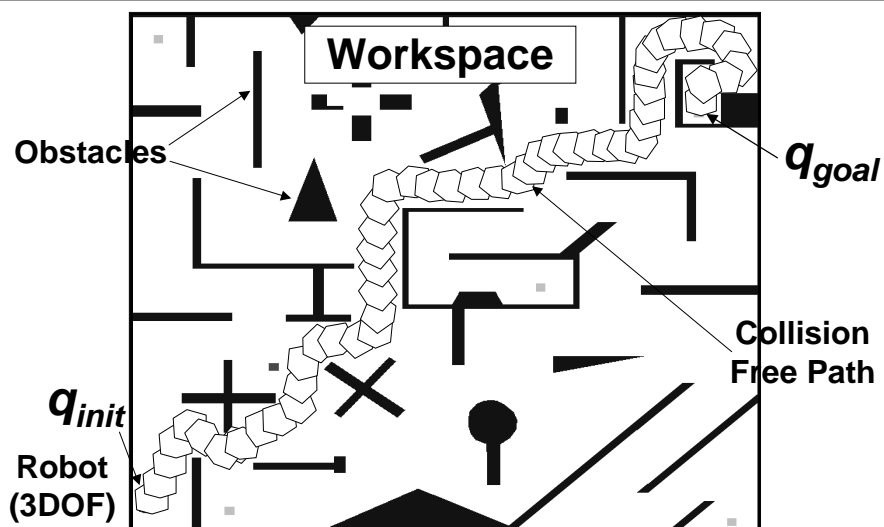
Outline of the Talk

- Introduction
- Related work
- Problem description
- Review of intelligent navigation interface
 - Limitations of the implementation
- Incremental roadmap update with RRT
- Experimental results
- Conclusions and future work

Introduction

- **Applications:**
 - 3D navigation in a virtual environment
 - Tele-operation applications
- **Problems:**
 - The level of navigation control that a user need to provides is too low.
 - The frame rate for complex scenes is still not high enough for precise control.
- **Demands for better navigation interface**
- **Proposal:** using motion planning techniques to assist user navigation

The Basic Path Planning Problem

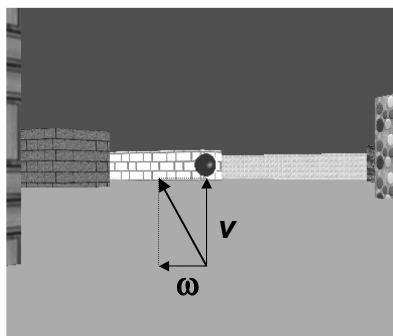


Related Work

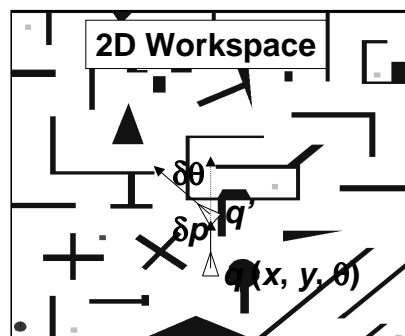
- **User Interface (UI) design in Graphics**
 - Direct manipulation metaphor [Shneiderman97]
 - Agent-based intelligent UI [Lieberman98]
- **Path planning in Robotics**
 - Best-first path planners (for low DOF) [Barraquand91, Latombe91]
 - Randomized path planners [Overmars94, Kavraki96]
- **Intelligent UI with motion planning**
 - For auto-navigation applications [Drucker94, Li99]
 - For computer animation applications [Kuffner98]
 - For intelligent navigation [Li00]

Navigation User Interface

- **Multi-shot path-planning problem**
- **Planning efficiency is critical.**

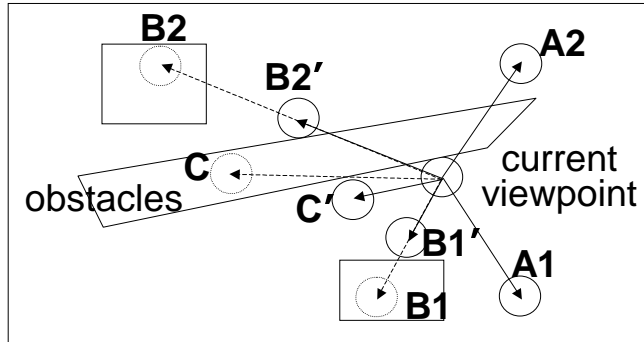


$$\begin{aligned}\delta p &= v * \delta t \\ \delta \theta &= \omega * \delta t\end{aligned}$$



q : current viewpoint config
 q' : next viewpoint config

Defining the Planning Problem: Predicting User Intention



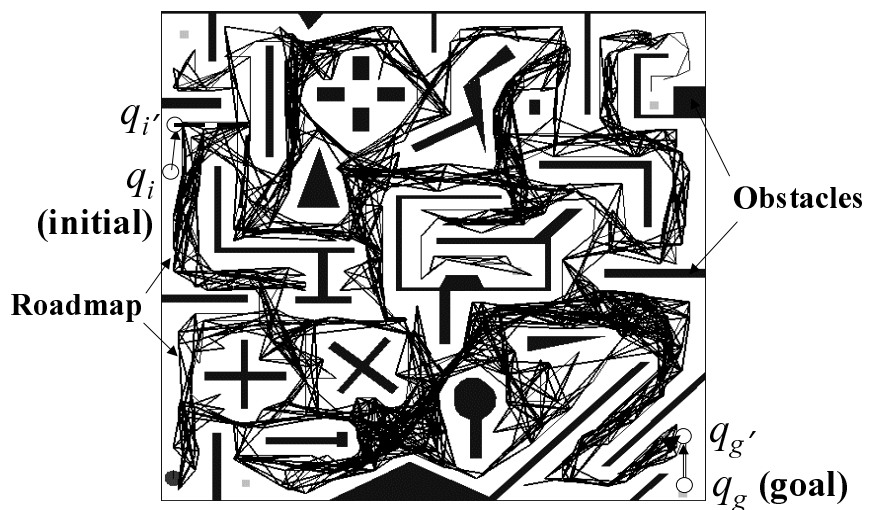
Possible Cases:

- A: No modification
- B: Direct modification
- C: Indirect modification

Results:

1. Trivial path: A1, B1', C'
2. Non-trivial path
3. No path

Randomized Roadmap Planner Improves Navigation Efficiency



Improvement: 73% on overall navigation time (VR2000)

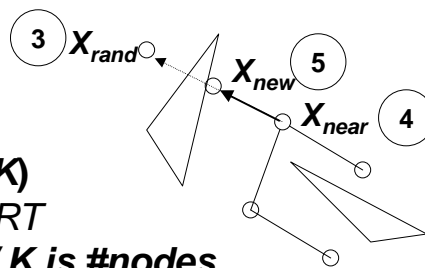
Scalability Problem for Large Environments

- **Previous roadmap planner is fast because:**
 - Roadmap construction is done in preprocessing.
 - Collision detections are through table lookups.
- **Main limitations:**
 - Roadmap is too large to compute and store for large environments.
 - **Observation:** confined problem in a local region.
 - **Proposal:** update the roadmap incrementally.
- **Randomized roadmap is too costly to update.**
 - **Proposal:** adopt a simpler representation such as Rapidly-Exploring Random Tree (RRT) [Lavalle00].

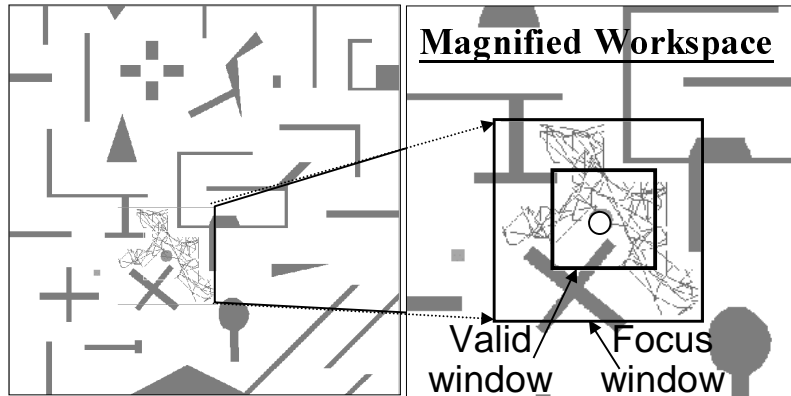
Algorithm for Generating RRT

Generate_RRT(x_{init} , K)

1. $T.init(x_{init});$ // T is RRT
2. **for** $k = 1$ **to** K **do** // K is #nodes
3. $x_{rand} \leftarrow$ Random_Configuration();
4. $x_{near} \leftarrow$ Nearest_Neighbor(x_{rand} , T);
5. $x_{new} \leftarrow$ New_Configuration(x_{near} , x_{rand});
6. $T.add(x_{near}, x_{new});$
7. **Return** T



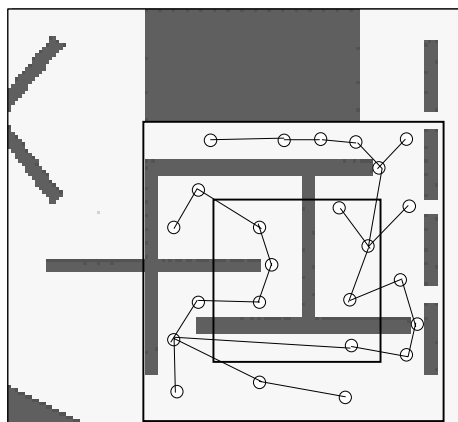
RRT and Update Windows



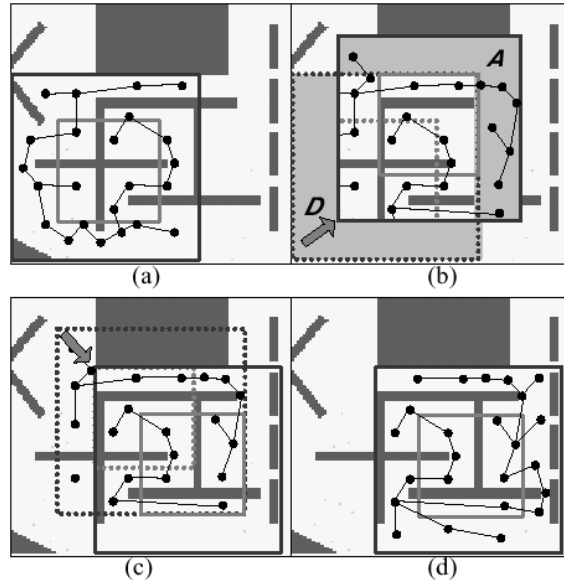
- **Focus window:** region of interest for path-finding
- **Valid window:** incrementally update roadmap as the viewpoint exits the window.

Incremental Update of RRT (1)

- **Dynamic addition and deletion of nodes**
- **2D range search tree for collision checks**

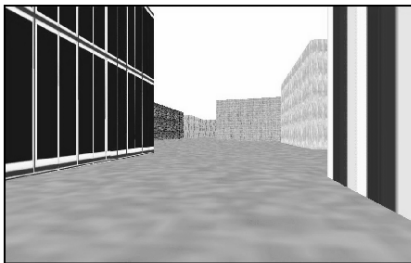


Incremental Update of RRT (2)

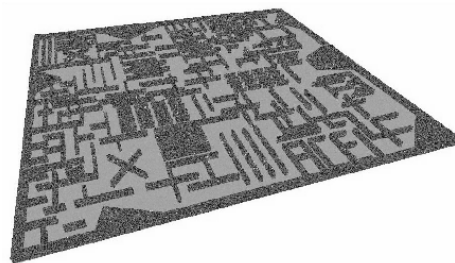


Experimental User Interface: A Maze in VRML

- **Planner:** implemented in Java.
- **VRML browser:** modifying in Java3D.
- **Mazes:** generated by a maze editor in Java.

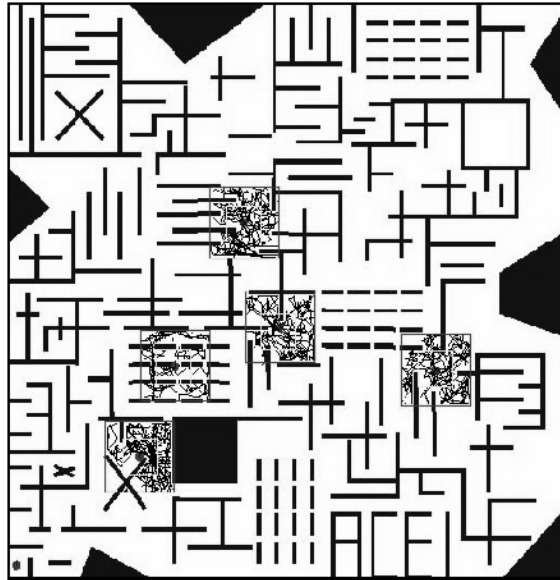


Perspective View



Top View

Examples of RRT's at Different Instances in a Large Virtual World



Experimental Data for Different Environments

	P1 with small world	P2 with small world	P2 with large world
Preprocessing time (ms)	4062	94	125
No. of total steps	1215	1202	3165
No. of planning requests	16	25	42
Time for each path search requests (ms)	275	50	83
Time for each path smoothing (ms)	186	49	81
No. of window updates	N/A	74	89
Time for each window update (ms)	N/A	41	22

P1: old randomized roadmap planner

P2: new incremental planner with RRT

Small world: 128x128 with 50 obstacles

Large world: 256x256 with 300 obstacles

Conclusions and Future Work

- **Conclusions:**

- Successfully extending the planner to consider the case of **large virtual worlds**
- Could be of general interests for on-line planning problems with **a large or unbounded world** in Robotics

- **Future work:**

- Data management for **dynamic object loading** and **graphics rendering** for large environments
- **Adaptive sizes** for valid window and focus windows with different CPU processing speeds