Exercise 1: Pursuit Evasion Problem

Suppose a certain number of robots are chasing a moving intruder through a known, bounded environment. The robots have omni-vision and can detect the intruder at any distance if the intruder is in the line-of-sight. Can you draw an environment where \( k \) robots can succeed in finding the intruder in finite time, but \( k - 1 \) robots cannot? Draw such an environment for \( k = 2, k = 3, \) and \( k = 4 \) robots. Describe the successful search strategy for \( k \) robots and explain why \( k - 1 \) robots could not accomplish the task.

Exercise 2: Entropy

1. Compute the entropy \( H(p) \) in bits (therefore use \( \log_2 \)) of the following discrete distribution \( p \):

\[
\begin{array}{ccccc}
  & p(x_1) & p(x_2) & p(x_3) & p(x_4) \\
 0.04 & 0.06 & 0.2 & 0.7
\end{array}
\]

2. Prove that the entropy of a grid map cell \( m_{x,y} \) is maximal for \( p(m_{x,y}) = 0.5 \).

3. Consider a discrete uniform distribution of a random variable with \( n \) possible outcomes. Prove that the entropy of the distribution decreases if you change the distribution by increasing the probability of a single event and accordingly reducing the probability of another event.

Exercise 3: Line Extraction

Assume you have an array of 2d-points (ordered by the \( x \) coordinate). These points should be approximated by a set of lines. Implement the Douglas-Peucker-Line-Approximation-Algorithm (“Split and Merge”).

A sample program you should complete (written in C/C++) is available on our web page.