IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

WEEK 7: ROBOTICS

PROF. (DR.) DILIP KUMAR PRATIHAR

MECHANICAL ENGINEERING DEPARTMENT, IIT KHARAGPUR

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Topic 7: Robot Vision

PROF. (DR.) DILIP KUMAR PRATIHAR
MECHANICAL ENGINEERING DEPARTMENT, IIT KHARAGPUR

Methods of Pre-Processing

a. Masking

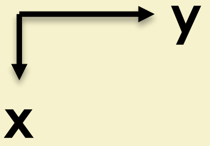
$$P(x, y) = O[f(x, y)]$$

Pre-processed intensity

operator

Input intensity

f(x,y) : Light intensity value at pixel Q



f(x-1,y-1)	f(x-1,y)	f(x-1,y+1)
f(x,y-1)	Q: f(x,y)	f(x,y+1)
f(x+1,y-1)	f(x+1,y)	f(x+1,y+1)

Let us consider the pixel Q having the coordinates (x,y). It has two horizontal and two vertical and four diagonal neighbors.

Let us consider a 3x3 mask with coefficients W_1, W_2, W_3, W_4, W_5, W_6, W_7, W_8, and W_9.

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

-1	-1	-1
-1	+8	-1
-1	-1	-1

Example of a 3x3 mask

$$
\begin{aligned}
P(x,y) = O[f(x,y)] \\
&= W_1 f(x-1,y-1) + W_2 f(x-1,y) + W_3 f(x-1,y+1) \\
&+ W_4 f(x,y-1) + W_5 f(x,y) + W_6 f(x,y+1) + W_7 f(x+1,y-1) \\
&+ W_8 f(x+1,y) + W_9 f(x+1,y+1)
\end{aligned}
$$

b. Neighborhood Averaging

Here, p(x,y) is calculated by averaging the intensity values of the pixels contained in a pre-defined neighborhood of f(x,y).

$$p(x, y) = \frac{1}{R} \sum_{(n,m) \in S} f(n, m)$$

Where, S is the set of pixels lying in the neighbourhood of (x,y) including itself and R is the total number of neighbourhood pixels including itself.

c. Median Filtering

To determine pre-processed light intensity value of a pixel Q, we consider light intensity values of all its neighboring pixels including itself. We sort light intensity values in the ascending order say, and then determine the median value. This median value is going to replace the intensity value at Q.

Step 5: Thresholding

- **To get clear distinction between objects and the background, let T be the threshold intensity**

$$g(x,y) = \begin{cases} 1, & if\ p(x,y) > T \\ 0, & if\ p(x,y) \le T \end{cases}$$

$$\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & & Y \\ 0 & 0 & 1 & 1 & 1 & 0 & & \\ 0 & 0 & 1 & 1 & 0 & 0 & & \\ 0 & 0 & 1 & 1 & 0 & 0 & & \\ 0 & 0 & 1 & 0 & 0 & 0 & & \\ X & & & & & & & \end{array}$$

For the black background and white object, 1 corresponds to object and 0 indicates the background.

Step 6: Edge detection

- To detect the edge of an object
- Gradient operator

$$G[p(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial p / \partial x \\ \partial p / \partial y \end{bmatrix}$$

Masks used for Gradient operator

-1	-2	-1
0	0	0
1	2	1

G_x

-1	0	1
-2	0	2
-1	0	1

G_y

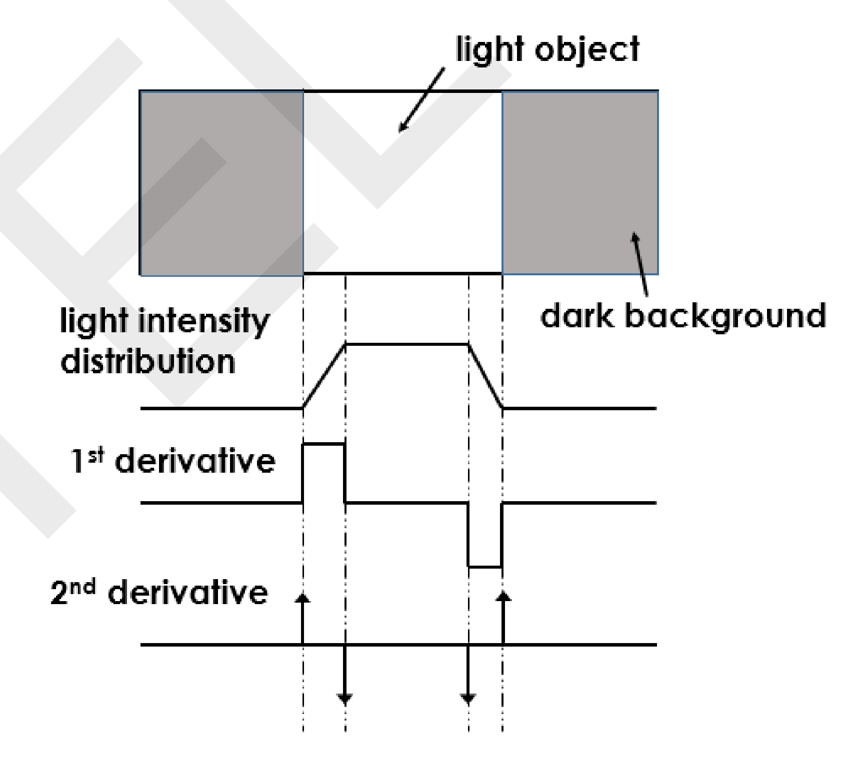
- **Laplace operator**

$$L[p(x,y)] = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2}$$

0	1	0
1	-4	1
0	1	0

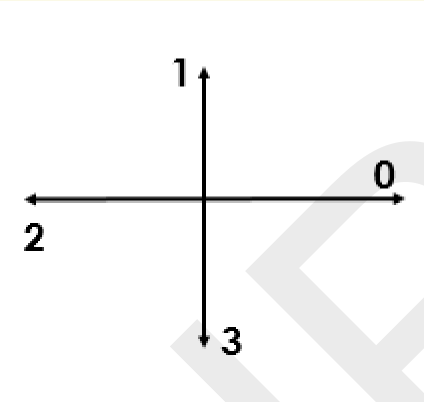
Edge Detection (contd.)

- **Light object on dark background**

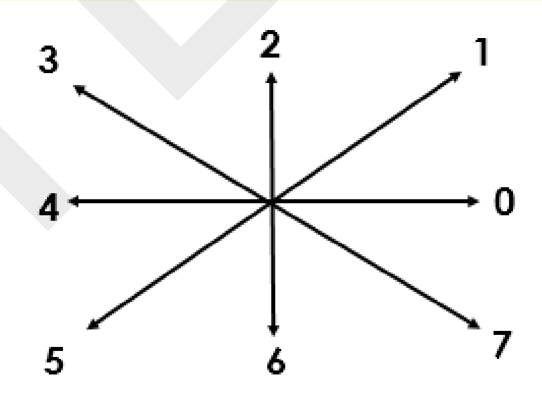


Boundary Descriptors

Chain Codes – used to represent the boundary of an object by a set of straight line segments of specified length and direction.

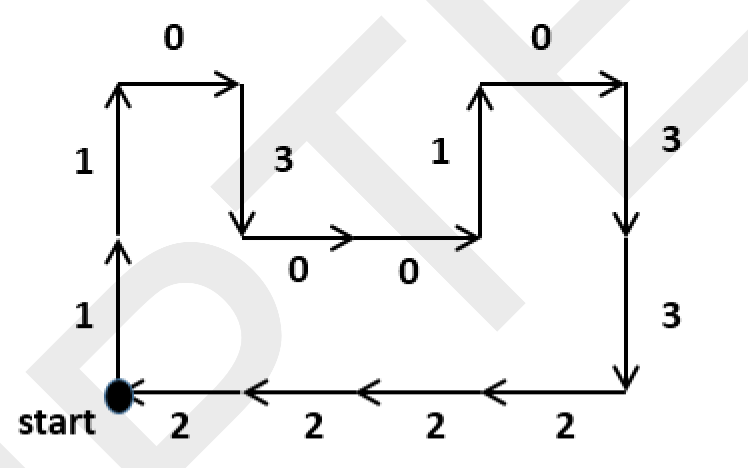


4-directional chain code



8-directional chain code

Example:

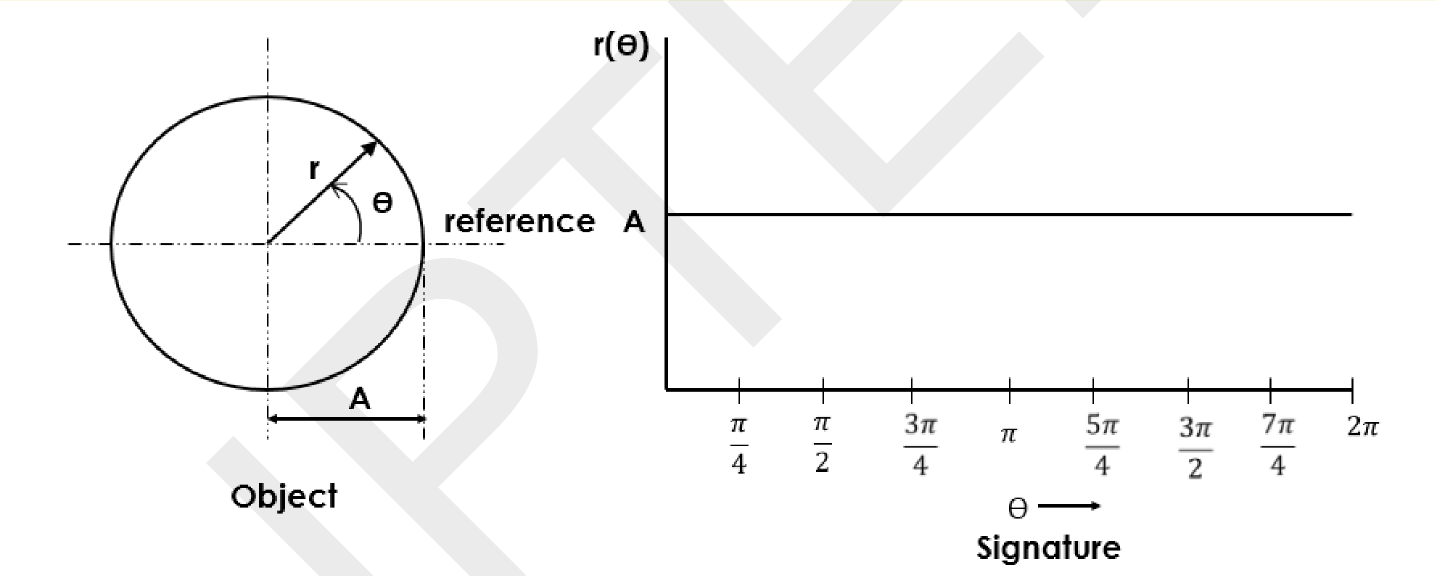


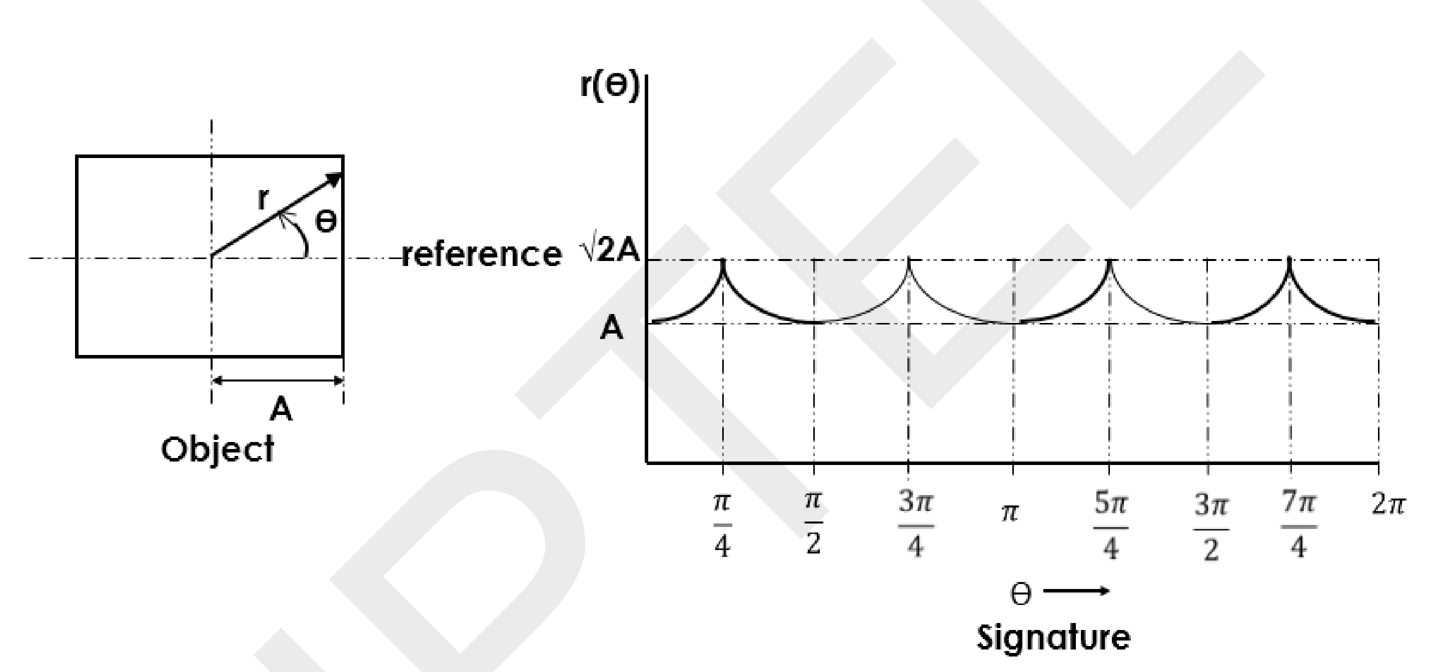
Chain code: 11030010332222

Signature

- One dimensional functional representation of a boundary

Examples:





Note: To identify multiple objects present in an image, we consider *Compactness = (perimeter^2/ area)*

Topic 8: Robot Motion Planning

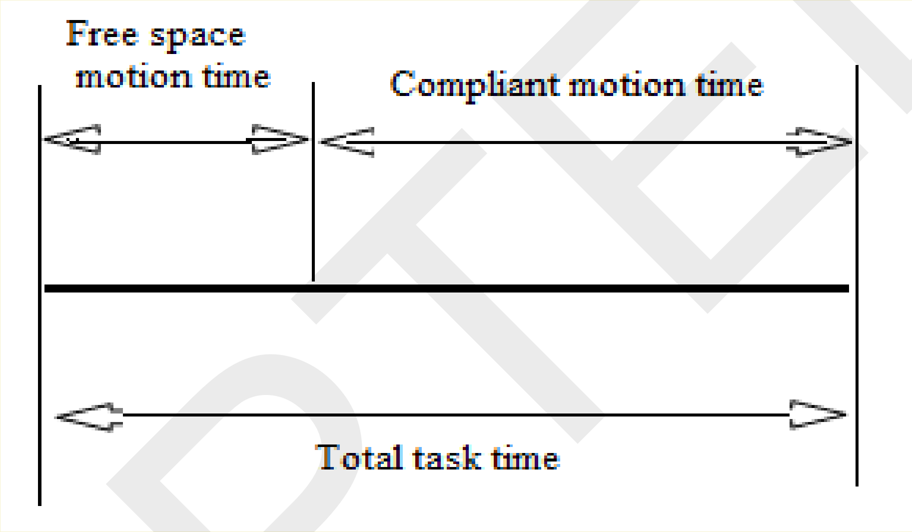
PROF. (DR.) DILIP KUMAR PRATIHAR
MECHANICAL ENGINEERING DEPARTMENT, IIT KHARAGPUR

Aim

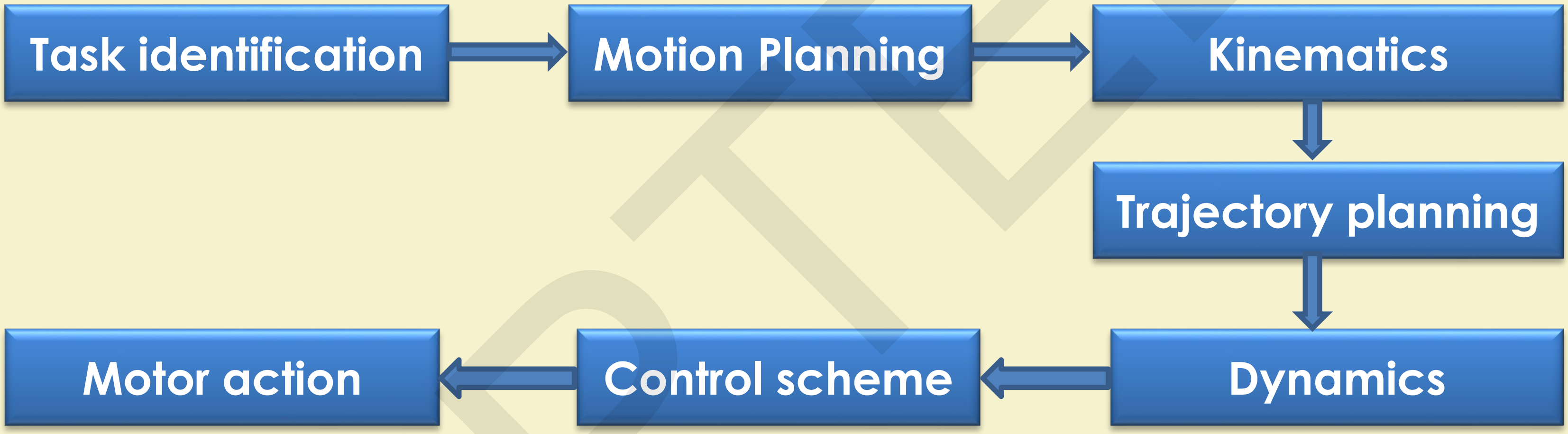
To determine the course of action/path while moving from an initial position to a final position.

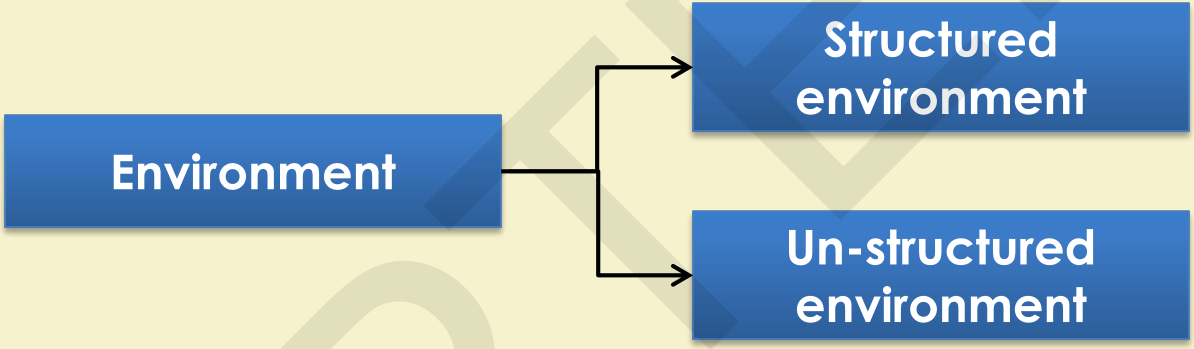
Robot Motion Planning

- Gross motion planning / Free space motion planning
  - Manipulation
  - Navigation
- Fine motion planning / Compliant motion planning


Free space
motion time
Compliant motion time
Total task time

Sequence of Robotic Action




Environment
Structured environment
Un-structured environment

Motion Planning Approaches

Global Approach/ Act-After-Thinking process/ Off-line planning (motion planning with complete information)

Local Approach/ Act-While-Thinking process/ On-line planning (motion planning with incomplete information)

Motion Planning Schemes

Traditional schemes/ Algorithmic Approaches

Graph-based methods

(i) Visibility graph
(ii) Voronoi diagram
(iii) Cell decomposition
(iv) Tangent graph
(v) Accessibility graph

Analytical Approaches

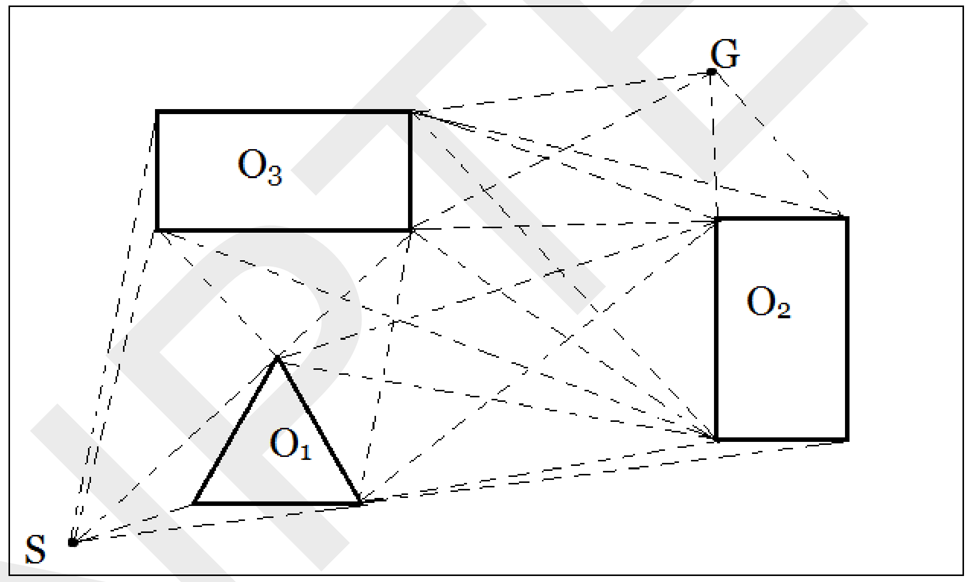
(i) Potential Field Approach
(ii) Path Velocity Decomposition
(iii) Incremental Planning
(iv) Probabilistic Approach
(v) Relative Velocity Approach
(vi) Reactive Control Strategies (Behavior-Based Robotics)

Non-Traditional schemes (using soft computing)

(i) Fuzzy logic – based
(ii) Neural networks – based approaches

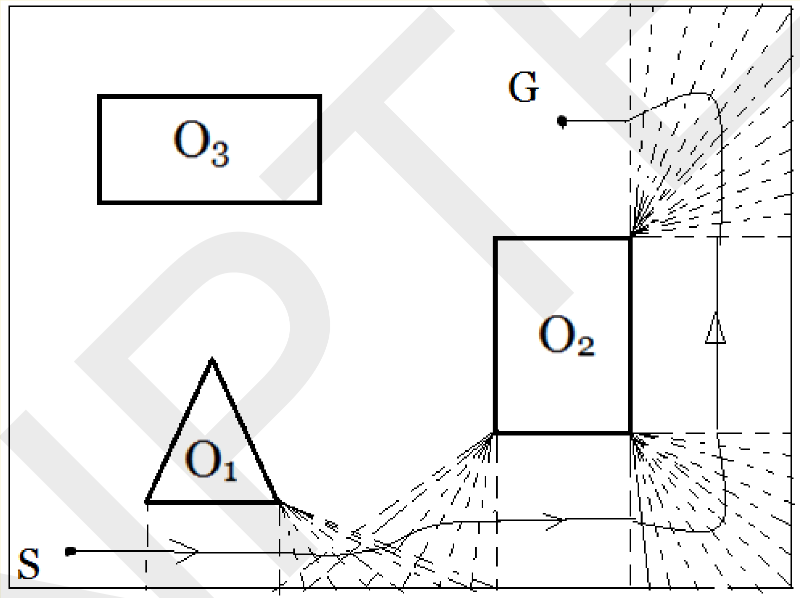
Visibility Graph (Nilsson 1969)

- It connects those vertices of obstacles, which are visible from one another.

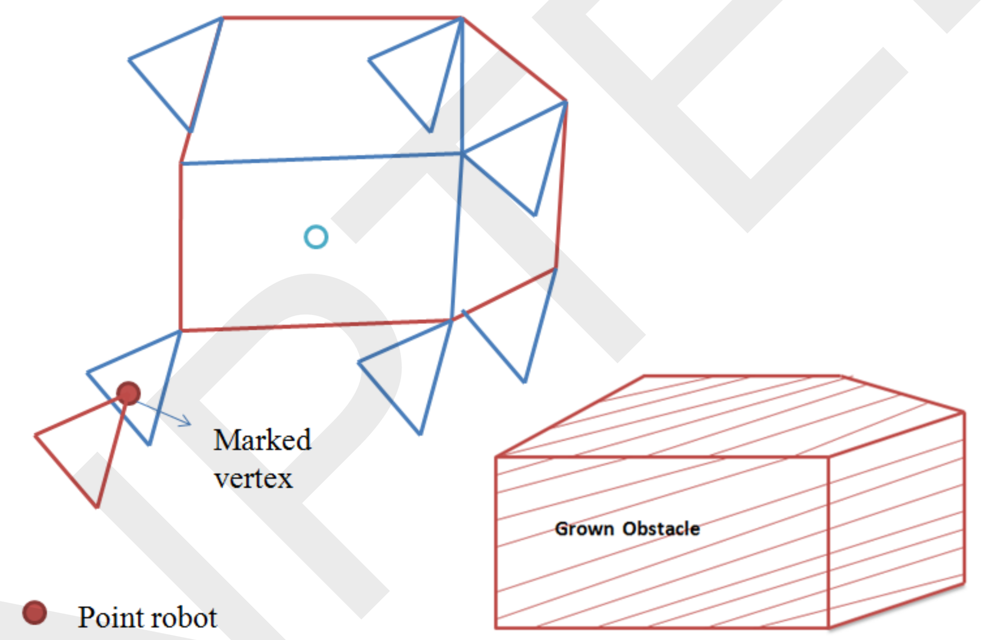


Voronoi Diagram (Dunlaing et al., 1986)

- Represents the locus of points those are equidistant from at least two of the boundaries (obstacle).

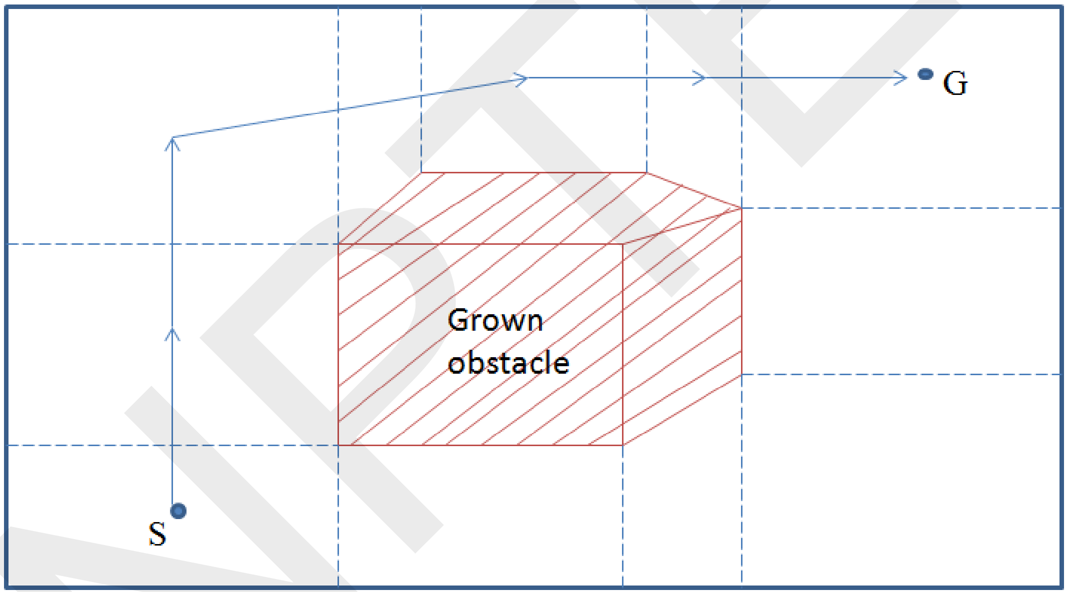


Cell Decomposition (Lozano Perez, 1983)
Configuration space (C-space)



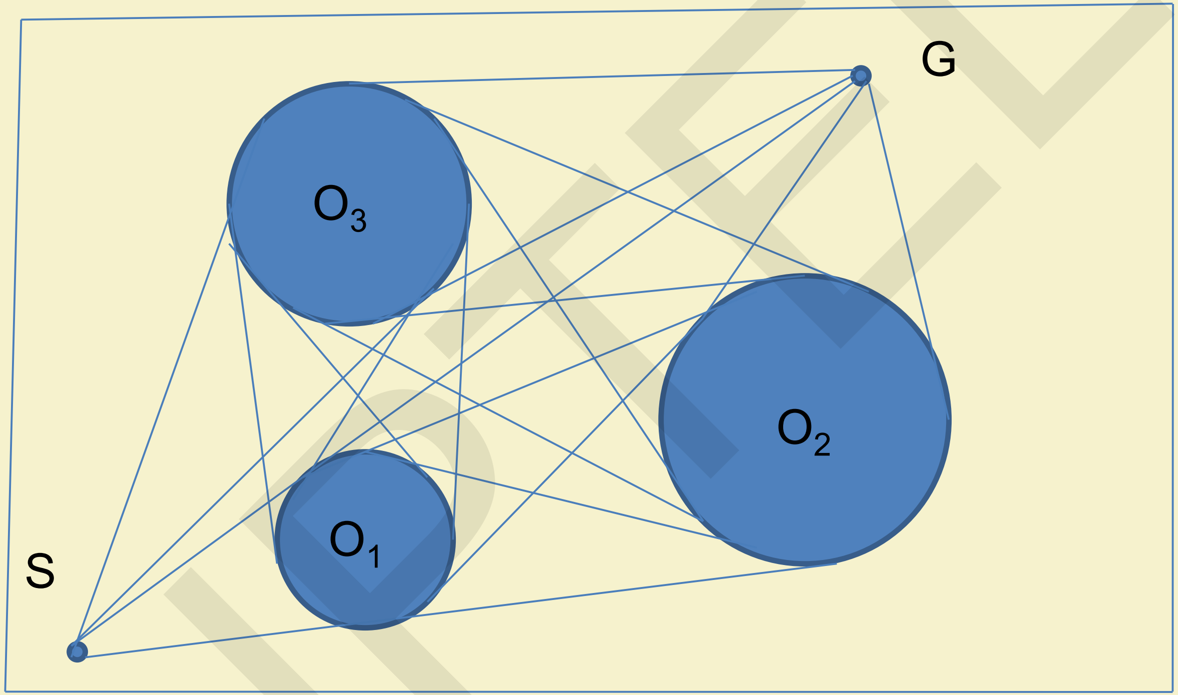
Cell Decomposition (contd.)

- Robot's free area is divided into a number of small regions called cells. A connectivity graph is then constructed and searched.



Tangent Graph (Liu & Arimoto 1991)

- Tangents are drawn from the starting point to the visible obstacle and then from one obstacle to another
- A path comprises of tangents and circular arcs
- Complexity: O (N^2), where N is the number of control points


G
O_3
O_2
O_1
S

Approaches to Solve Moving Obstacle Problems

Path Velocity Decomposition (Kant and Zucker, 1984)

This problem is decomposed into two sub-problems as follows:

(i) Path planning problem (PPP) – to plan a path to avoid collision with static obstacles

(ii) Velocity planning problem (VPP) – to plan the velocity of the robot along the above planned path to avoid collision with moving obstacles

Drawbacks

- The path may not be always good, particularly when there are many obstacles
- As there is a sudden change of velocity of the robot, it will have jerky motion, which is not desirable.

Accessibility Graph (Fujimura and samet, 1988)

- Generalization of the visibility graph
- At a particular instant of time, motion planning problem in dynamic environment is converted into find-path problem, which is solved using visibility graph
- In dynamic environment, visibility graph will go on changing.

Incremental Planning Scheme (Slack & Miller, 1987)

Plan a collision – free path based on predicted position of obstacles

↓

Follow the path until it becomes invalid (validity is checked based on the data collected using sensor(s) / camera(s)

↓

Adjust the predictions & plan a new collision – free path

↓

Follow the path until it becomes invalid

Continue until the robot reaches its destination

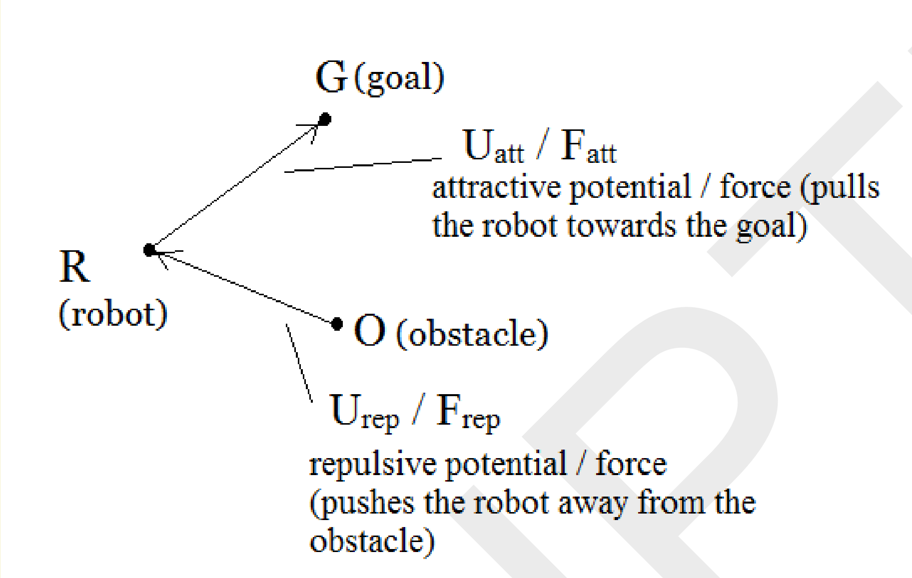
Relative Velocity Scheme

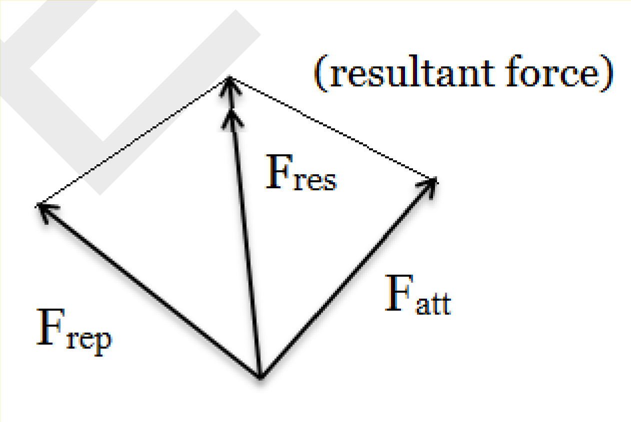
- Consider relative velocity of the robot with respect to obstacles
- Dynamic motion planning problem is converted into several static problems

- Several static problems are then converted into a single problem by means of a vector transformation

- Set of velocity vectors are then computed, so that the robot avoids collision with all the moving obstacles

Potential Field Approach (Khatib, 1986)

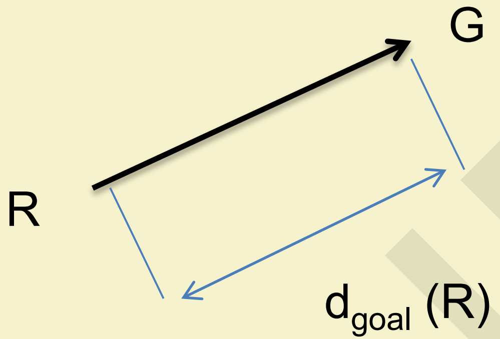


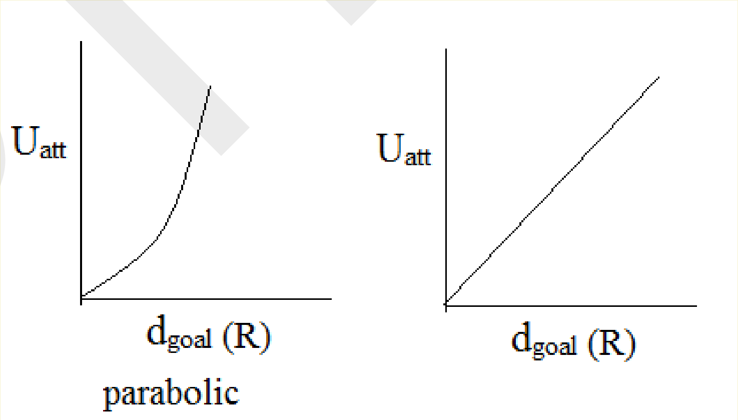


- Speed of the robot α F_{res}
- Direction of movement of the robot is along the direction of resultant force

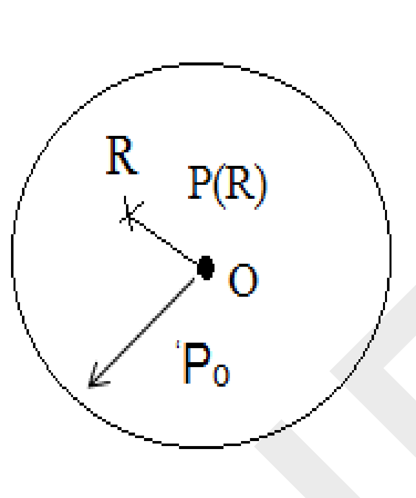
Attractive Potential

$$U_{att}(R) = \frac{1}{2}\xi d_{goal}^{2}(R), \text{ if parabolic}$$
$$= \xi d_{goal}(R), \text{ if conic-well}$$

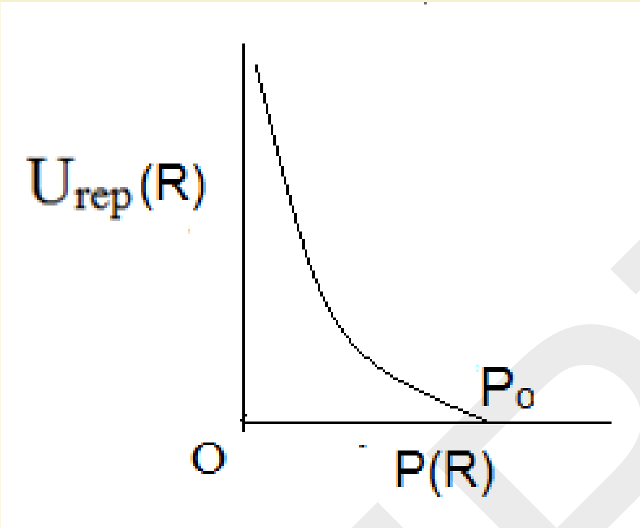




Repulsive Potential



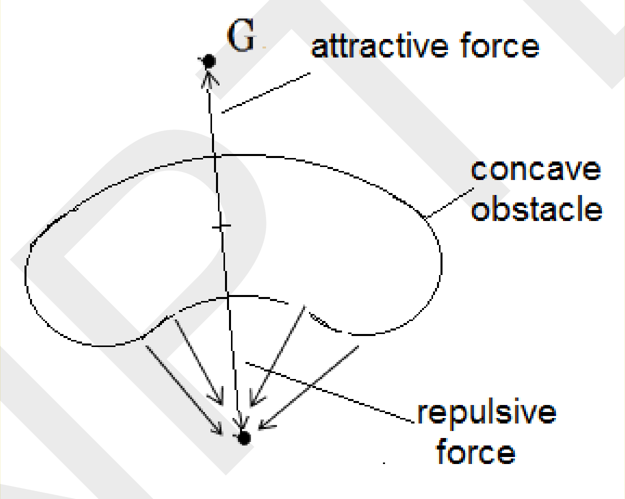
$$U_{rep}(R) = 0$$



$$U_{rep}(R) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{P(R)} - \frac{1}{P_o}\right)^2, & \text{if } P(R) < P_o \\ 0, & \text{if } P(R) \geq P_o \end{cases}$$

Drawbacks

- Solution depends on the chosen potential function
- Chance of local minima problem – when the attractive force is balanced by the repulsive force



- **When the robot travels in a narrow corridor, it experiences repulsive forces simultaneously from the opposite sides, and consequently the motion becomes unstable**

- **Unable to find a path among closely spaced obstacles**

Reactive Control Strategy (Brooks, 1986)

- Robotic action is decomposed into some independent primitive behaviours like move-to-goal, avoid-obstacle, etc.
- Basic behaviours are controlled at different layers of control architecture
- Basic behaviours are coordinated by a central mechanism (Behaviour-Based Robotics)

Drawbacks

- The behaviours are hard-wired, thus it is unable to handle behaviours which the programmer did not foresee beforehand

- The number of layers increases with the complexity of the problem. It requires a large amount of computer memory

Computational Complexity

- **Canny and Reif (1987) –**

  Motion planning for a point robot among moving obstacles in 2D plane with bounded velocity is NP-hard.

- **Reif and Sharir (1985) –**

a) Motion planning among moving obstacles in 3-D space without velocity bound is NP-hard

b) Motion planning among moving obstacles in 3-D space with velocity bound is PSPACE-hard

Drawbacks of the Traditional Methods of Motion Planning

- Traditional methods are computationally expensive even for a simple problem
- No versatile algorithm, which is applicable to all the problems
- As most of the algorithms do not have an optimization module, the generated path may not be optimal in any sense.

So, there is still a need for the development of an efficient, versatile and computationally tractable algorithm for solving the motion planning problems of robots.