Introduction to Robotics

Lecture 11

Jianwei Zhang, Lasse Einig
[zhang, einig]@informatik.uni-hamburg.de

University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Technical Aspects of Multimodal Systems

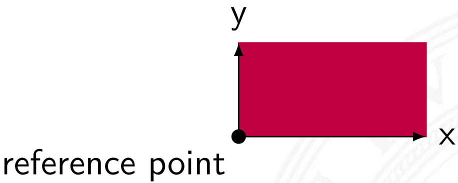
June 24, 2016

Outline

Outline (cont.)

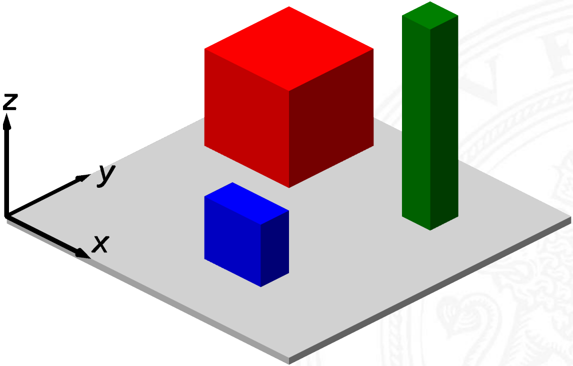
Task-level Programming – Basics

Robot → Single reference point with physical attributes



Task-level Programming – Basics

Work space $\rightarrow$ The cartesian space of the environment

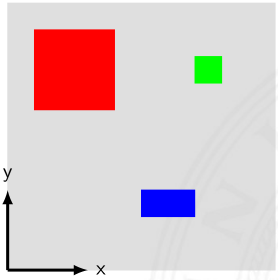


Task-level Programming – Basics

Configuration space `C` $\rightarrow$ Set of all possible configurations

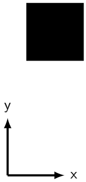
Task-level Programming – Basics

Obstacles in work space → `C-Obstacles` in configuration space



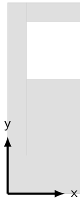
Task-level Programming – Basics

Obstacle space $\rightarrow$ Union of C-Obstacles



Task-level Programming – Basics

Free space $\mathtt{C_{free}}$ the complement of Obstacle space



Task-level Programming – Basics

Robot → Single reference point with physical attributes

Work space → The cartesian space of the environment

Configuration space `C` → Set of all possible configurations

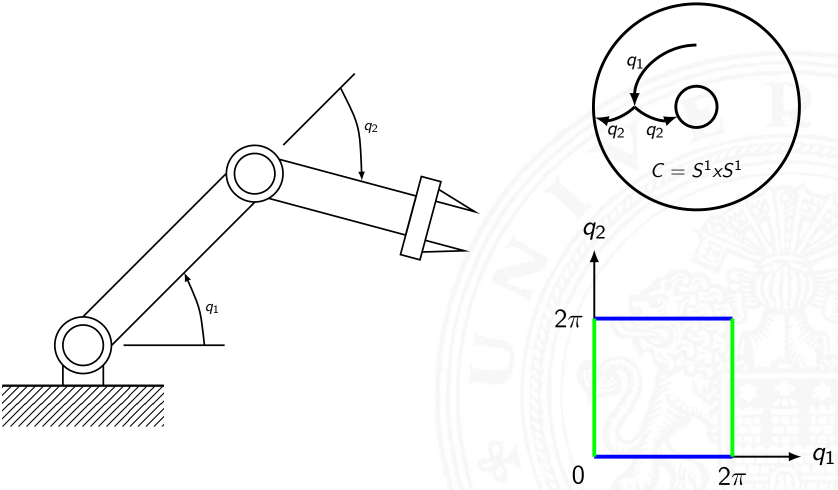
Obstacles in work space → `C-Obstacles` in configuration space

Obstacle space → Union of C-Obstacles

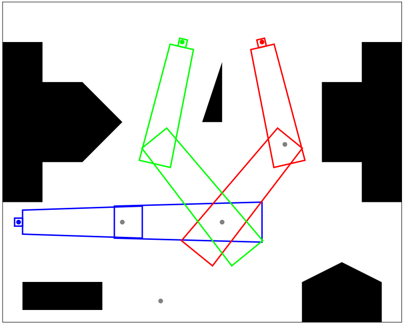
Free space $\mathtt{C}_{\mathtt{free}}$ the complement of Obstacle space

Path-planning → Search for a path for the reference point of the artifact in the free space
Configurations of the artifact in free space have no intersection with obstacles

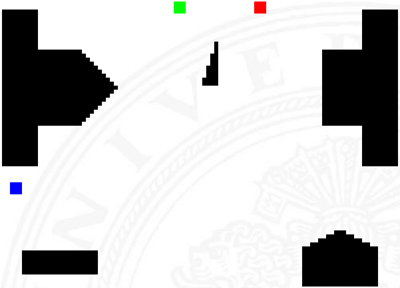
Work Space to Configuration Space – Illustration



Work Space to Configuration Space – Example

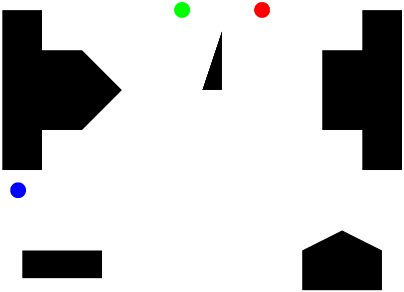


Workspace scheme with start and goal positions

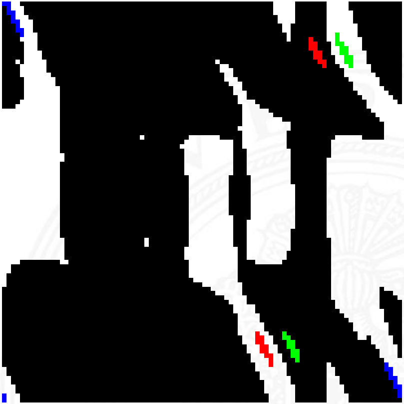


Discretized workspace
$xscale = 100,\ y^{scale} = 80$

Work Space to Configuration Space – Example

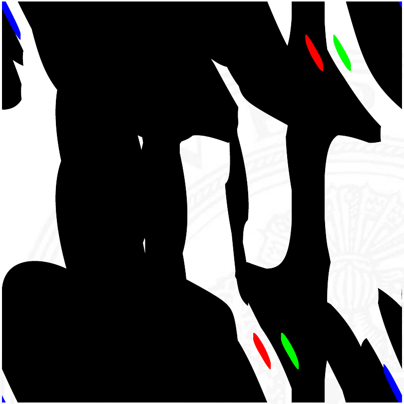


Discretized workspace $x^{scale} = 2000$, $yscale = 1600$



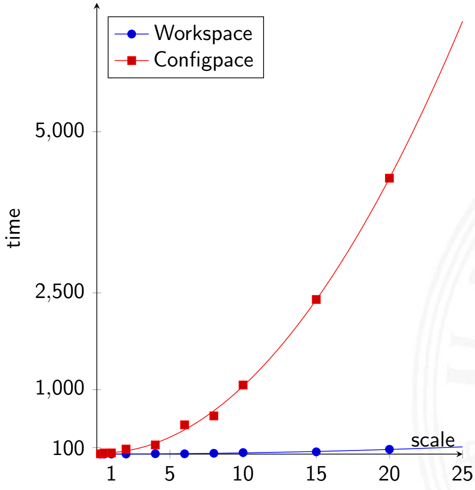
Discretized configuration space $q_1^{scale} = 90$, $q_2^{scale} = 90$

Work Space to Configuration Space – Example



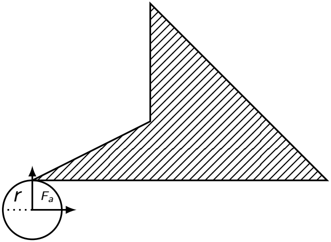
Discretized configuration space $q_1^{scale} = 3600$, $q_2^{scale} = 3600$

Work Space to Configuration Space – Complexity

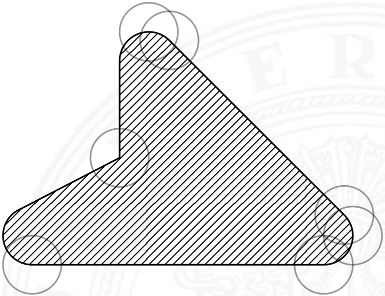


- Python
- Brute forward kinematics
- using polygon collisions
  - shapely library
- 24 cpus
  - Intel Xeon E5-2420 (1.90GHz)

C-Obstacle for a circular artifact

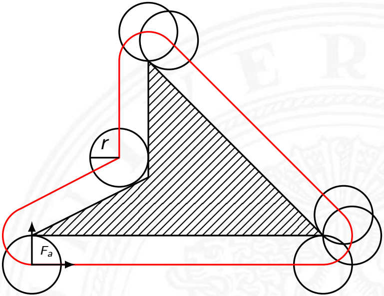


Obstacle & circular artifact of radius r



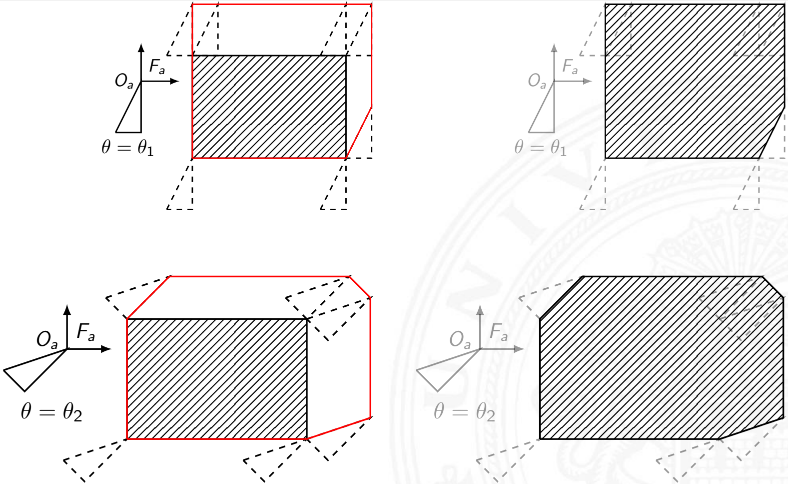
Expanded C-Obstacle

C-Obstacle for a circular artifact



Path of minimal distance to obstacle

C-Obstacle for Polygons



Obstacle & polygon artifact of with $\theta = \theta_1 \vee \theta_2$; minimum distance to obstacle.

Minkowski Sum & Difference

A C-Obstacle of a fixed, convex obstacle with respect to a moving convex robot (part) my be theoretically represented as the `Minkowski Difference` of the corresponding objects.

$C_O(H)$ is the C-obstacle of a fixed convex polyhedra H, with respect to the (moving) convex object O.

Minkowski-Sum (Minkowski-Difference) of H and O (H and $-O$)

$$C_O(H) = H \ominus O = H \oplus (\ominus O)$$

where

$$H \ominus O := \{h - o \mid h \in H \wedge o \in O\}$$

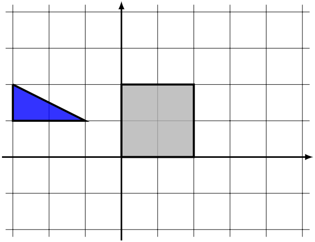
Minkowski Sum & Difference – 2D Example

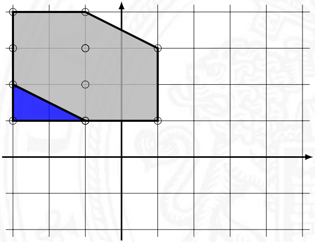
$$A = \{(0,0),(2,0),(2,2),(0,2)\} \qquad B = \{(-1,1),(-3,2),(-3,1)\}$$

$$A \oplus B = \{(-1,1),(-3,2),(-3,1),(1,1),(-1,2),(-1,1),$$
$$(1,3),(-1,4),(-1,3),(-1,3),(-3,4),(-3,3)\}$$

The convex hull (eliminating duplicates & inner points)

$$conv\{A \oplus B\} = \{(-3,1),(1,1),(1,3),(-1,4),(-3,4)\}$$





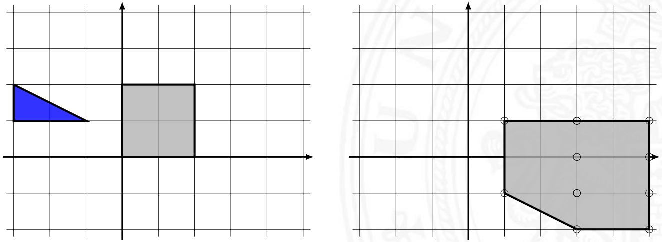
Minkowski Sum & Difference – 2D Example (cont.)

$A = \{(0,0),(2,0),(2,2),(0,2)\} \qquad B = \{(-1,1),(-3,2),(-3,1)\}$

$$A \ominus B = \{(1,-1),(3,-2),(3,-1),(3,-1),(5,-2),(5,-1),(3,1),(5,0),(5,1),(1,1),(3,0),(3,1)\}$$

The convex hull (eliminating duplicates & inner points)

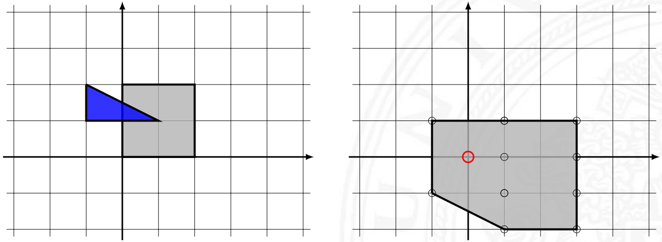
$$conv\{A \ominus B\} = \{(1,-1),(3,-2),(5,-2),(5,1),(1,1)\}$$



Minkowski Sum & Difference – 2D Example (cont.)

Collision detection

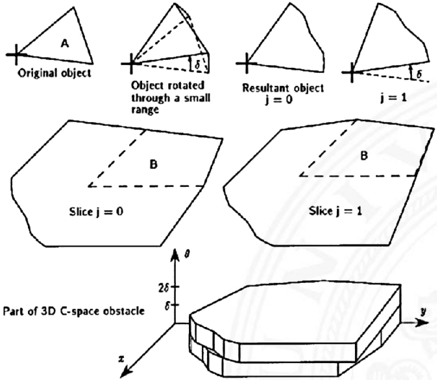
Two objects are colliding, if their Minkoswki difference contains the origin of the coordinate frame.



There is an interactive applet on the web:

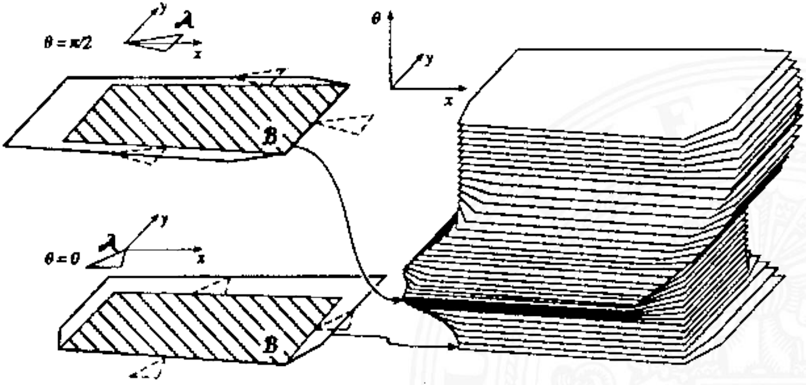
`http://www.cut-the-knot.org/Curriculum/Geometry/PolyAddition.shtml`

C-Obstacles for 2-D translation and 1-D rotation



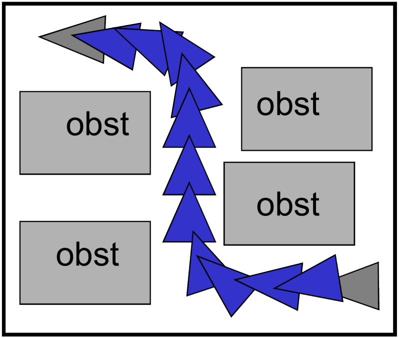
Represent rotational configuration of the C-obstacle as slice for each θ configuration of the robot.

C-Obstacles for 2-D translation and 1-D rotation (cont.)

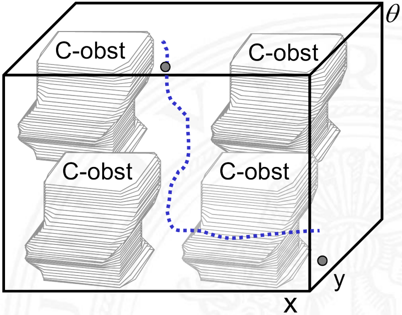


The configuration space for a k-DOF robot is a k-Dimensional coordinate system.

C-Obstacles for 2-D translation and 1-D rotation (cont.)

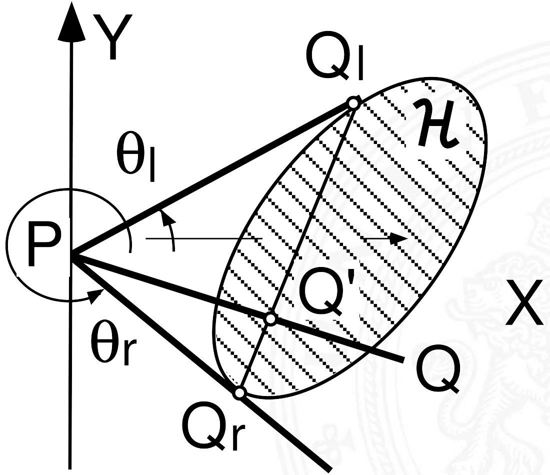


Work space (x, y)

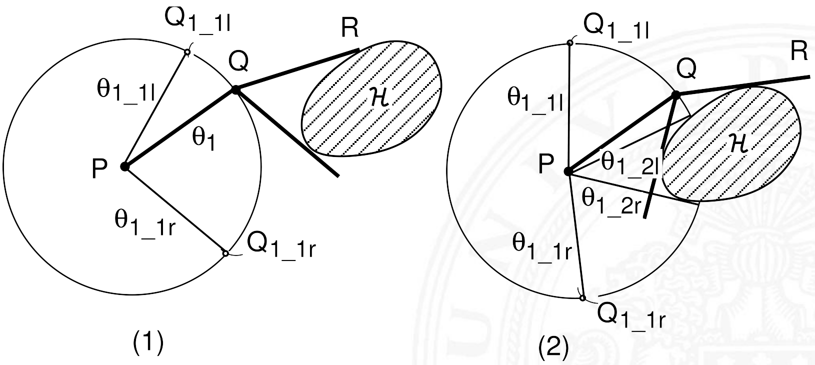


Configuration space (x, y, θ)

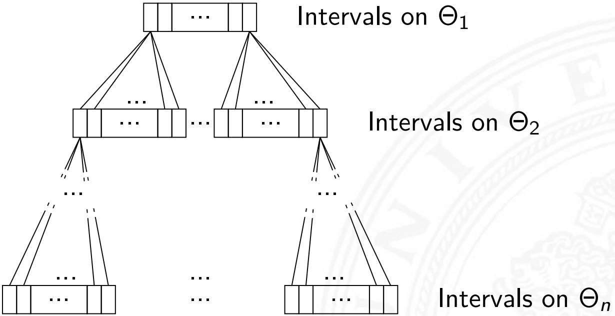
Calculation of the C-obstacles of a pole



Calculation of the C-obstacles of a 2-DOF chain of poles

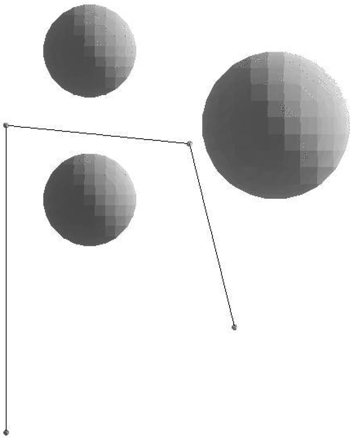


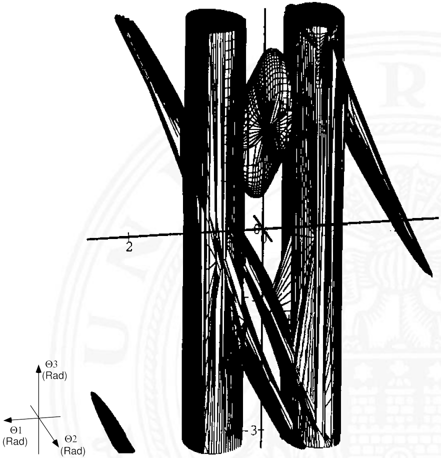
Tree-structure for the partition of the configuration space



Curse of dimensionality

Configuration space of a 3-DOF chain of poles





Partition Representation of C-Space

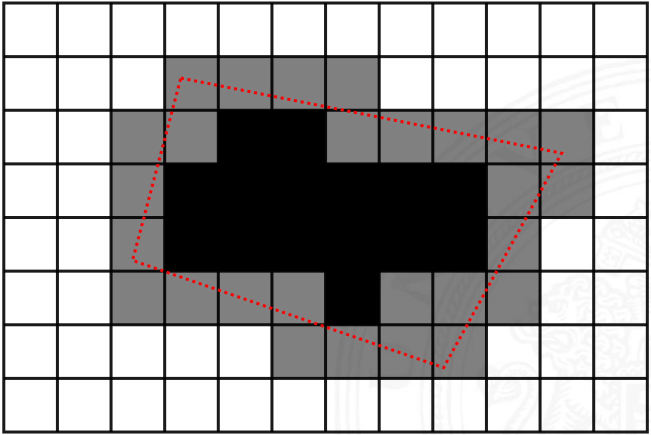
The free space is partioned into cells using

- Geometrical partition
  - uniform cubes
  - a hierarchical tree-structure (Quad-tree, Oct-tree, etc.)
  - slices and scanlines
  - `bubbles` of variable size

  The union of the non-overlapping cells is part of the free space.
  Neighborship graphs represent the connectivity of free space.
- Topological partition
  - overlapping generalized cones
  - critical points of the C-obstacle connection graph

  The union of the overlapping cells is equal to the free space.

Partitioning of Configuration Space Using Squares

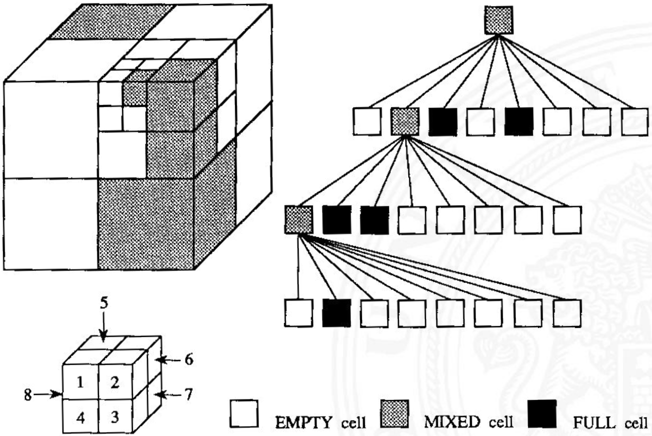


Resulting bitmap of configuration space using squares partitioning

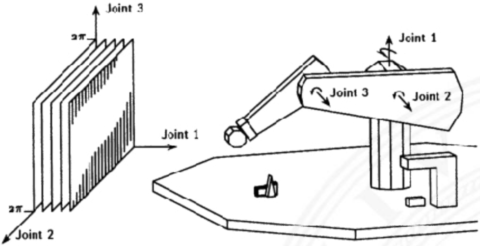
Partitioning of Configuration Space Using Squares (cont.)

Bitmap of configuration space

Partitioning of the configuration space using Octrees



Partitioning of the configuration space using Slices

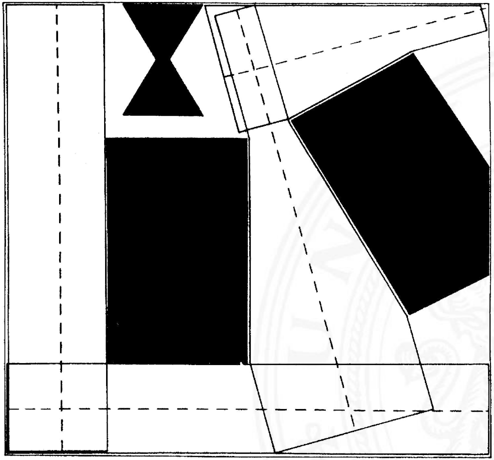


Complexity regarding the transformation of the C-obstacles

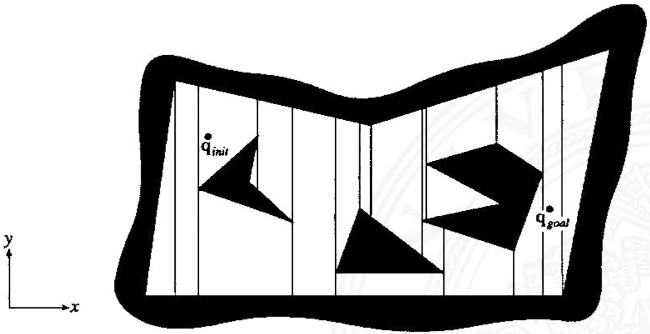
$$r^{d-1}f(m)$$

where r: the number of discretization steps for each DOF,
d: DOF of the robot arm
$f(m)$: the computing time of one slice
m: the number of edges of all obstacles

Representation of free space with generalized cones

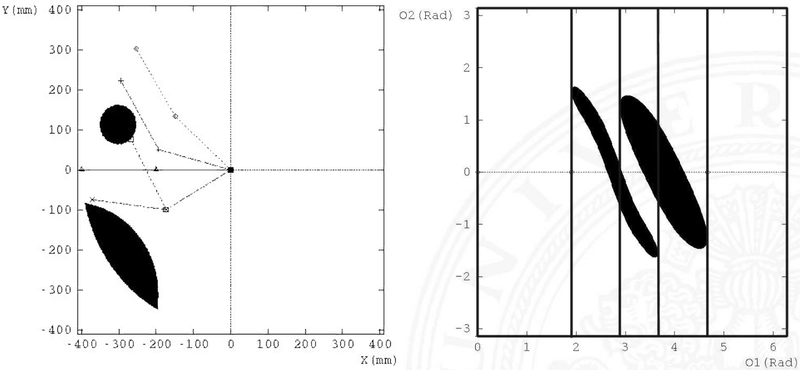


Exact Partition of Configuration Space



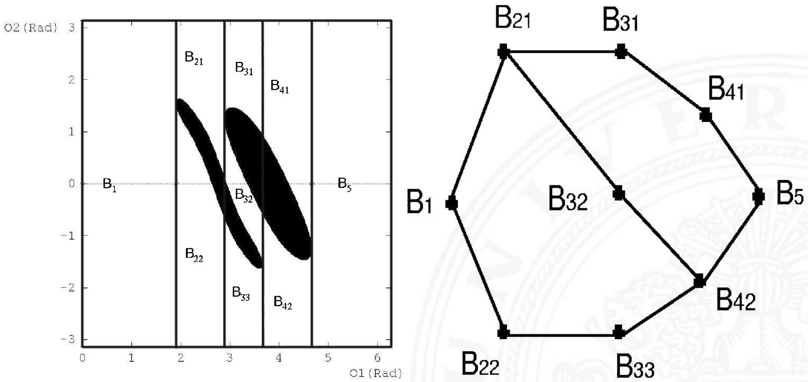
Trapezoidal partitioning of the configuration space

Exact Partition of Configuration Space (cont.)



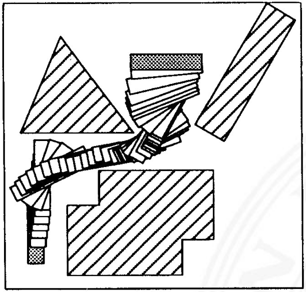
Cylindrical partitioning using critical points

Exact Partition of Configuration Space (cont.)



Cylindrical partitioning and connectivity graph

Planning Results



[13]

`Piano movers problem:` 3-DOF configuration space Serial computing: 3-DOF C-space
Massive-parallel computing: up to 6-DOF C-Space

Partition based Path Planning

Advantages:

- Complete in case of sufficient resolution
- Global overview

Disadvantages:

- High demand for RAM
  - `Curse of Dimensionality`
- Complex to implement
- Practically implementable only for few degrees of freedom

$$\Downarrow$$

Path planning without explicit representation of free space?

$\Longrightarrow$ next Lecture!

[1] K. Fu, R. González, and C. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*.
McGraw-Hill series in CAD/CAM robotics and computer vision, McGraw-Hill, 1987.

[2] R. Paul, *Robot Manipulators: Mathematics, Programming, and Control : the Computer Control of Robot Manipulators*.
Artificial Intelligence Series, MIT Press, 1981.

[3] J. Craig, *Introduction to Robotics: Pearson New International Edition: Mechanics and Control*.
Always learning, Pearson Education, Limited, 2013.

[4] J. F. Engelberger, *Robotics in service*.
MIT Press, 1989.

[5] W. Böhm, G. Farin, and J. Kahmann, "A Survey of Curve and Surface Methods in CAGD," *Comput. Aided Geom. Des.*, vol. 1, pp. 1–60, July 1984.

[6] J. Zhang and A. Knoll, "Constructing fuzzy controllers with B-spline models-principles and applications," *International Journal of Intelligent Systems*, vol. 13, no. 2-3, pp. 257–285, 1998.

[7] M. Eck and H. Hoppe, "Automatic reconstruction of b-spline surfaces of arbitrary topological type," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, (New York, NY, USA), pp. 325–334, ACM, 1996.

[8] M. C. Ferch, *Lernen von Montagestrategien in einer verteilten Multiroboterumgebung*.
PhD thesis, Bielefeld University, 2001.

[9] N. J. Nilsson, "A mobile automaton: An application of artificial intelligence techniques," tech. rep., DTIC Document, 1969.

[10] J. H. Reif, "Complexity of the mover's problem and generalizations extended abstract," *Proceedings of the 20th Annual IEEE*

Conference on Foundations of Computer Science, pp. 421–427, 1979.

[11] J. T. Schwartz and M. Sharir, "A survey of motion planning and related geometric algorithms," *Artificial Intelligence*, vol. 37, no. 1, pp. 157–169, 1988.

[12] J. Canny, *The complexity of robot motion planning*. MIT press, 1988.

[13] T. Lozano-Pérez, J. L. Jones, P. A. O'Donnell, and E. Mazer, *Handey: A Robot Task Planner*. Cambridge, MA, USA: MIT Press, 1992.