המסלול האקדמי
המכללה למינהל

Introduction to Robotics
Lesson 4 – Occupancy Grid Maps

LLecturer: Dr. Yehuda Elmaliah, Mr. Roi Yehoshua
Elmaliahy@colman.ac.il
roiyeho@gmail.com

Agenda

המסלול האקדמי
המכללה למינהל

- Occupancy grid maps
- Computing obstacles positions in map
- Using the Laser Sensor

Occupancy Grid Map

המסלול האקדמי
המכללה למינהל

- Now that we have learned how to make the robot move while avoiding obstacles, our next mission is to create a map of the environment
- There are different ways to represent the world space, such as:
  - Voronoi graphs
  - Grid maps
  - Quadtrees
  - and more

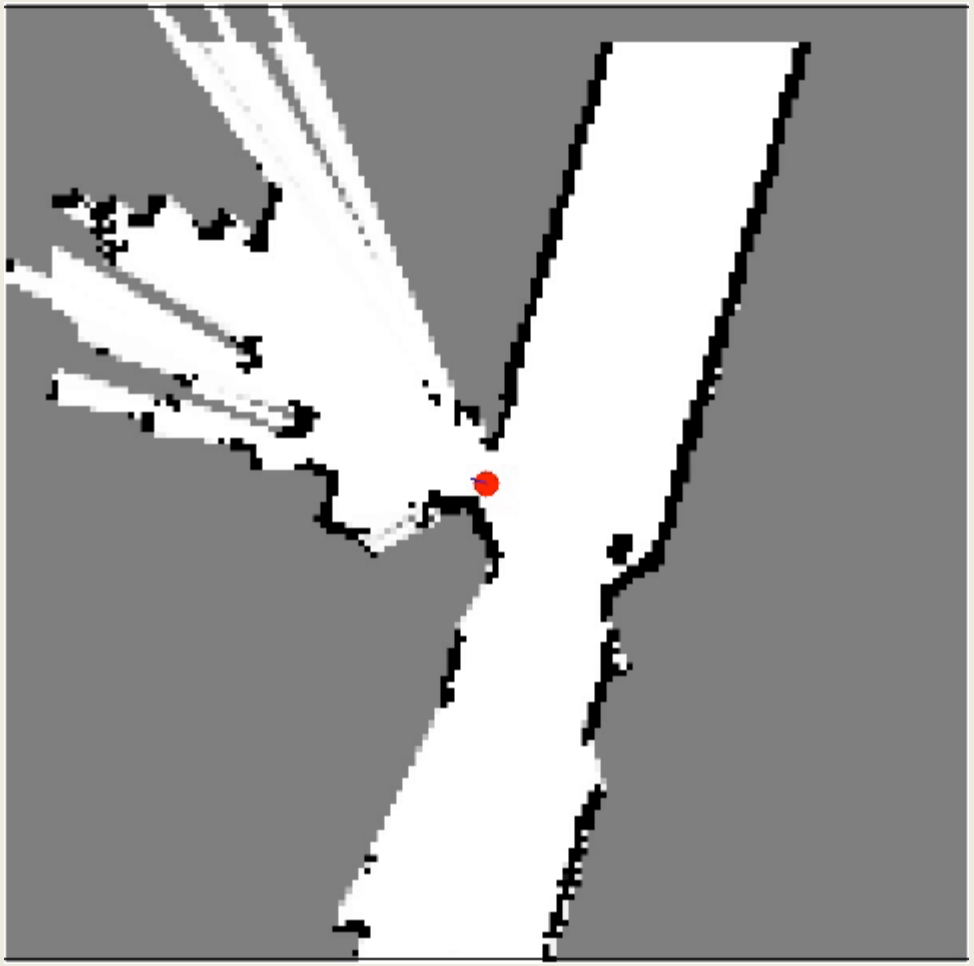
Occupancy Grid Map (OGM)

המסלול האקדמי
המכללה למינהל

- Maps the environment as an array of cells
- Each cell holds a probability value
  - that the cell is occupied
- Useful for combining different sensor scans
  - Sonar, laser, IR, bump, etc.
- Many map building algorithms use this type of map

Occupancy Grid Map

המסלול האקדמי
המכללה למינהל



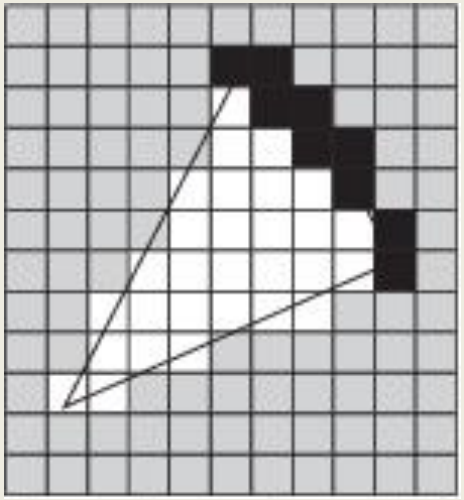
Marking and Clearing

המסלול האקדמי
המכללה למינהל

- Each sensor can be used to either **mark** (insert obstacle information into the map), **clear** (remove obstacle information from the map), or both
- A marking operation is just an index into the array to change the status of a cell
- A clearing operation, however, consists of **raytracing** through a grid from the origin of the sensor outwards for each observation reported

Ray Tracing

המסלול האקדמי
המכללה למינהל



Problems with Grid Maps

המסלול האקדמי
המכללה למינהל

- Grids introduce digitization bias
- If an object falls inside a portion of a grid cell, the whole cell is marked occupied
- This leads to wasted space and very jagged objects
- To reduce wasted space, regular grid for an indoor room are often finely grained, on the order of a few centimeters
- This fine granularity means a high storage cost, and a high number of nodes for a path planning algorithm to consider

Occupancy Grid Map

המסלול האקדמי
המכללה למינהל

- First step – compute the position of obstacles as the robot moves around the environment
- Computation is based on the current position of the robot and the distance of the obstacle from the robot

Localization

המסלול האקדמי
המכללה למינהל

- The attribute **localization** in the Stage world file tells the model how it should record the odometry
- Use SetOdometry() …

Get Robot's Position

המסלול האקדמי
המכללה למינהל

- Use the following methods of Position2dProxy to monitor where the robot thinks it is based on the robot's odometry:
  - **GetXPos()**: gives current x coordinate relative to its x starting position.
  - **GetYPos()**: gives current y coordinate relative to its y starting position.
  - **GetYaw():** gives current yaw relative to its starting yaw.

Get Robot's Position

המסלול האקדמי
המכללה למינהל

```
int main()
{
    PlayerClient pc;
    Position2dProxy pp(&pc);

    pc.Read();

    double x_rob = pp.GetXPos();
    double y_rob = pp.GetYPos();
    double yaw = pp.GetYaw();

    cout << "robot's pose is: (" << x_rob << "," << y_rob << "," <<
yaw << ")" << endl;

    return 0;
}
```

Obstacle's Position

המסלול האקדמי
המכללה למינהל

- Variables:

  x_{rob} – x position of the robot

  y_{rob} – y position of the robot

  α – robot's orientation

  β – angle of the sensor relative to the robot

  d – distance between the robot and the obstacle

  x_{obs} – x position of the obstacle

  y_{obs} – y position of the obstacle

Obstacle's Position

המסלול האקדמי
המכללה למינהל

$$x_{obs} = x_{rob} + d \cdot \cos(\alpha + \beta)$$

$$y_{obs} = y_{rob} + d \cdot \sin(\alpha + \beta)$$

- Make sure α and β are measured in radians (the cos and sin functions work with radians)

Exercise

המסלול האקדמי
המכללה למינהל

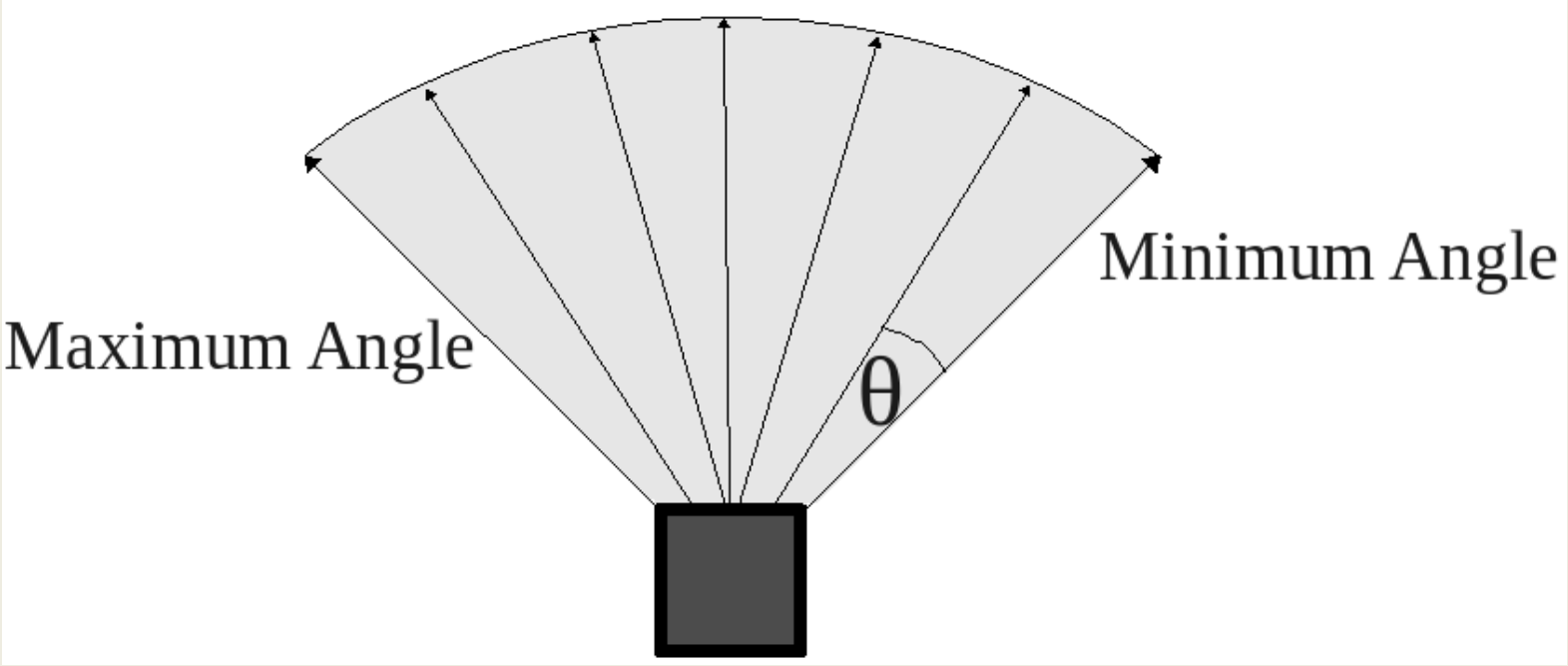
- Use the sonar sensors to compute the position of the obstacles that the robot senses from its current location

Laser Scanner

המסלול האקדמי
המכללה למינהל

- A laser is a special case of ranger device
- It makes regularly spaced range measurements turning from a minimum angle to a maximum angle.
- Each measurement, or scan point, is treated as being done with a separate ranger.
- The angles are given with reference to the laser's centre front

Laser Scanner



Hokuyo Laser

המסלול האקדמי
המכללה למינהל

- PcBot uses the Hokuyo URG laser

  http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html

Hokuyo Laser

המסלול האקדמי
המכללה למינהל

Model No.	URG-04LX
Power source	5VDC ± 5%*1
Current consumption	500mA or less(800mA when start-up)
Measuring area	60 to 4095mm(white paper with 70mm□)
	240°
Accuracy	60 to 1,000mm : ± 10mm, 1,000 to 4,095mm : 1% of measurement
Repeatability	60 to 1,000mm : ± 10mm
Angular resolution	Step angle : approx. 0.36° (360° /1,024 steps)
Light source	Semiconductor laser diode(λ =785nm), Laser safety class 1(IEC60825-1, 21 CFR 1040.10 & 1040.11)
Scanning time	100ms/scan
Noise	25dB or less
Interface	USB, RS-232C(19.2k, 57.6k, 115.2k, 250k, 500k, 750kbps), NPN open-collector(synchronous output of optical scanner : 1 pce)
Communication specifications	Exclusive command(SCIP Ver.1.1 or Ver.2.0)*2
Ambient temperature/humidity	-10 to +50 degrees C, 85% or less(Not condensing, not icing)
Vibration resistance	10 to 55Hz, double amplitude 1.5mm Each 2 hour in X, Y and Z directions
Impact resistance	196m/s^2, Each 10 time in X, Y and Z directions
Weight	Approx. 160g
Accessory	Cable for power・communication/input・output(1.5m) 1 pce, D-sub connector with 9 pins 1 pce*3

Hokuyo Laser

המסלול האקדמי
המכללה למינהל

- Adjust the Hokuyo laser definition in the robot's definition file wbr914.inc to the laser's spec

```
define hokuyo_URG_laser laser
(
    range_min 0.0
    range_max 4.095
    fov 240.0
    samples 666

    color "red"
    size [ 0.05 0.05]
)
```

Laser Readings

המסלול האקדמי
המכללה למינהל

- The number of samples you will receive in the scan array is:

range (240°) / resolution (0.36°)

Laser Readings

המסלול האקדמי
המכללה למינהל

- The relation between the scan's index in the array and its angle is:

$$\vartheta = i \cdot \text{angular\_resolution} - \text{min\_angle}$$

$$\vartheta = i \cdot 0.36^{\circ} - 120^{\circ}$$

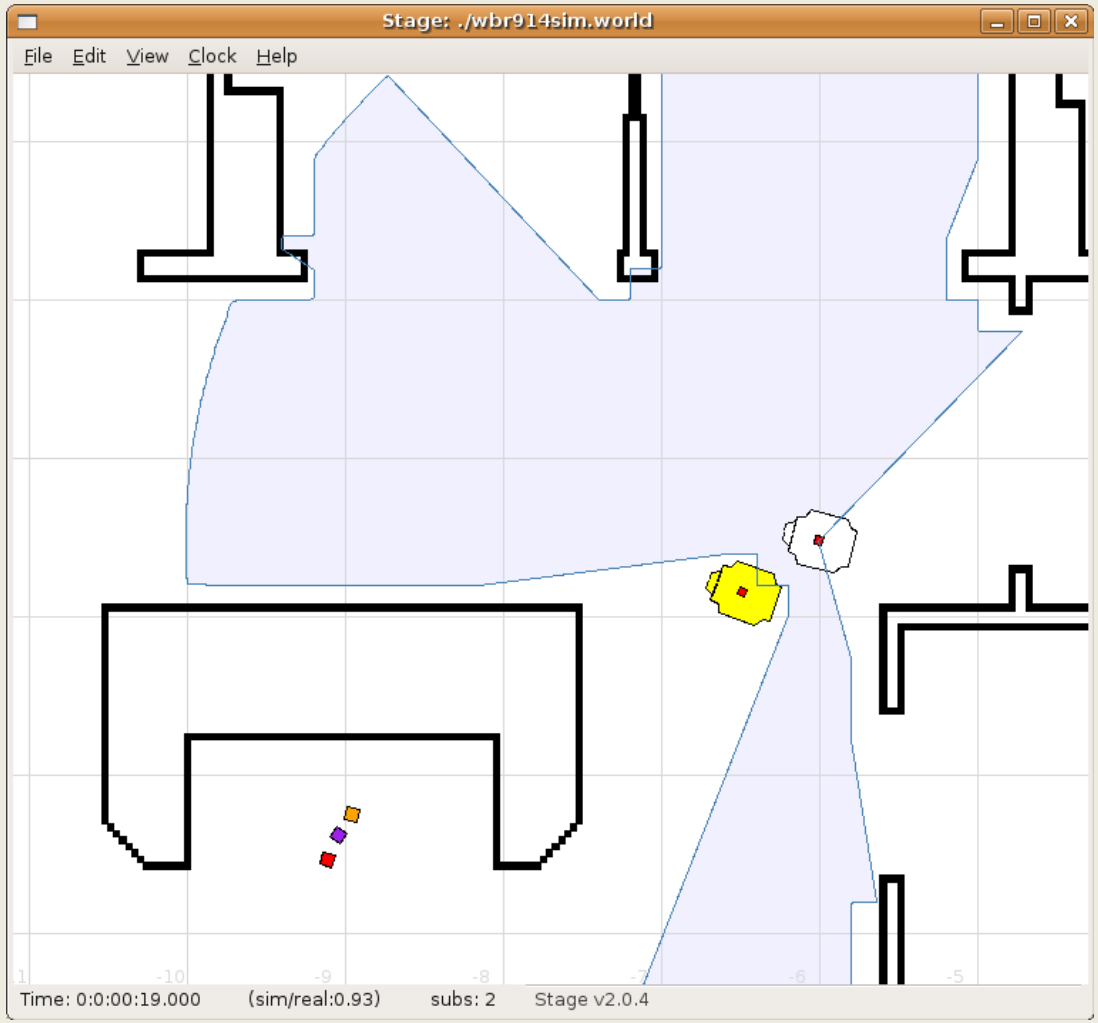
LaserProxy Example

המסלול האקדמי
המכללה למינהל

```
#define SAMPLES_NUM 666

int main()
{
    PlayerClient pc;
    LaserProxy lp(&pc);
    Position2dProxy pp(&pc);

    while (true) {
        pc.Read();

        for (int i = 0; i < SAMPLES_NUM; i++)
            cout << lp[i] << endl;

        sleep(1.0);
    }
    return 0;
}
```

LaserProxy Example

המסלול האקדמי
המכללה למינהל



Exercise

המסלול האקדמי
המכללה למינהל

- Use the laser sensor to compute the position of the obstacles in the map

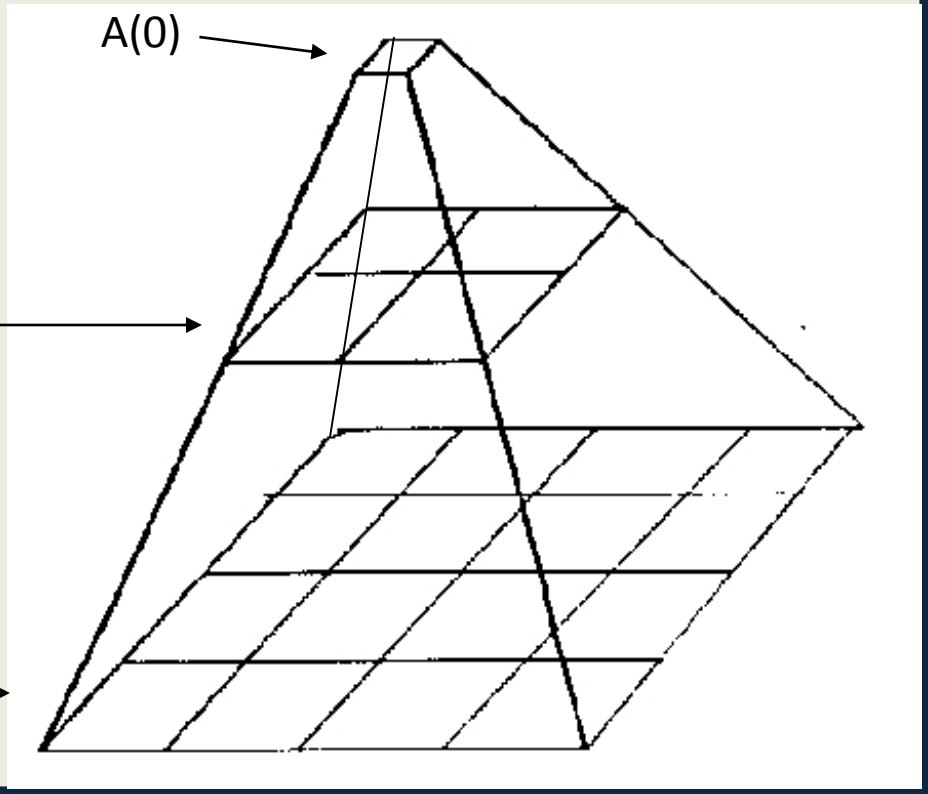
Resolution in Maps

המסלול האקדמי
המכללה למינהל

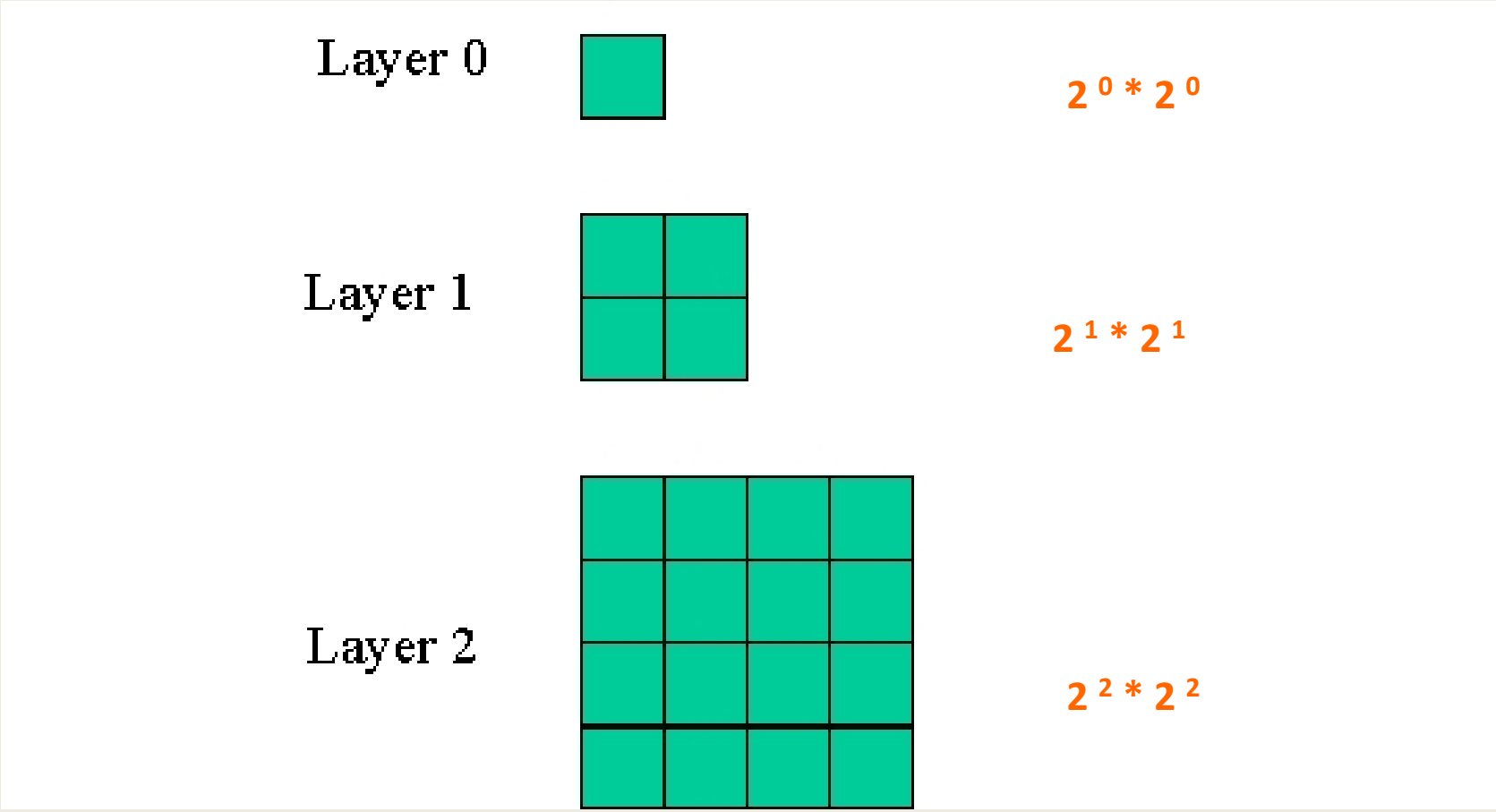
- Given a 2^n by 2^n map array, say A(n), pyramid is a sequence of arrays {A(i)} such that A(i-1) is a version of A(i) at half of the resolution of A(i) and so on.

$2^1 * 2^1$

$2^2 * 2^2$

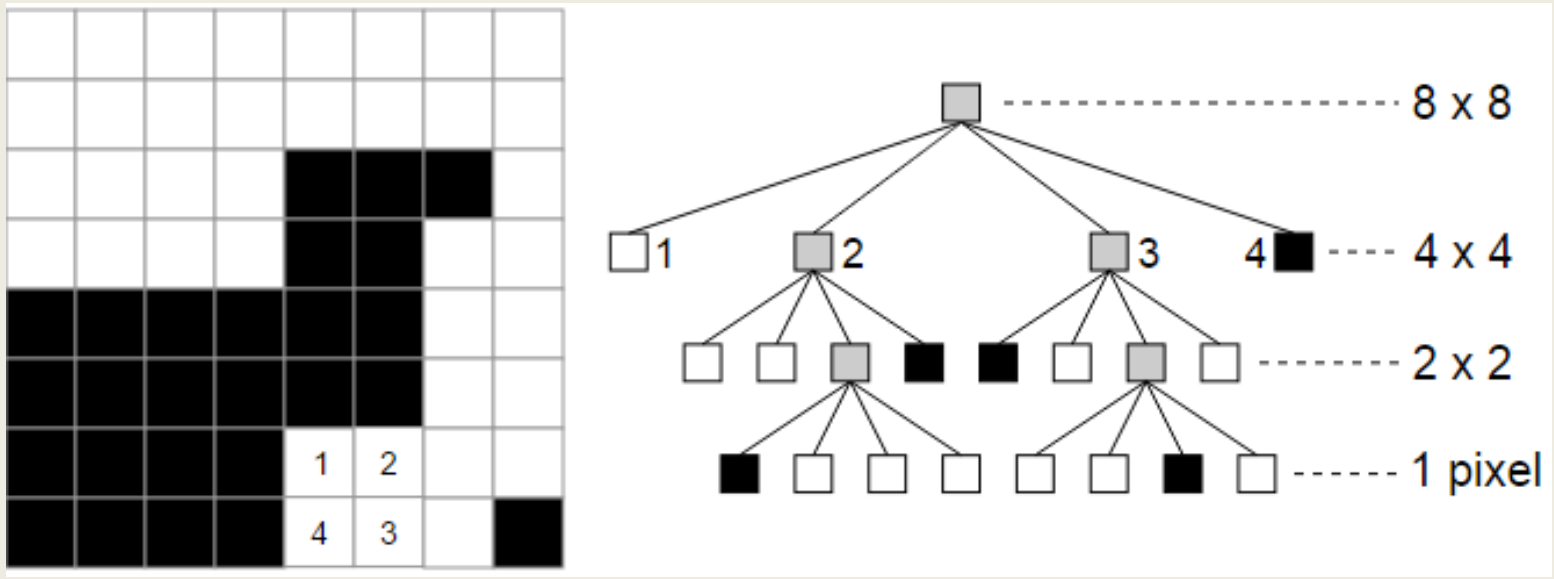


Example of a Four Layer Pyramid



Quadtree

- A quadtree is a tree data structure in which each internal node has exactly four children.
- Map representation with Quadtree:



Blow Maps

המסלול האקדמי
המכללה למינהל

- White spaces on map defines empty spaces
- Robot can’t move at any empty spaces because of it size
- Solution blow obstacles by the half size of the robot

Working with PNG

המסלול האקדמי
המכללה למינהל

- lodepng.cpp and lodepng.h will help you to work with .png files
- http://lodev.org/lodepng/
- Exercise: write a program that load a .png file and blow it.