



## Problem

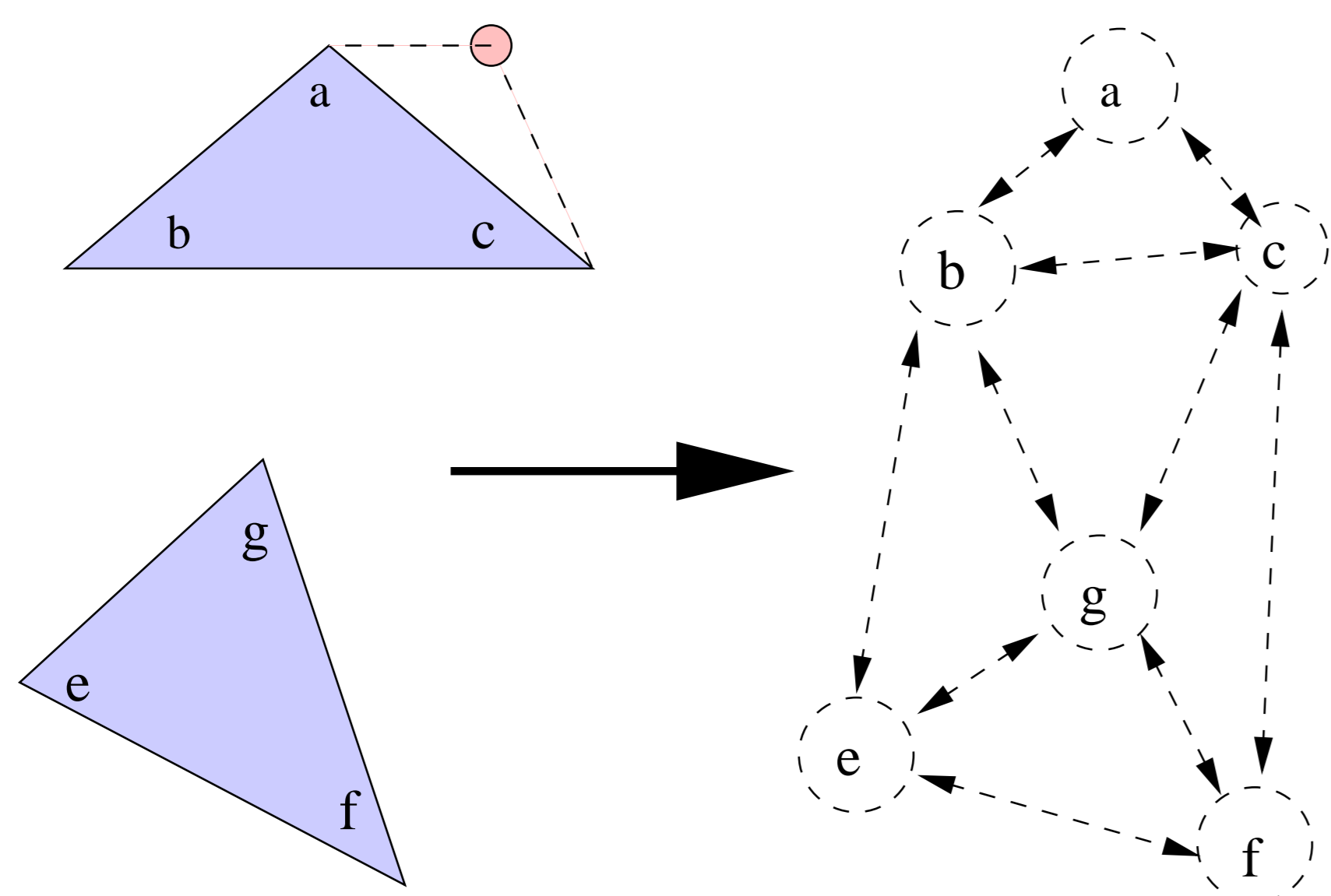
A robot explores a 2D environment, and builds a topological map that allows it to navigate along shortest paths.

- Robot is equipped with a *metric gap sensor*, a sensor which detects and reports depth discontinuities in all directions around the robot
- Odometry is not used.
- World contains polygonal obstacles and may have a polygonal boundary

## Motivation

- Autonomous map generation is useful for mobile robots
- Topological Maps
  - Provide compact representation
  - Record details used for navigation
  - Simplify path planning
- Gap Sensing
  - Can provide information about visibility constraints

## Example

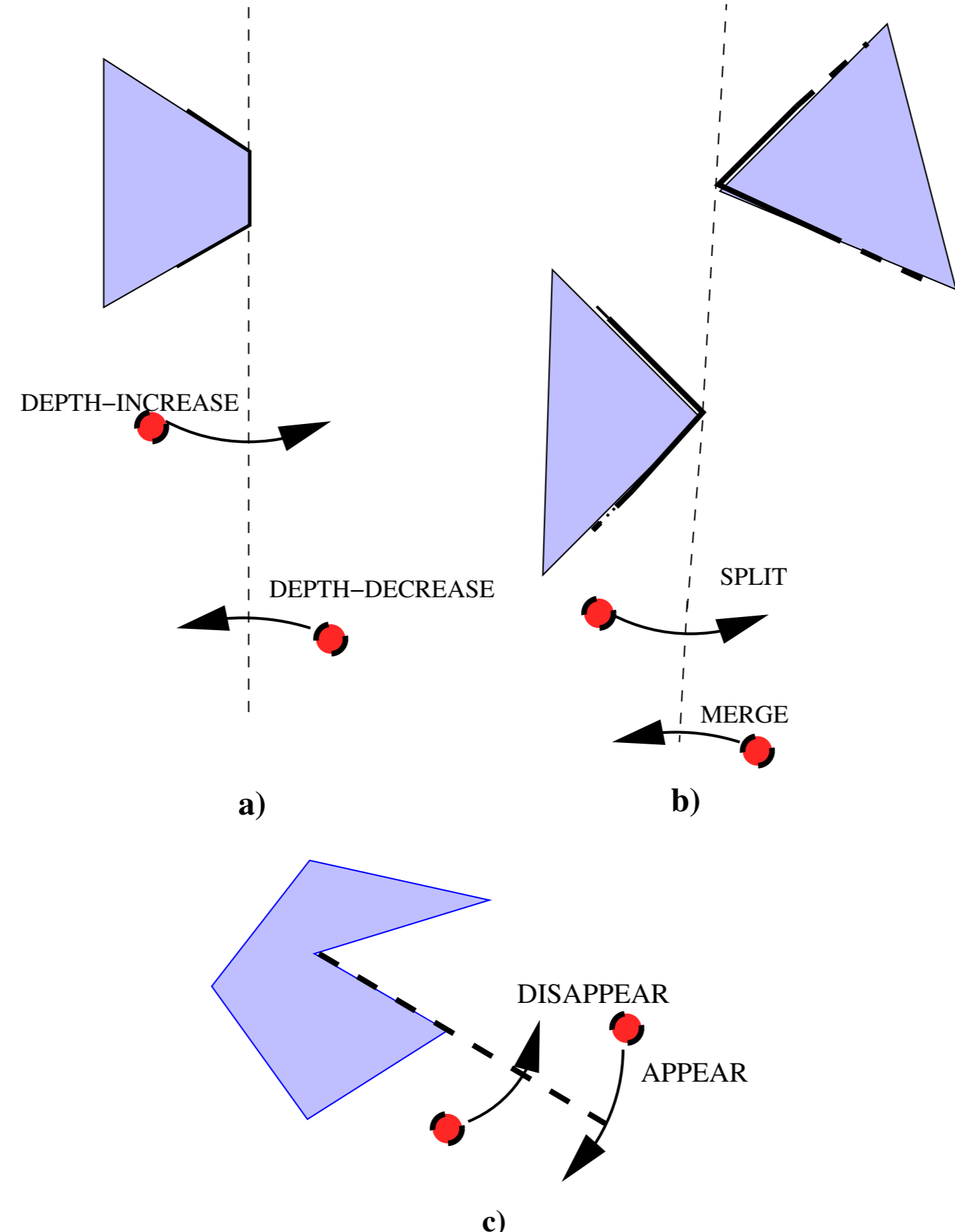


## Gap Detection

- A *gap sensor* takes a 360 degree scan around the robot and reports discontinuities in distance
- These discontinuities are called *gaps* and are classified by whether the discontinuity is from near to far or far to near
- A gap implies that there is some portion of the world is hidden from the robot
- Robot can track gaps and detect events as it moves
- *Metric gap sensor* also reports distance to closer portion of the gap

## Gap Events

Six types of events:  
Split, Merge, Depth-Increase,  
Depth-Decrease Appear, Disappear



## Gap Event Information

- In polygonal worlds, a gap is always caused by a vertex
- A gap event occurs when the robot crosses an extended bitangent or non-convex edge
- *Merge*, *Split*, *Depth-increase*, *Depth-decrease* events imply that two vertices are visible to each other
- Each gap visible from a vertex represents a neighboring vertex in the map
- Map is a superset of the reduced visibility graph and a subset of the visibility graph

## Gap Classification

Handedness of Gaps

- gap is a *left gap* if obscured portion is on the left side
- *right gap* if on right side

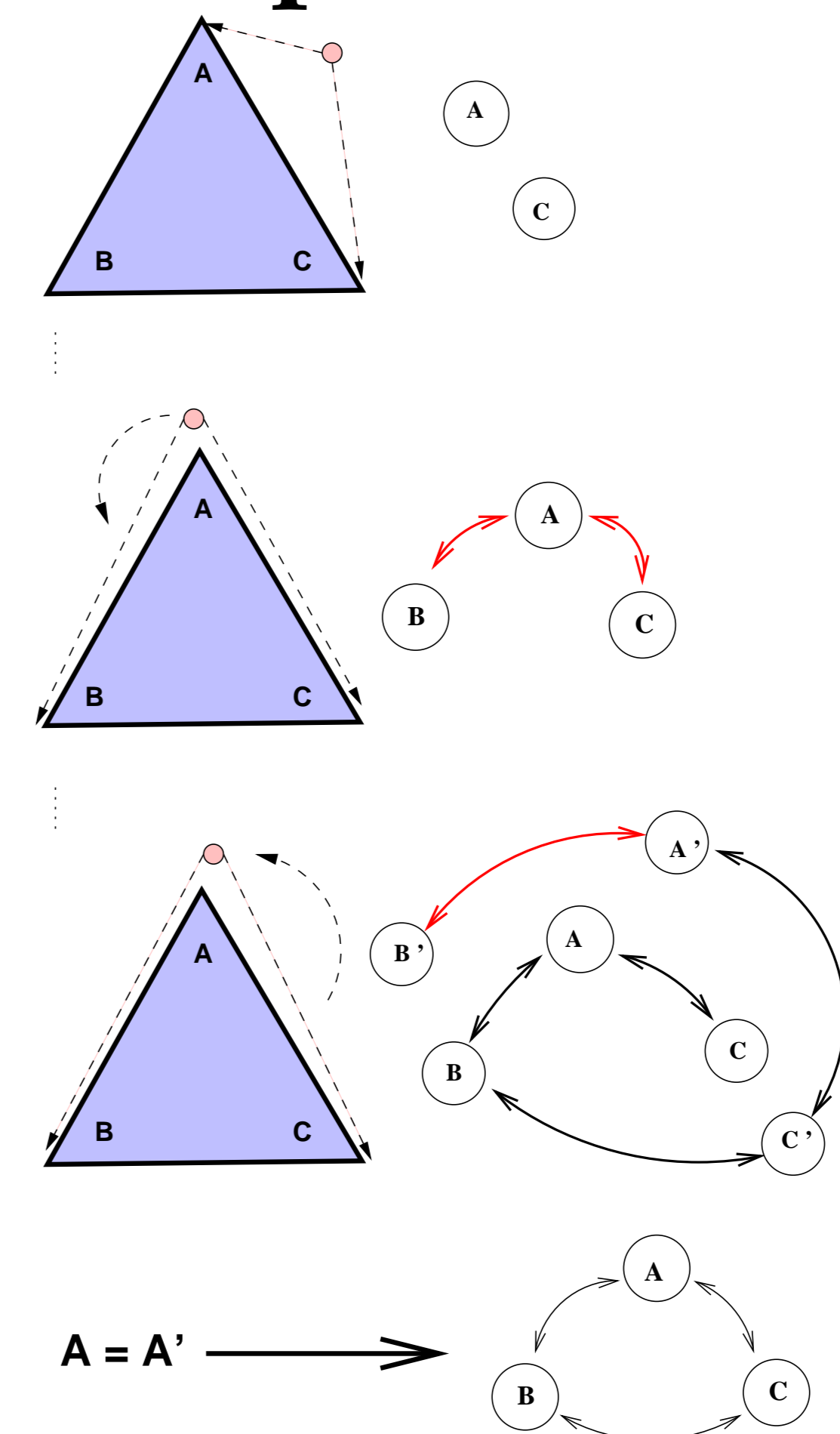
## Robot model

- Robot can follow a boundary at some  $\epsilon$  distance
- Robot can drop and recover a marker to determine when it has circumnavigated an obstacle
- The robot has the ability to *chase a gap*
- Robot can reach a vertex if gap distance goes to  $\epsilon$  and then increases discontinuously or if gap disappears
- Reaching a vertex generates an event
- Robot leaves a vertex if there is a depth decrease to  $\epsilon$ , or if gap reappears

## Uniqueness Problem

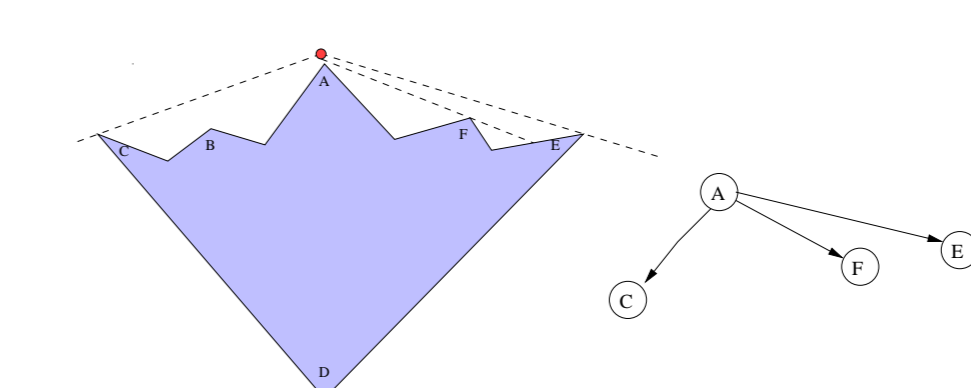
- The robot observes a vertex on another obstacle
- The vertex disappears as the robot circumnavigates an obstacle and later reappears
- How does the robot know that the two vertices are the same?
- Recording gap events provides information that helps to overcome this problem
- The number of gaps visible from a vertex is equal to the number of neighbors it has in the map.

## Example 1

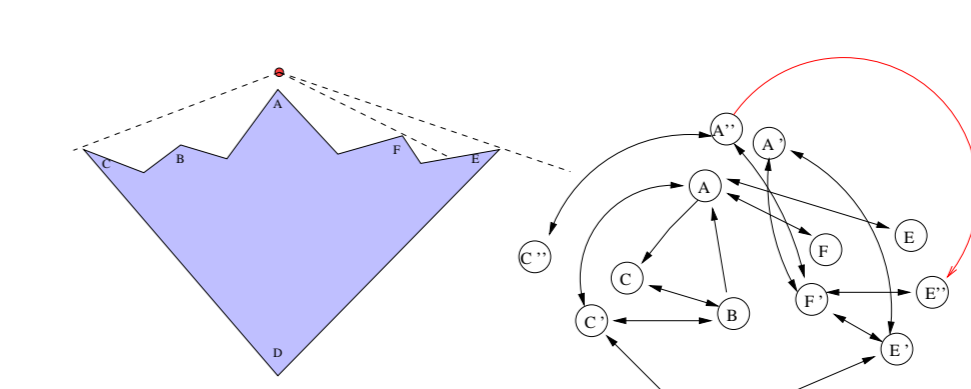


## Example 2

Initially two gaps are visible



Robot Circles the obstacle



Marker is recovered and Graph is reduced

