Hidden Markov Models

Jesus Savage

School of Engineering, Bio-Robotics Laboratory

National Autonomous University of Mexico, UNAM

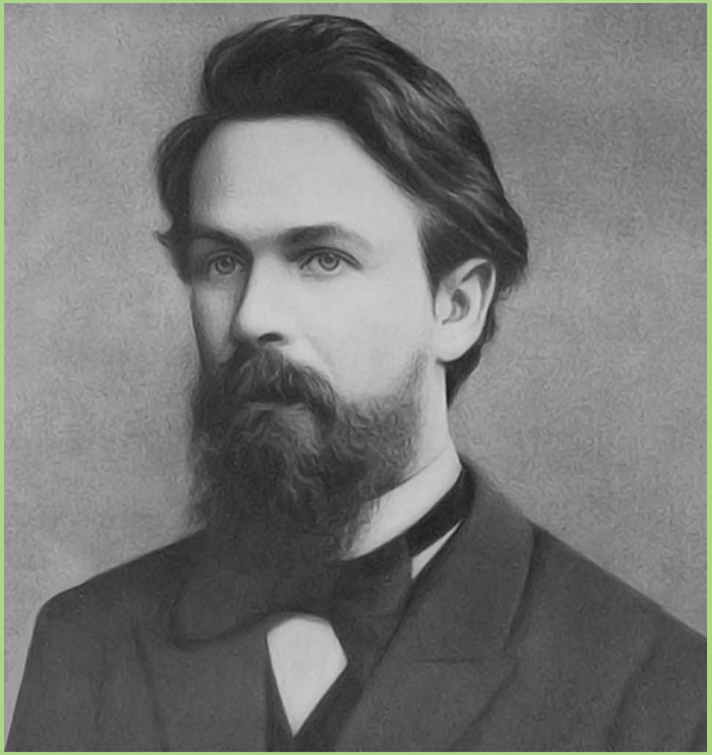
Workshop ICR 2024
October 13th 2024

OUTLINE

- Introduction
- Markov Chain Models
- Hidden Markov Models
- Applications of Hidden Markov Models in Robotics

Andrei Andreyevich Markov
1856 Ryazan, Russia, 1922 St. Petersburg, Russia

In 1874, he entered the Faculty of Physics and Mathematics in St. Petersburg. He attended classes under Korkin, Zolotarev and especially Chebyshev, who was the head of the mathematics department. Markov was the most elegant spokesman and follower of Chebyshev's ideas. His contributions to Jacob Bernoulli's theorem known as the Law of Large Numbers, to two fundamental probability theorems due to Chebyshev, and to the method of least squares are notable.



Andrei Andreyevich Markov

- Markov is particularly remembered for his study of the so-called Markov chains, sequences of random variables in which the next variable is determined by the current variable but is independent of the previous ones. With this a new branch of probability theory arose and the theory of stochastic processes began. In 1923, Norbert Wiener rigorously introduced continuous Markov processes. However, the general theory was studied, in the 1930s, by Andrei Kolmogorov.

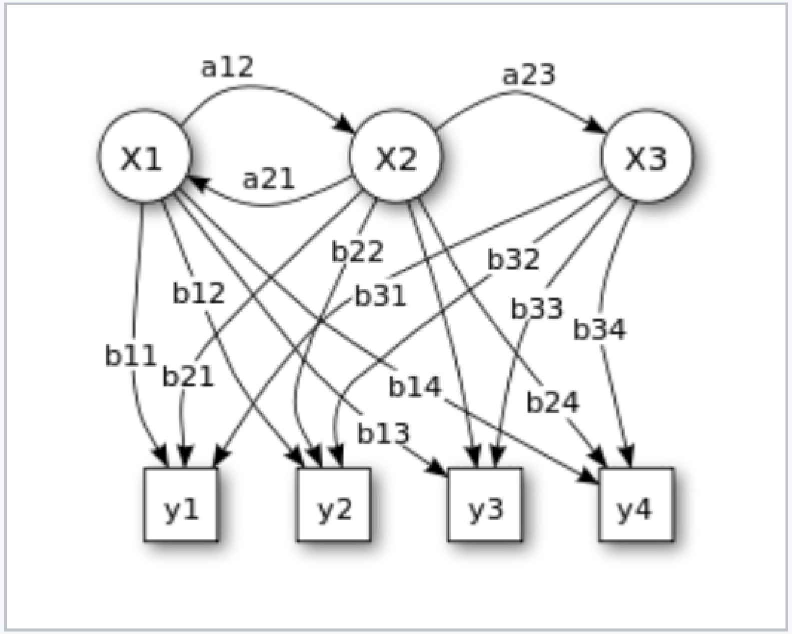


Figure 1. Probabilistic parameters of a hidden Markov model (example)

X — states

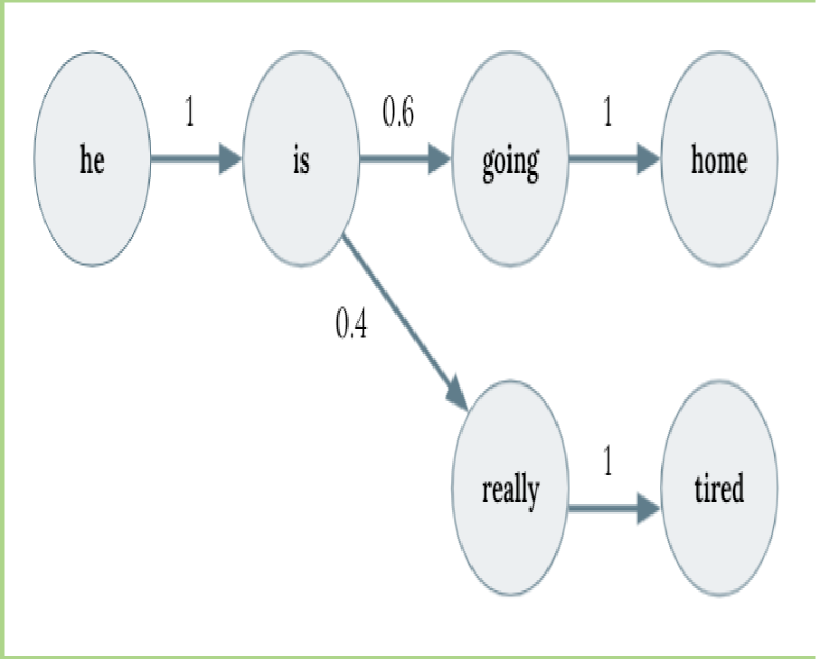
y — possible observations

a — state transition probabilities

b — output probabilities

Andrei Andreyevich Markov

- Markov developed his theory of chains, from a completely theoretical point of view, he also applied these ideas to chains of two states, vowels and consonants, in literary texts.
- Beginning in 1889, Georgy Voronyi studied at Saint Petersburg University, where he was a student of Andrey Markov.

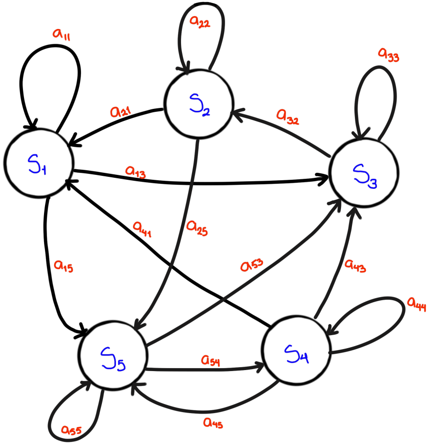


Hidden Markov Models

Markov chain theory has been extensively used to solve pattern recognition problems. The main assumption is that one-, two-, and three-dimensional signals can be characterized as a parametric random process, and that the parameters of the stochastic process can be determined (estimated) in a precise and well-defined manner. They are mainly used for time-varying type of processes.

Discrete-time Markov processes

A Markov chain consists of a set of states linked by links. The transition from one state to another is determined by probabilities.



Discrete-time Markov processes

The changes of states are called q_t at time t. Probability that the Markov chain is in a given state j

$$P[q_t = j \mid q_{t-1} = i, q_{t-2} = k, \ldots] = P[q_t = j \mid q_{t-1} = i]$$

$$a_{ij} = P[q_t = j \mid q_{t-1} = i], 1 \leq i\, j \leq N$$

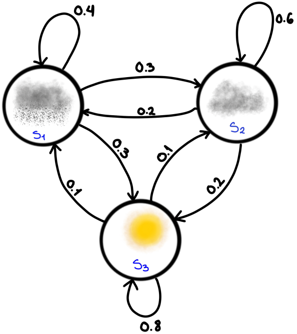
$$0 \leq a_{ij} \leq 1 \ \forall \ i, j$$

$$\sum_{j=1}^{N} a_{ij} = 1 \ \forall i$$

This model is observable since the output is a set of states at each instant where each state represents an observable event.

Example

Suppose we have a weather model as described below:



Where: $S_1 = Rain$; $S_2 = Cloudy$; $S_3 = Sunny$
Time t is taken at 12:00 on the day.

Example

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

Problem: What is the probability that the weather on 8 consecutive days will be:
sunny, sunny, sunny, rainy, rainy, sunny, cloudy, sunny?
Defining the observation sequence $\underline{O}$, as

$$\underline{O} = (Sunny, Sunny, Sunny, Rainy,$$

$$Rainy, Sunny, Cloudy, Sunny)$$

$$\underline{O} = (S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3)$$

$$P(\underline{O}|model) = Probability\ of\ observing\ the\ sequence\ \underline{O},$$

$$given\ the\ climate\ model$$

Example

$$= P(S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3|model)$$

$$= P(S_3)P\left[S_3|S_3\right]^2 P\left[S_1|S_3\right] P\left[S_1|S_1\right] P\left[S_3|S_1\right] P\left[S_2|S_3\right] P\left[S_3|S_2\right]$$

$$= \pi_3(a_{33})^2\; a_{31}\; a_{11}\; a_{13}\; a_{32}\; a_{23}$$

$$= (1.0)(0.8)^2(0.1)(0.4)(0.3)(0.1)(0.2) = 1.536x10^{-4}$$

Where the following notation is used:

$$\pi_3 = P\left[q_1 = 3\right] = 1$$

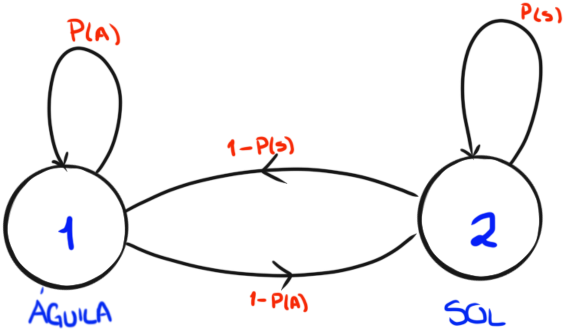
π_i are the initial probabilities, in this example when the experiment started it was already a sunny day, therefore $\pi_3 = 1$.

Example

Suppose a person is flipping coins behind a curtain, with different coins loaded.
The person only tells the result obtained without saying which coin he used. How would a Hidden Markov Chain model be constructed that would explain the recurrence of heads and tails obtained?

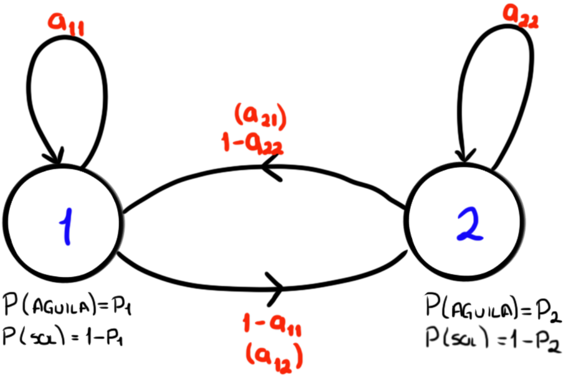
$$\underline{O} = (O_1, O_2, O_3, ..., O_T)$$
$$= (TTHHT...H)$$

Model only one coin



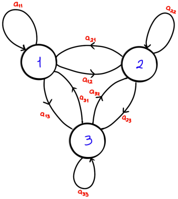
Observations = TTHHTHTHH
States = 112212122
The states correspond to the observations.

Two-coin model



Observations = TTHHTHTHH
States = 211122211

Two-coin model



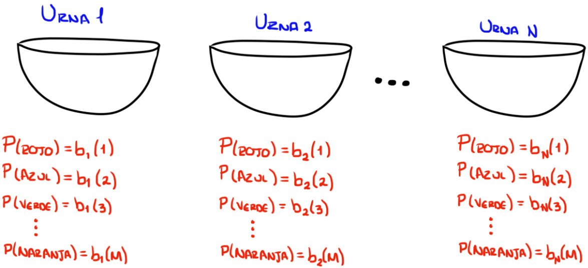
$$Observations = \underline{O} = SSAASASAASSA$$
$$States = \underline{q} = 312331111313$$

$$P_1(T) = P_1; \quad P_2(T) = P_2; \quad P_3(T) = P_3$$
$$P_1(H) = 1 - P_1; \quad P_2(H) = 1 - P_2; \quad P_3(H) = 1 - P_3$$

The $a_i s$ represent the probabilities of changing the coins when the toss is made.

Model of the ballot boxes with balls

There are N urns with M balls of different colors. A person randomly chooses a ball from one of the urns and shows it, repeating this operation several times. The entire process corresponds to an observable output of an HMM.



$$\underline{O} = \{Green, Green, Blue, Red, Yellow, ..., Blue\}$$

Elements of Hidden Markov Models (HMM)

Elements of HMM for discrete symbol observations.

1 N number of states in the model.
  Although the states are hidden, for many practical applications there is physical evidence for each of the states. Each state is denoted as $\{1, 2, ..., N\}$, and the time in the state as q_t

2 M, the number of distinct observations in each state, i.e., the discrete-sized alphabet.
  Observations correspond to physical outputs of the system being modeled.
  Individual symbols are represented as $V = \{V1, V_2, ..., V_M\}$

Model of the ballot boxes with balls

3 The probability distribution is of state change $A = \{a_{ij}\}$ where

$$a_{ij} = P\{q_{t+1} = j | q_t = i\}, 1 \leq i, j \leq N;$$

$$0 \leq a_{ij} \leq 1; \quad \forall i, j$$

4 The probability distribution of the observed symbol, $B = \{b_j(k)\}$

$$b_j(k) = P[O_t = V_k | q_t = j],\ 1 \leq k \leq M$$

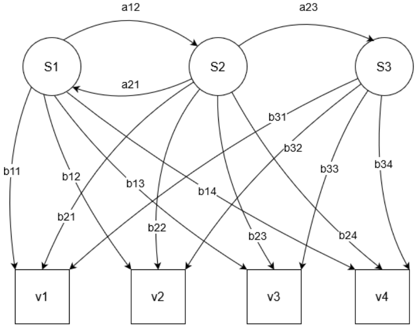
5 The distribution of the initial state $\pi = \pi_j$

$$\pi_i = P[q_1 = i],\ 1 \leq i \leq N$$

Hidden Markov Models

The complete Hidden Markov Model (HMM) then requires the values of N, M, the output symbols V and a specification of the probabilities A, B and π. For convenience the following notation is used:

$$\lambda = (A, B, \pi)$$



Basic problems of HMM's

Problem 1:
Given the sequence of observations $\underline{O} = (O_1, O_2, ..., O_T)$ and the model $\lambda = (A, B, \pi)$, how do efficiently calculate the probability of the observed sequence, given the model λ?
This problem is called the evaluation problem.

Problem 2:
Given the sequence of observations $\underline{O} = (O_1, O_2, ..., O_T)$ and the model λ, how do you choose the optimal sequence $\underline{q} = (q_1, q_2, ..., q_T)$ that best explains the given observations?
It is about finding the hidden variables in the model.

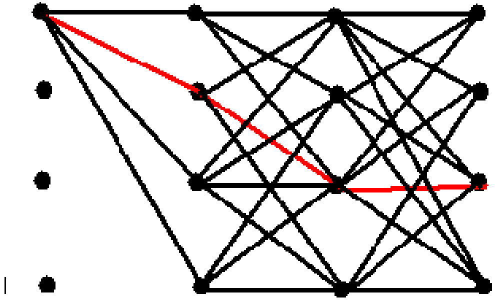
Problem 3:
How are the parameters of the model $\lambda = (A, B, \pi)$ calculated that maximize the $p(\underline{O}|\lambda)$?
The procedure that maximizes the $p(\underline{O}|\lambda)$ is used to train the HMMs.

Optimal sequence of states of an HMM

Given the observation sequence $\underline{O} = (O_1, O_2, ..., O_T)$ and the model λ of an HMM, how do you choose the optimal sequence $\underline{q} = (q_1, q_2, ..., q_T)$ that best explains the given observations?

It is about finding the hidden variables in the model. One of the most used algorithms to solve this problem is the Viterbi algorithm.

Trellis diagrams are used to visualize this algorithm.

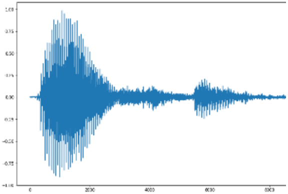


Applications of Hidden Markov Models in Robotics

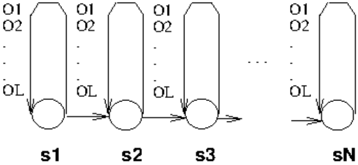
1. Speech Recognition
2. Objects and Places Recognition
3. Robots Behaviors

Single word recognition using HMMs

Each word is represented by a hidden Markov chain model, λ_i. For speech signals, such as the one shown below:

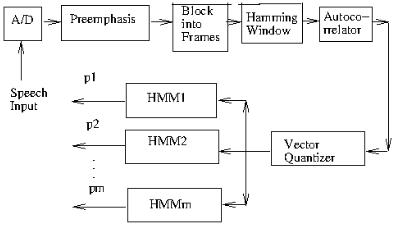


The type of HMM to represent this type of speech signals is a left-to-right one, as shown in the following figure.



Single word recognition using HMMs

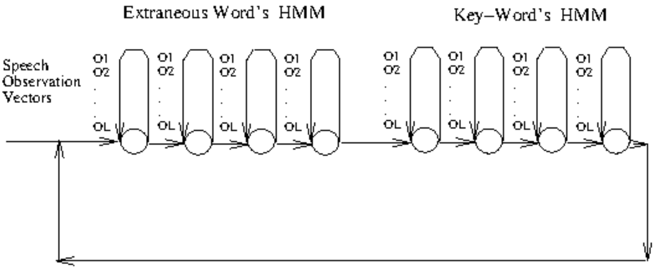
The block diagram for a single word recognizer using HMMs is shown below.

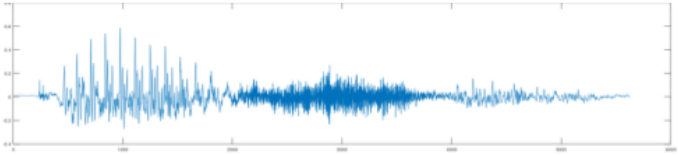


Given the observation vector $\underline{O} = (O_1, O_2, ..., O_t, ..., O_T)$, obtained after quantizing the speech signal, where each observation O_t corresponds to the centroid index that most closely resembles the LPC vector of one of the blocks of the speech signal, the probability $p_j = P(\underline{O}|\lambda_j)$ is obtained. The word i is recognized if:

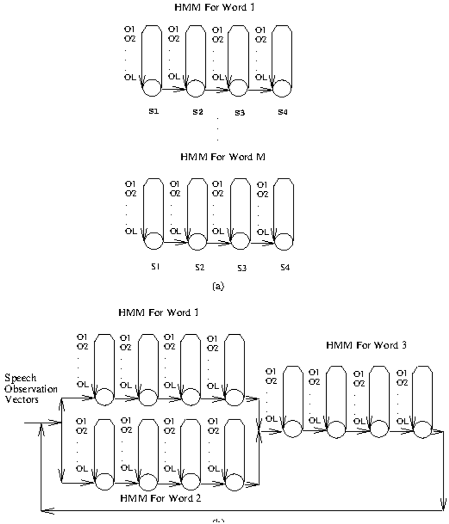
$$i = argmax[P(\underline{O}|\lambda_j)], 1 \leq j \leq M$$

Keyword recognition spotting using HMMs





Creating grammars with keywords using HMMs



Introduction

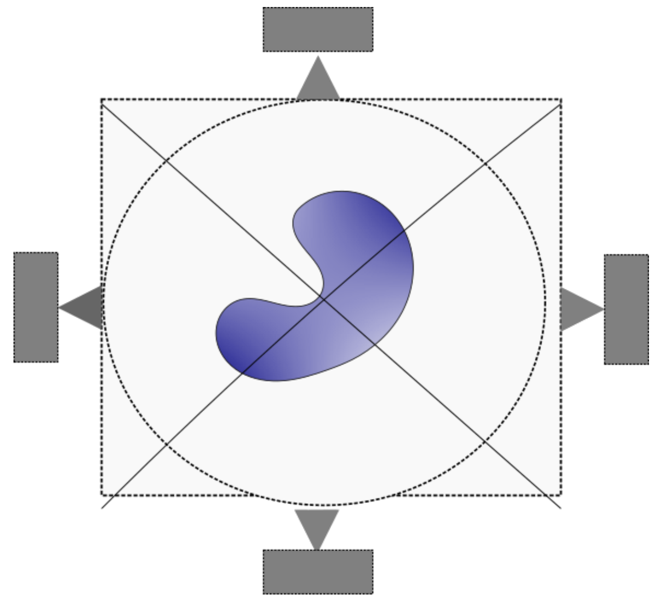
The recognition and manipulation of every-day objects represent a key challenge for achieving proper operation of robotics.

It is desired that the training consume little time, few computational resources and the minimum of training images.

Robot Takeshi (Toyota Human Support Robot [Yamamoto et al., 2018]) viewing an object

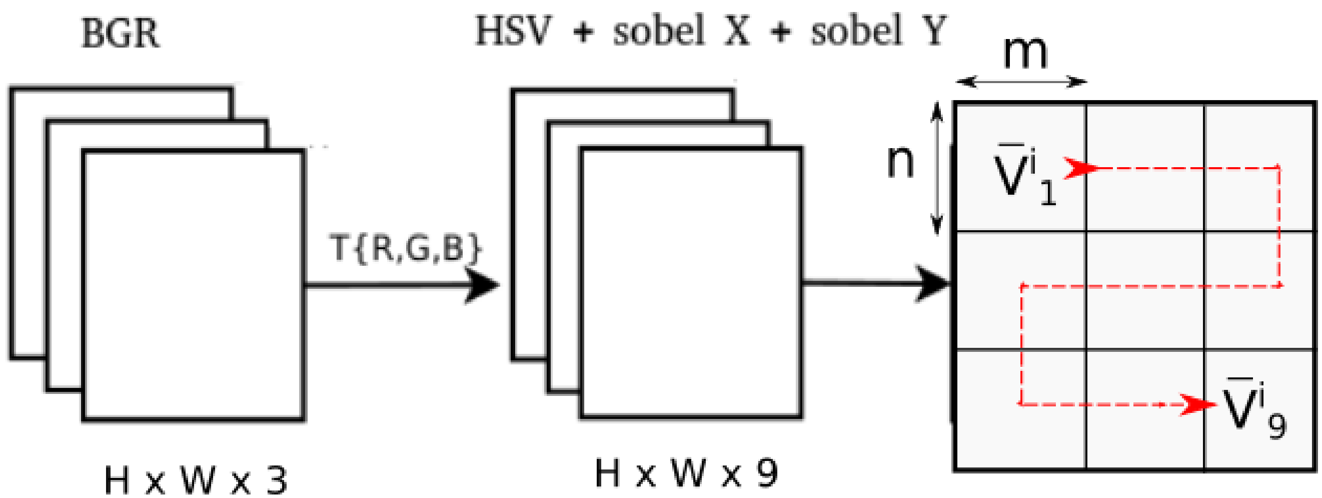
Proposal

In this work, we tackle this problem by estimating the object view dividing the space around it in 4 possible areas Then, by taking any image of the object it should be possible to classify it as an image that belongs to one of the feasible perspectives.



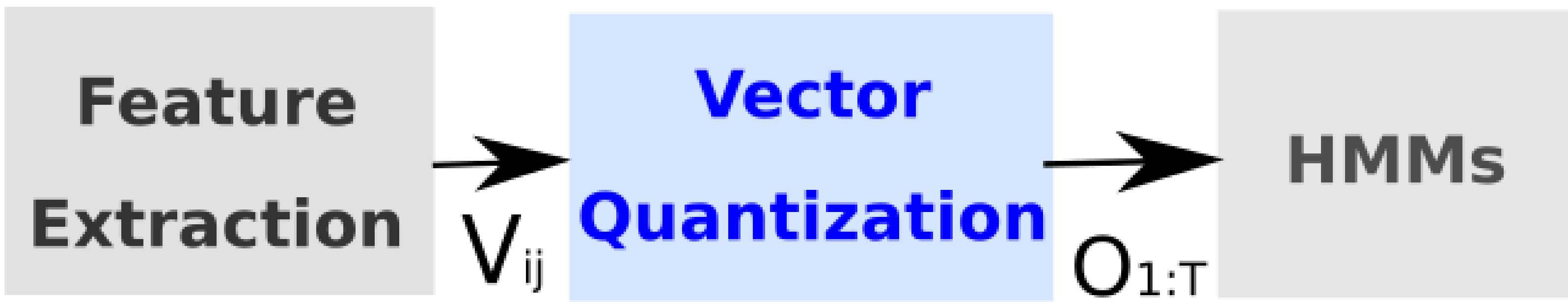
Top view of an object with its candidates pose

A sample of possible sequence

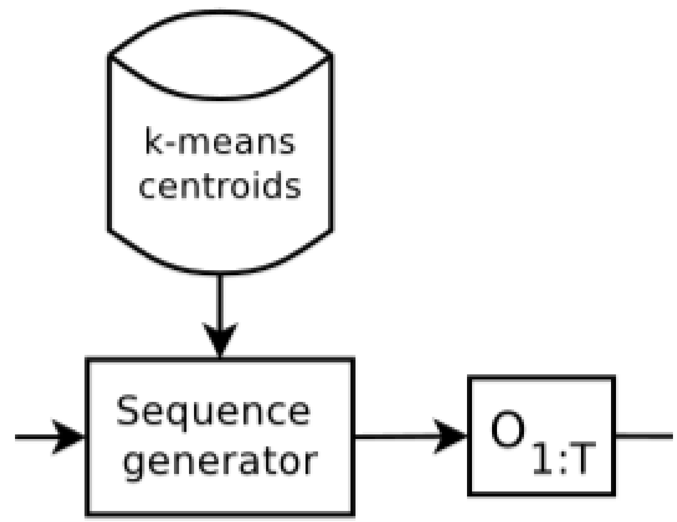


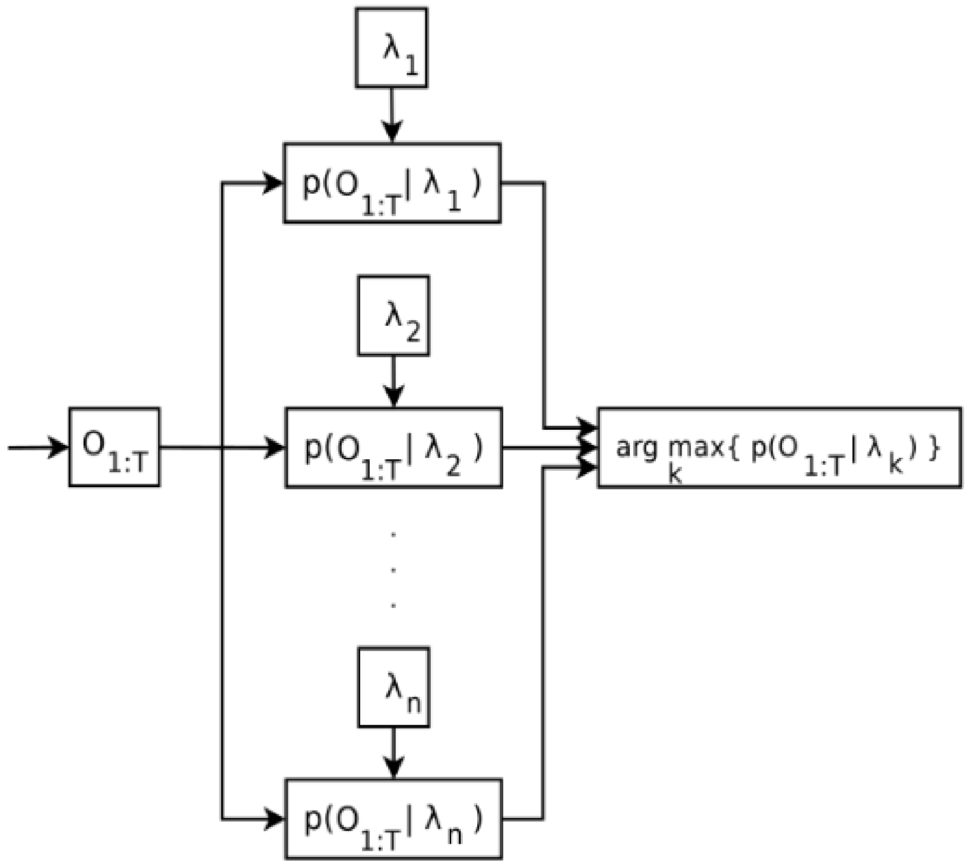
$$
\begin{aligned}
\vec{V}_1^{\,i} &= p_{1\,1}, p_{1\,2} \cdots p_{m\,n} \\
\vec{V}_2^{\,i} &= p_{1\,n+1}, p_{1\,n+2} \cdots p_{m\,2n} \\
\vec{V}_3^{\,i} &= p_{1\,2n+1}, p_{1\,2n+2} \cdots p_{m\,3n} \\
&\vdots \\
\vec{V}_9^{\,i} &= p_{2m+1\,2n+1}, p_{2m+1\,n+2} \cdots p_{3m,3n}.
\end{aligned}
$$

H = 3n, W = 3m



- Vectors V_{ii} feed the VQ for latter generate the index centoids.
- $O_i = \{X^*_0, X^*_1, \ldots, X^*_9\}$, where X^* is the closest codeword index to the associated input roi.
- Kmeans ++ algorithm was used.

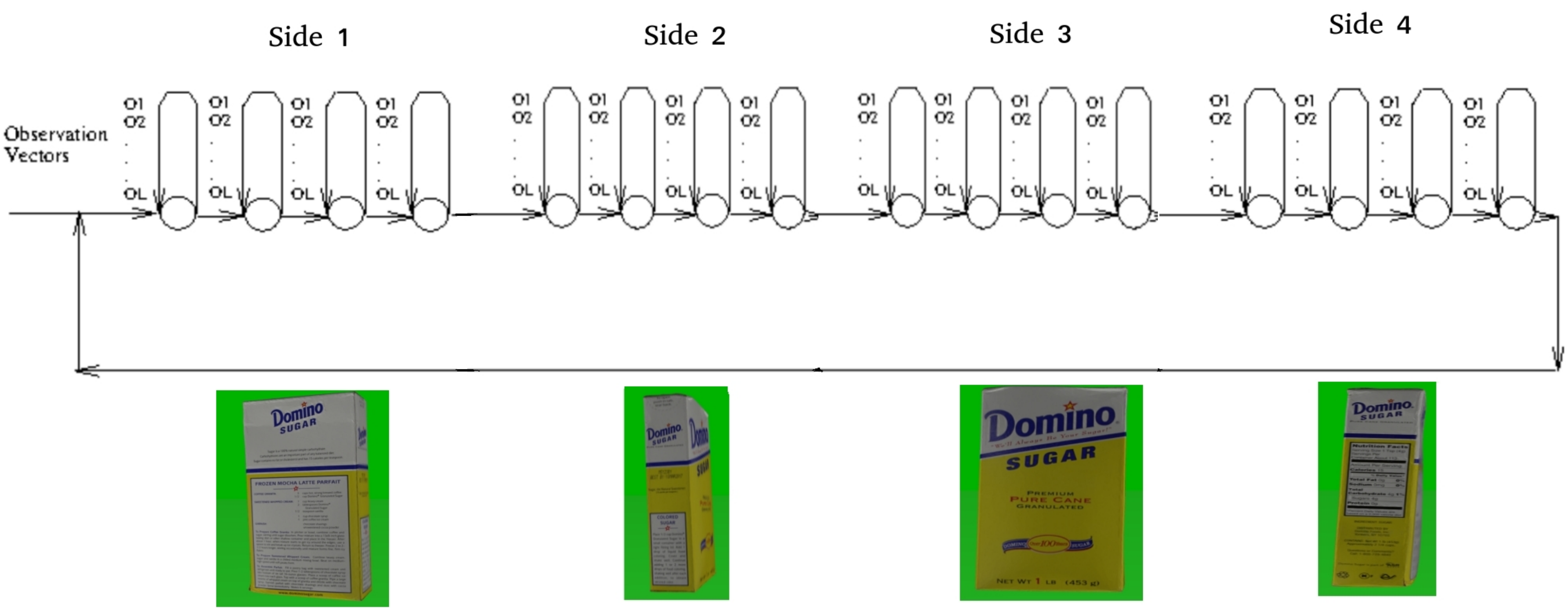




Each model is defined by

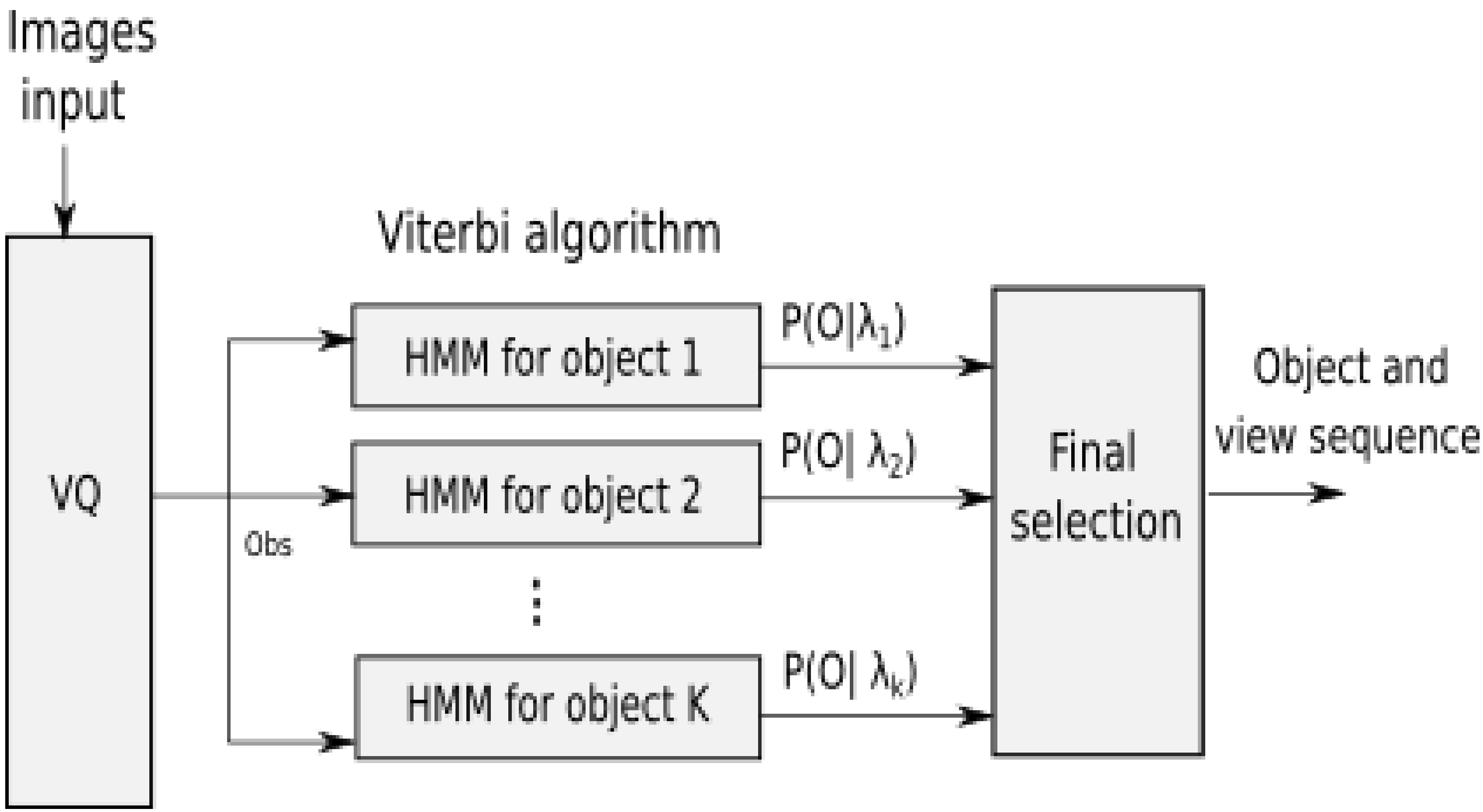
$$\lambda = (A, B, \pi). \tag{1}$$

Development of the proposed architecture: Concatenation



The previous models are concatenated in order to form a new HMM for each object. The news chains are re- trained.

Development of the proposed architecture: Inference



Block diagram of the complete solution

Forward algorithm is used to predict the Object and the Viterbi algorithm for the view sequence.

Introduction

- Localization for a service robot is of the utmost importance when it navigates in structured environments.
- At any given time, the mobile robot takes readings from its sensors, each of them subject to noise.
- The robots also make movements, also subject to errors.

Introduction

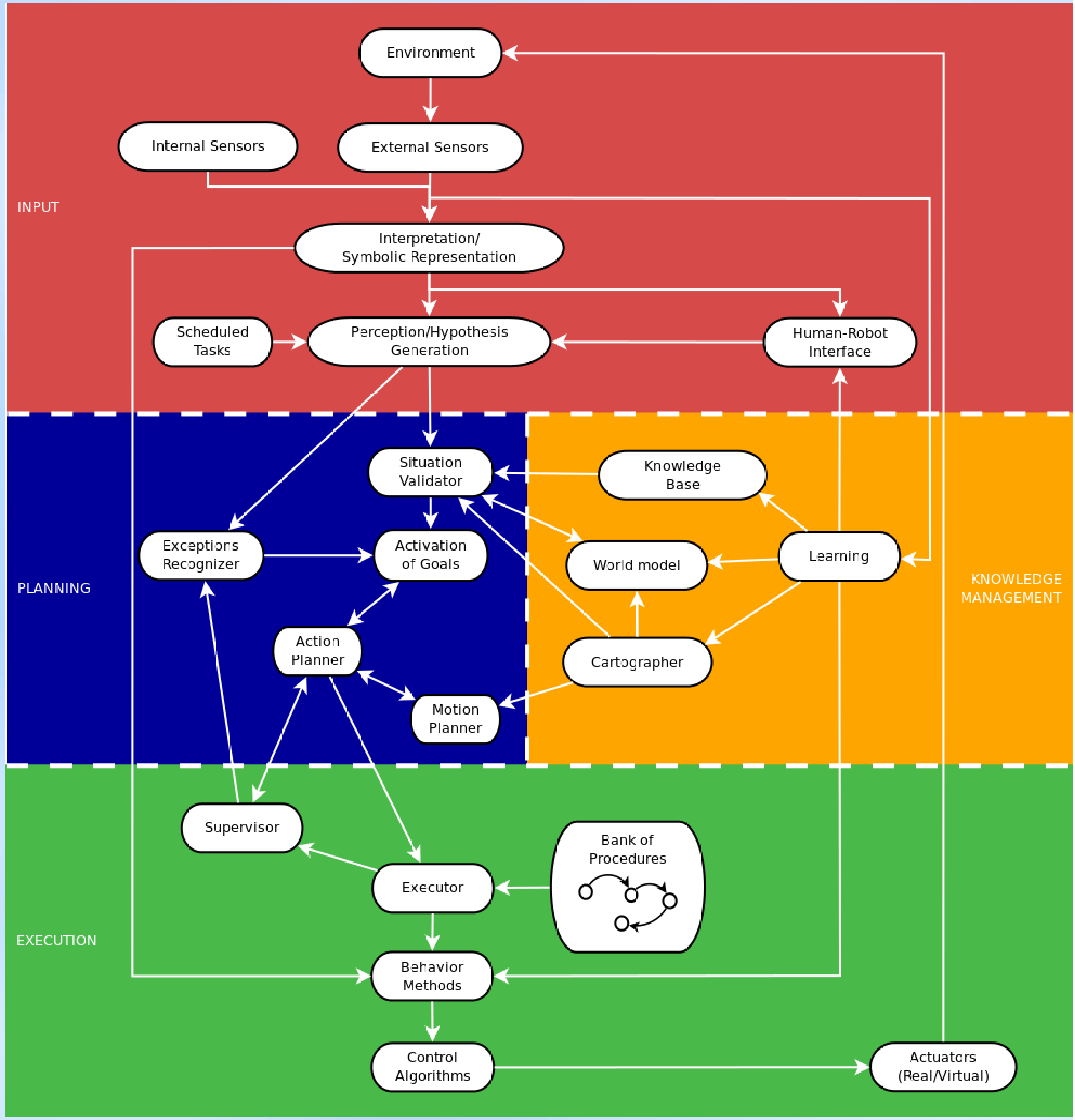
- The sum of this errors over time makes their position unreliable . Hence the need of estimating positions and the need of creating convenient environment representations.
- Localization consists of estimating the location and orientation of a mobile robot given a map (or model) and sensor data.
- In this work a stochastic model, Hidden Markov Models, is proposed for map representation of the environment where the robot navigates.

ViRBoT

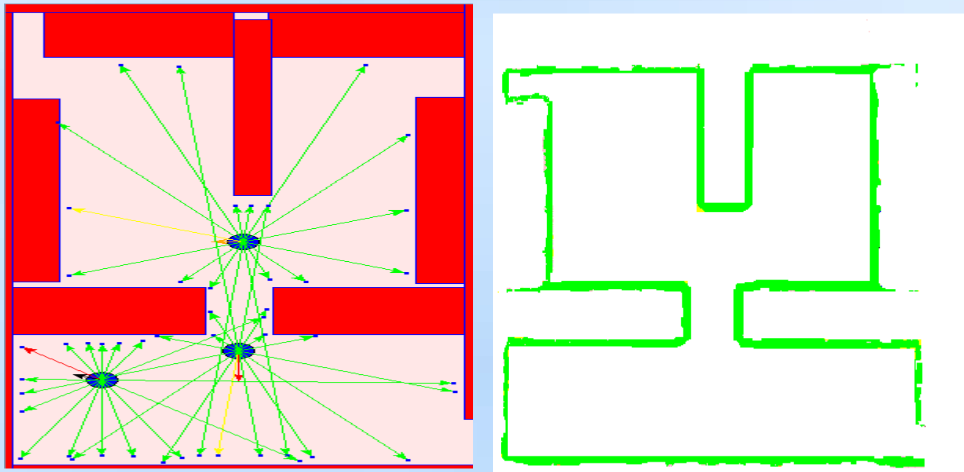
- We have developed a system, named the VIRBOT, where operational algorithms for mobile robots can be tested.

  The system consists of several layers that control the operation of robots.

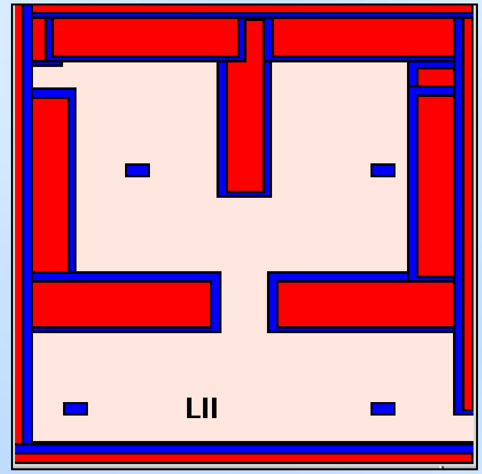
- The Knowledge Management layer contains a Cartographer Module, in which several map representations of the environment are available.



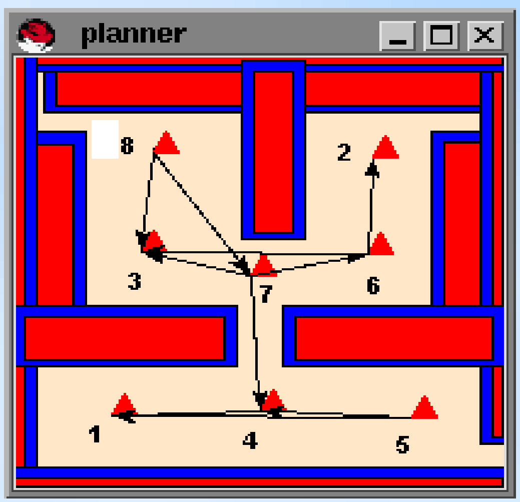
Cartographer



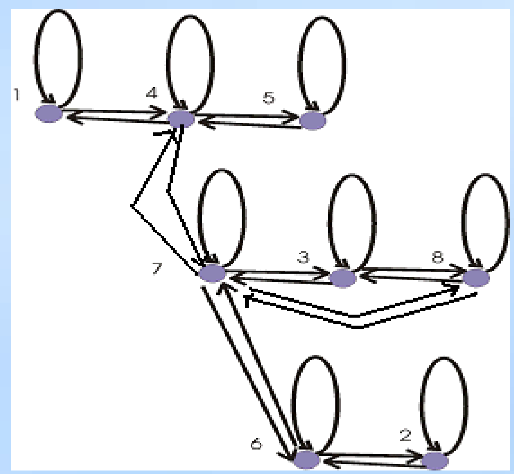
- RAW MAPS



- SYMBOLIC MAPS

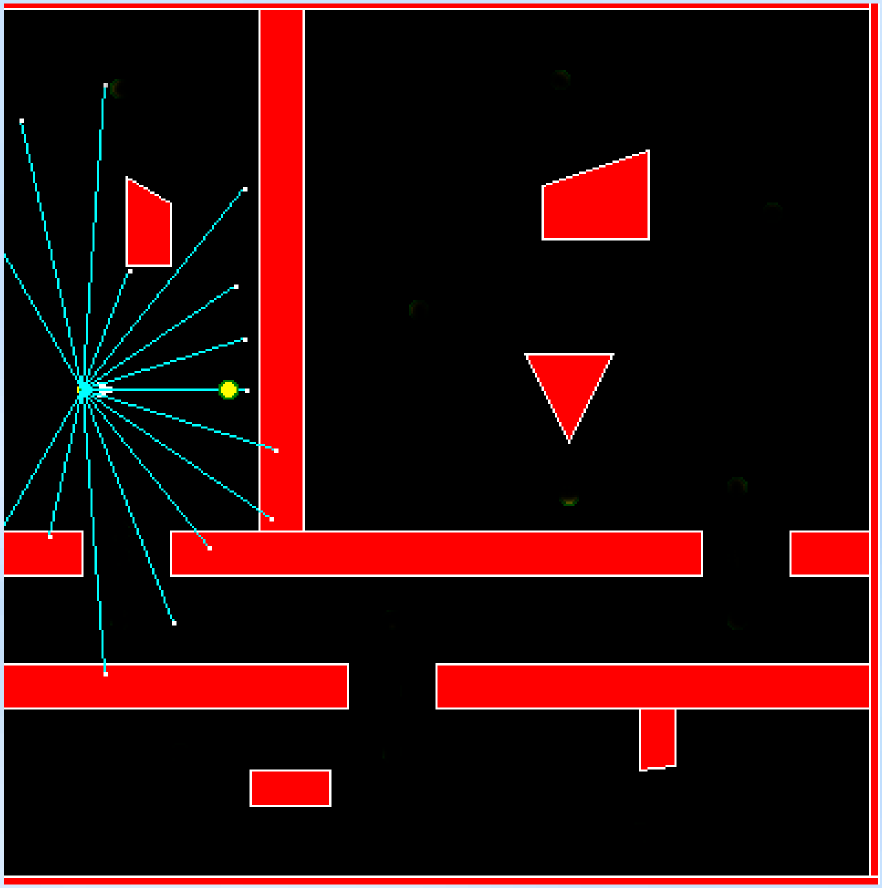


- TOPOLOGIC MAPS



- PROBABILISTIC MAPS
Using a Hidden Markov Model.

Generation of the observable variable **O**



Laser Readings

- A lidar gives range distance from the sensor to nearby surfaces in different directions.
- An Mth order vector is formed using M laser readings.

$$\mathbf{S_t = [r^t_1, r^t_2 , \ldots , r^t_M]}$$

Generation of the Observations **O** Using Vector Quantization

- Given a set of range vectors, **S**, a clustering technique, vector quantization, is used to find a set of centroids that best represent the vectors.

$$S_1 = [r^1_1, r^1_2, \ldots, r^1_M]$$

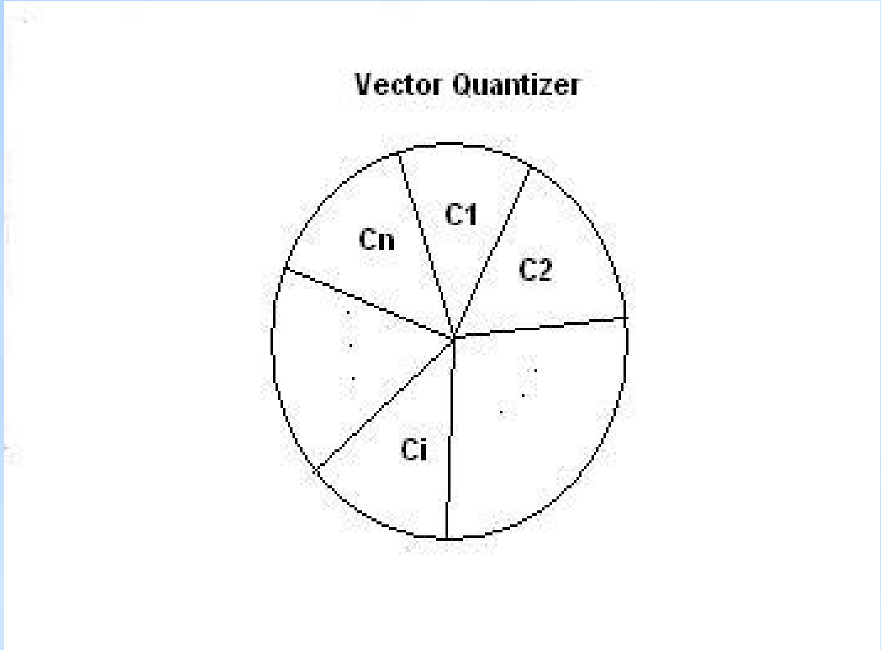
$$S_2 = [r^2_1, r^2_2, \ldots, r^2_M]$$

.

.

.

$$S_J = [r^J_1, r^J_2, \ldots, r^J_M]$$

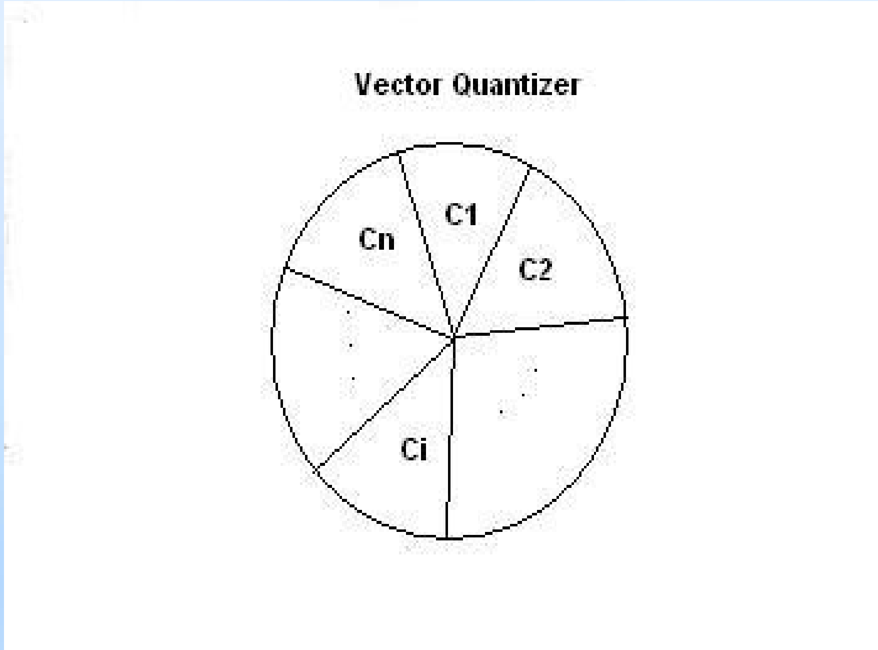


Generation of the Observations **O**

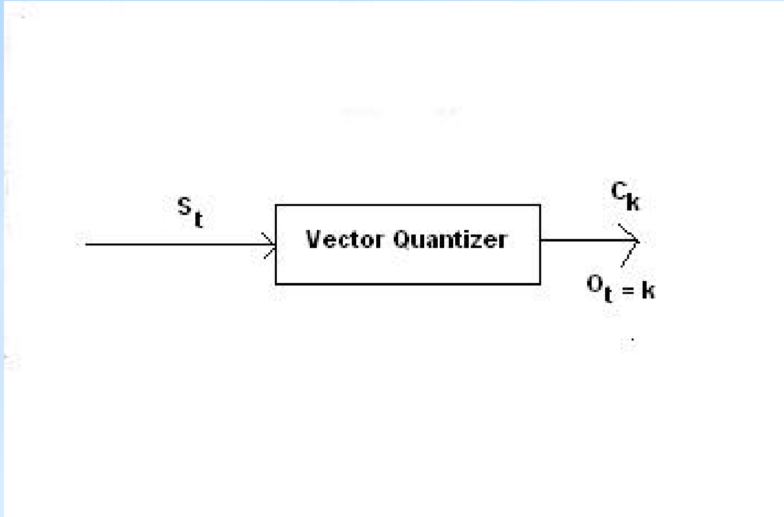
- With the set of centroids:

  C = {**C1, Ck,…,Cn**}

  the observation **O** is found by finding the centroid **Ck** that best represent the input vector $\mathbf{S_t = [r^t_1, r^t_2, \ldots , r^t_m]}$



$\mathbf{S_t = [r^t_1, r^t_2 , \ldots , r^t_m]}$



Ck ← St

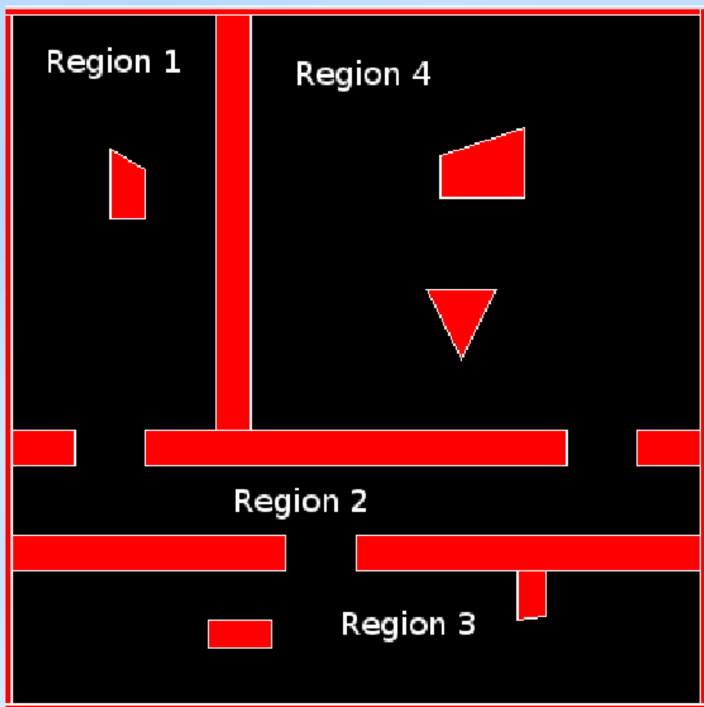
D(St,Ck) = min(D(St,Cj))

j = 1,…,Size Vector Quantizer

Ot = k

HMM Hidden Variable **X**

- The hidden random variable **X** represents the robot's orientation.
- A λ_i *HMM* model is found for each region ***i*** in the environment.



Training Behavior.

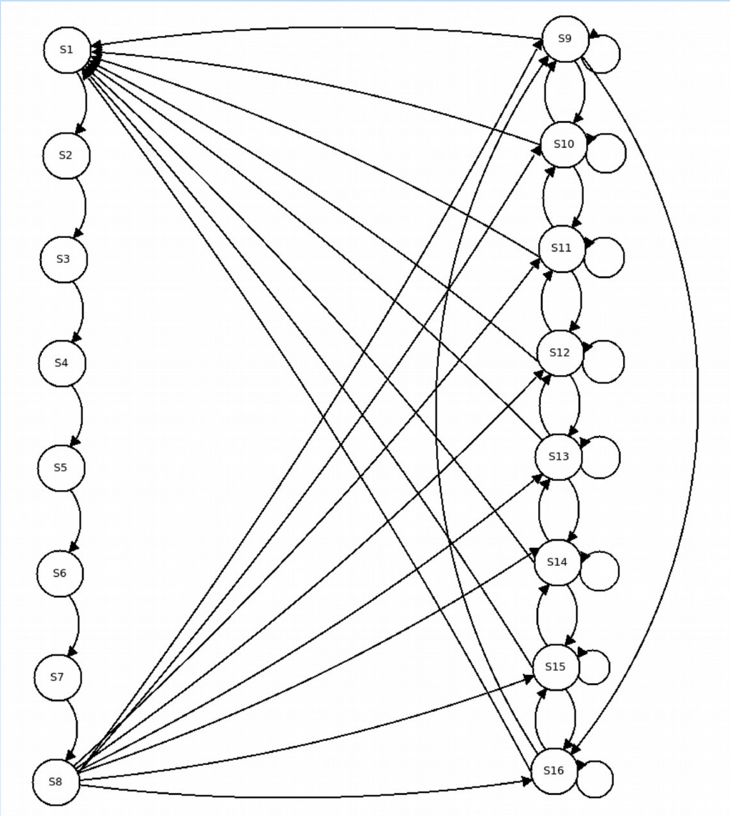
- Measuring Mode:

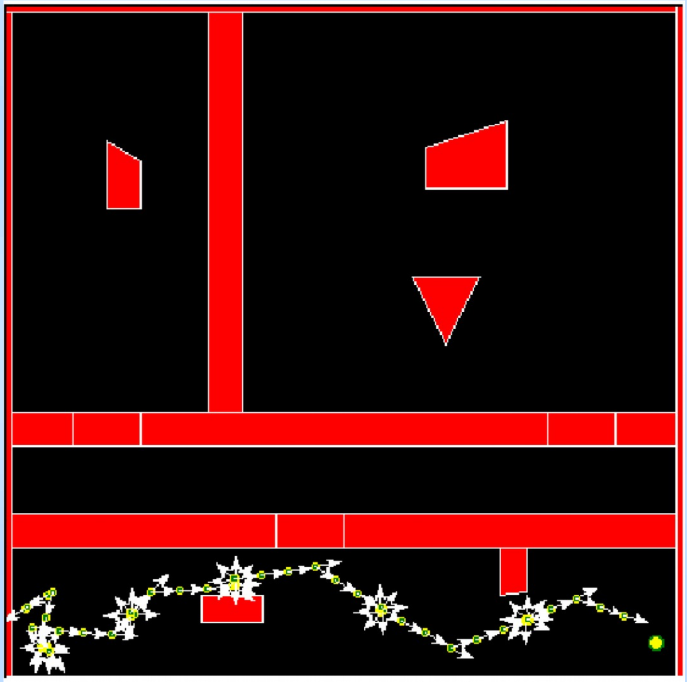
  The robot rotates incrementing of 45° at a time until back to starting point.

- When the rotation is complete the robot enter in Reactive Mode.

- Reactive Mode:

  The robot avoids obstacles with a max turning angle of 45°, every predetermined number steps go to measuring mode.





Baum-Welch Algorithm

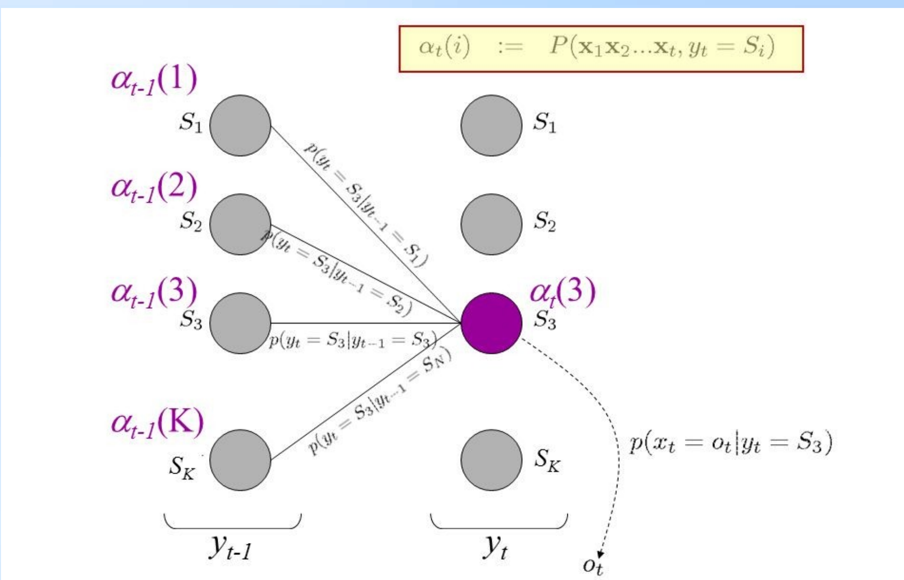
- Given an initial HMM $\lambda=[A,B,\pi]$, the goal is find an optimum $\boldsymbol{\lambda}$ that best represents the topological map that generates the set of observations **O**.
- The optimum HMM is found using the Baum-Welch algorithm. Given a number of training vectors, a λ *HMM* model is found, with the criterion to maximize $P(O|\lambda)$.

Forward Algorithm

- To find the region where the robot is, is by calculating the probability of the regions' model that most likely produced the observations:

$$\text{ModelChosen} = \text{ArgMax}[\ P(O|\ \lambda_{class})].$$

- This probability is too expensive to calculate directly (order of $N^t(ON^t)$), thus the Forward algorithm is used.



1. Initialization

$$\alpha_1(i) = \pi_i.for1 \leq i \leq N$$

2. For t=1,2,....T:

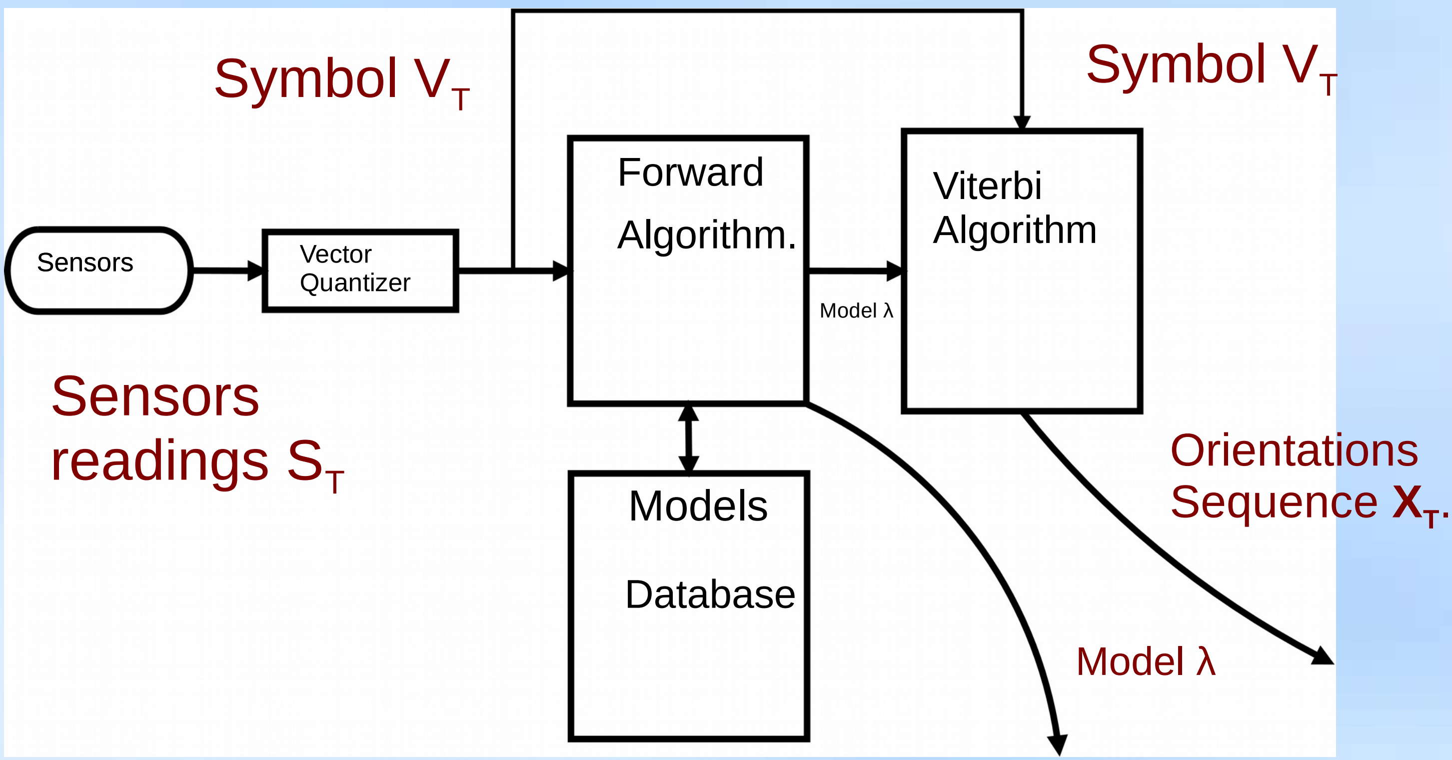
$$\alpha_{t+1}(i) = \sum_{i=1}^{N} (\alpha_t(i)a_{i,j})\, b_j(O_{t+1}).$$

3. End

$$p(O \mid \lambda) = \sum_{i \in terminalstates} \alpha_{T+1}(i) = \sum_{i=1}^{N} p(O_1, O_2, ..., O_T, q_t = i \mid \lambda).$$

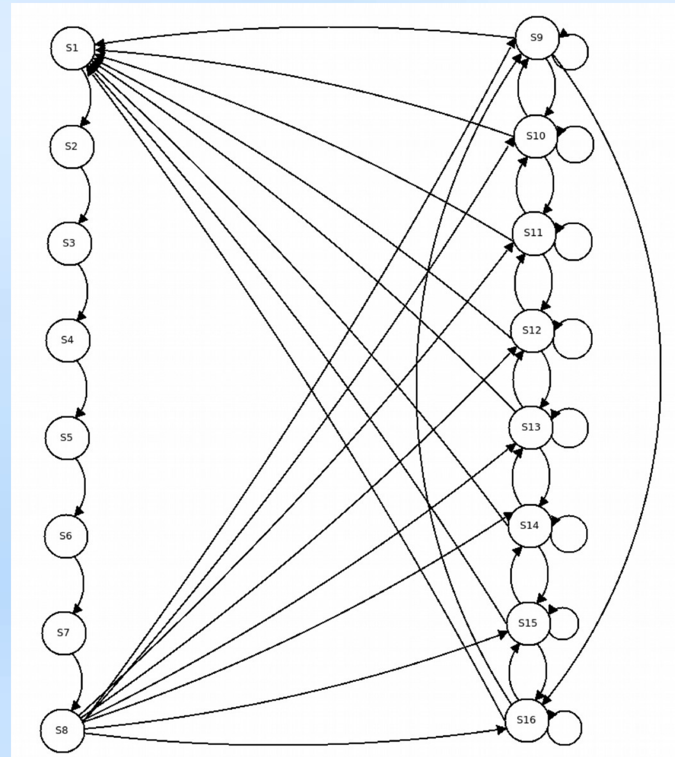
X=Orientation, system overview.

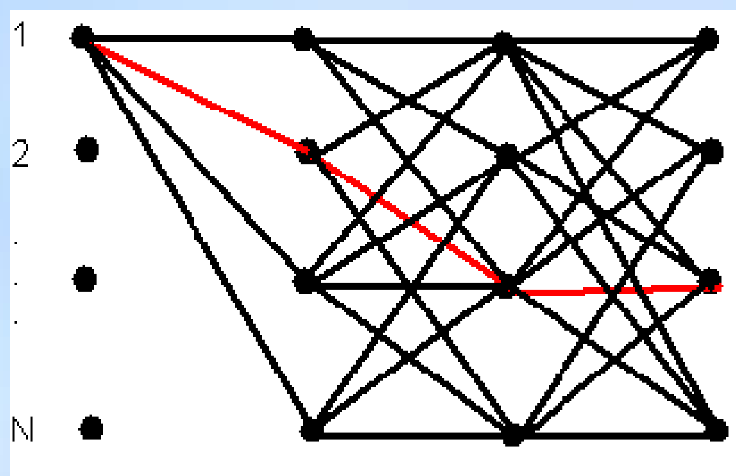
- Once a model is obtained from each region, localization is possible with the forward algorithm.



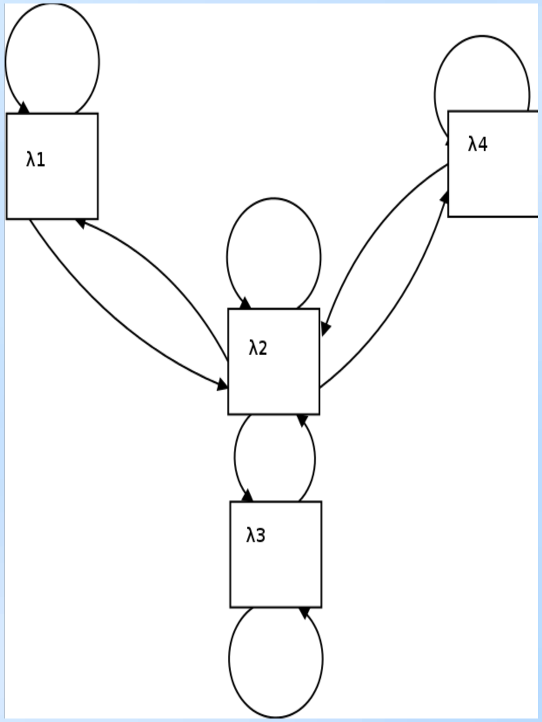
X=Orientation.

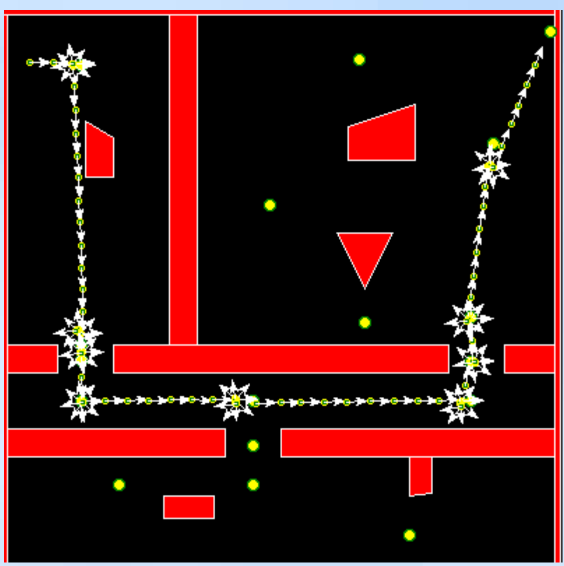
- The region is found by comparing which model most likely produced the observations.
- Once a model is chosen, the orientations sequence is obtained with the Viterbi algorithm.





HMM For Regions

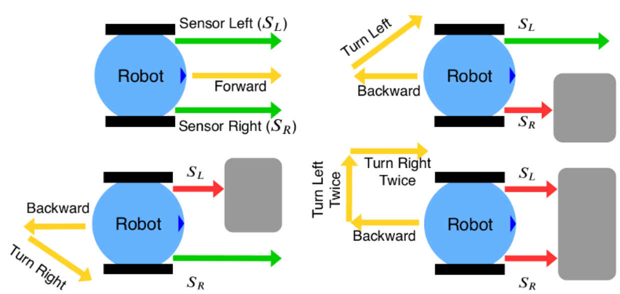


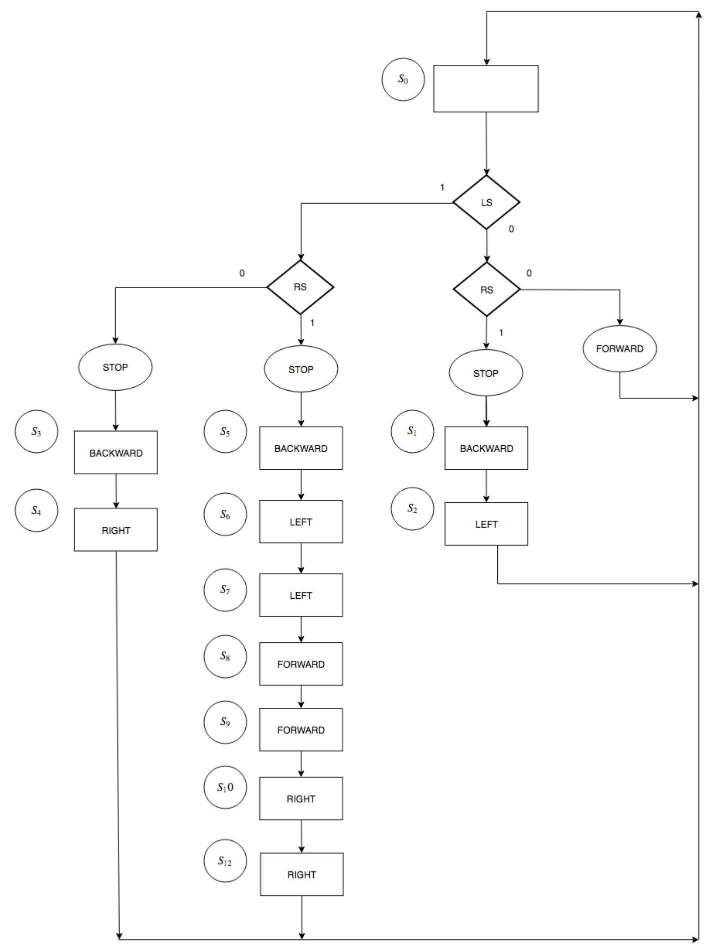


- Given a topological map of the environment an optimal node trajectory is found, using the Dijkstra algorithm, for going from a robot's origin to a destination.
- Then the Viterbi algorithm is used to find the sequence of nodes most likely traversed.
- In this trajectory the robot makes a rotation to take readings at every 45° at each topological node.

Robot Behavior Using a FSM to Avoid Obstacles

The following figure shows an algorithn state machine (ASM) for an obstacle avoidance behavior. The robot has two sensors in its left and right side to detect obstacles.





Discrete Hidden Markov Models for Robots Behaviors

In this work, we propose a probabilistic representation of a FSM through HMM using a direct matching between the FSM inputs as observations in the HMM, and the transitions from one state to another.
The following figures shows the deterministic FSM and its equivalent stochastic FSM behavior of a robot that avoids obstacles while trying to reach a light source:

Discrete Hidden Markov Models for Robots Behaviors

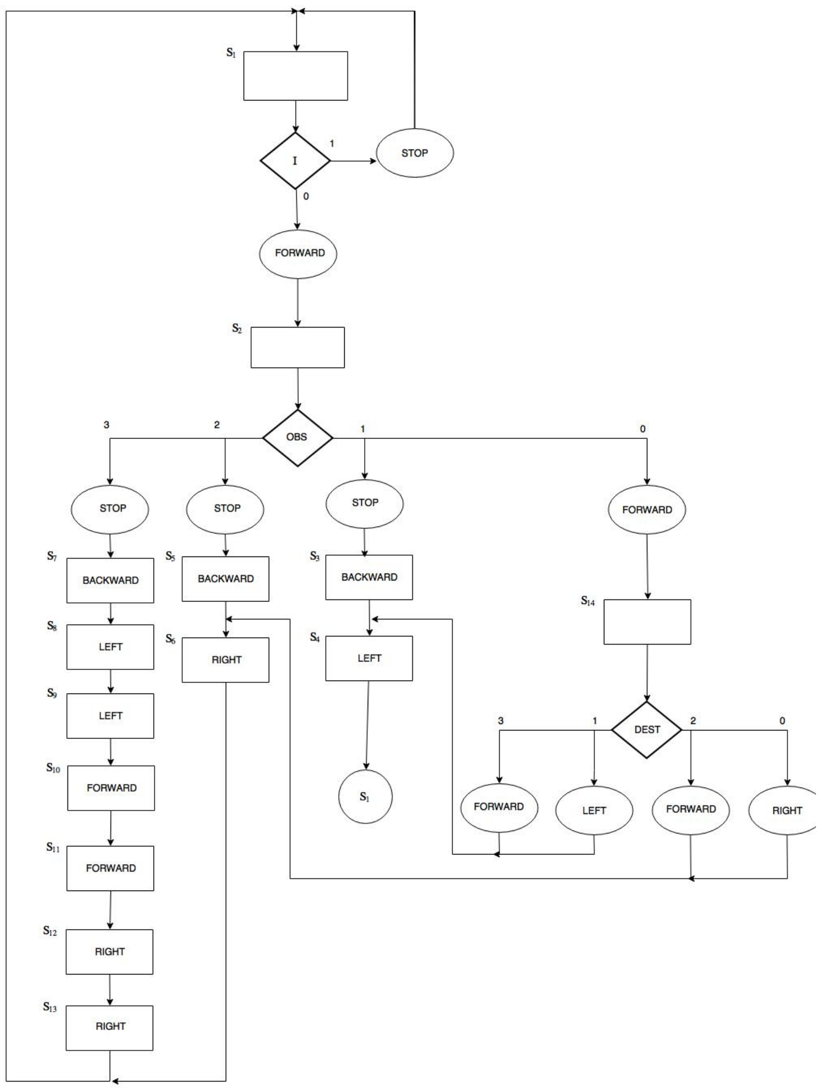


Figure: Deterministic FSM.

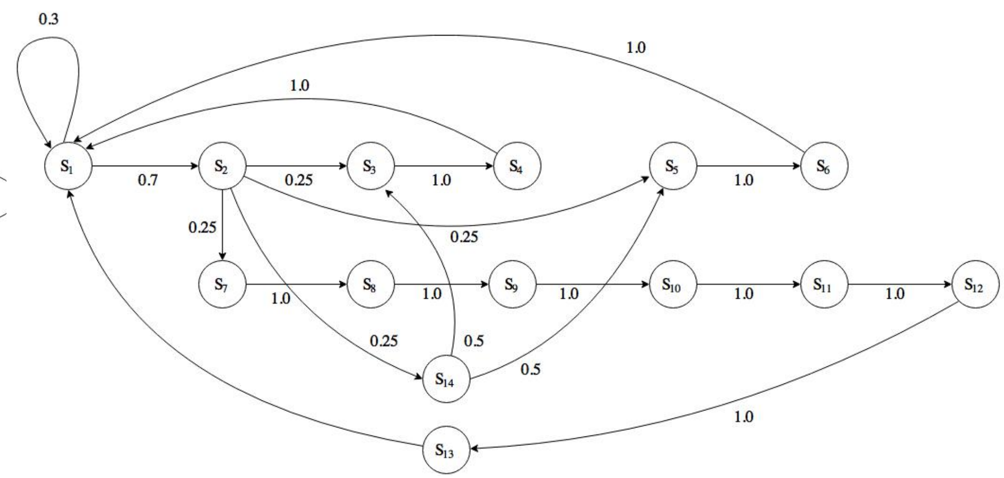
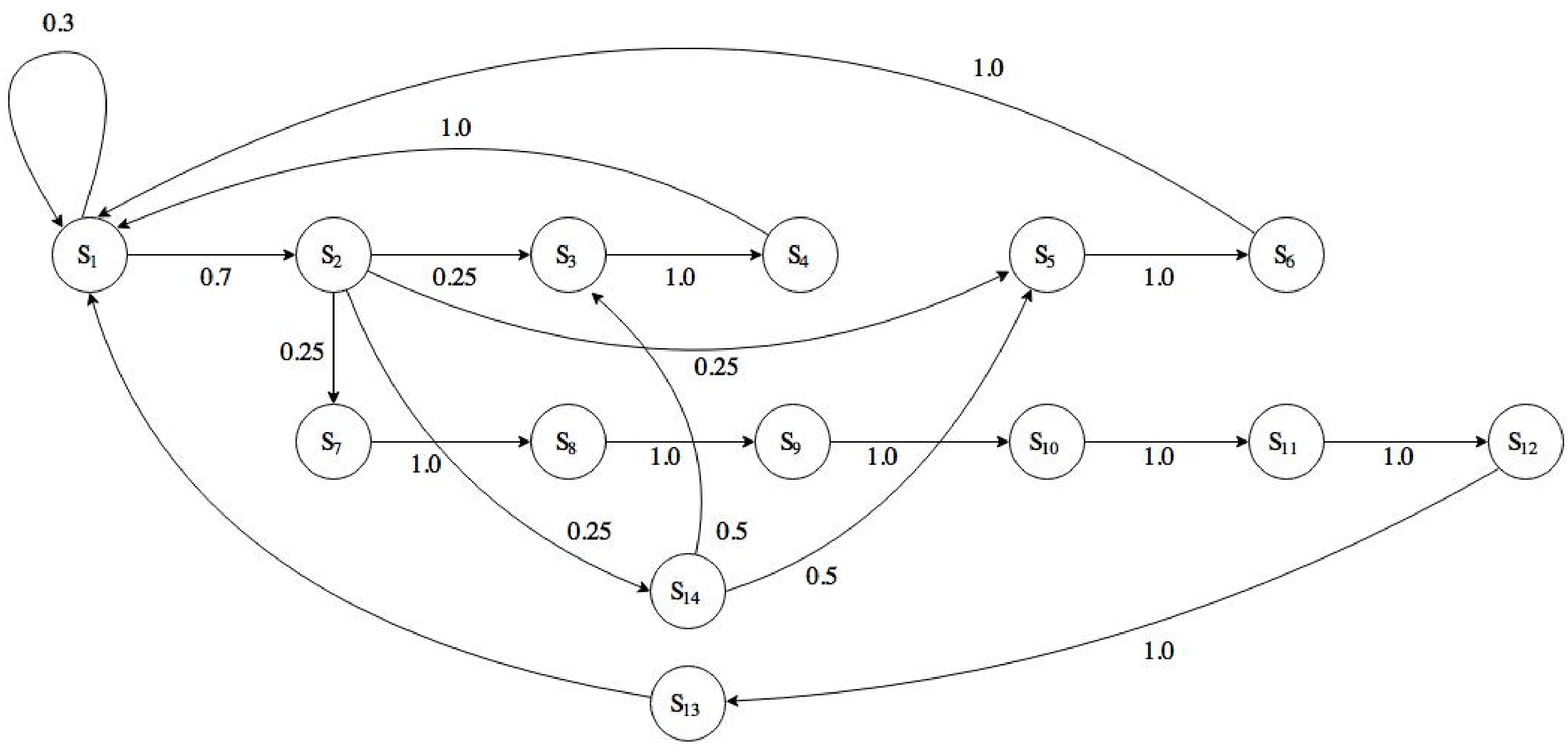


Figure: Stochastic FSM.

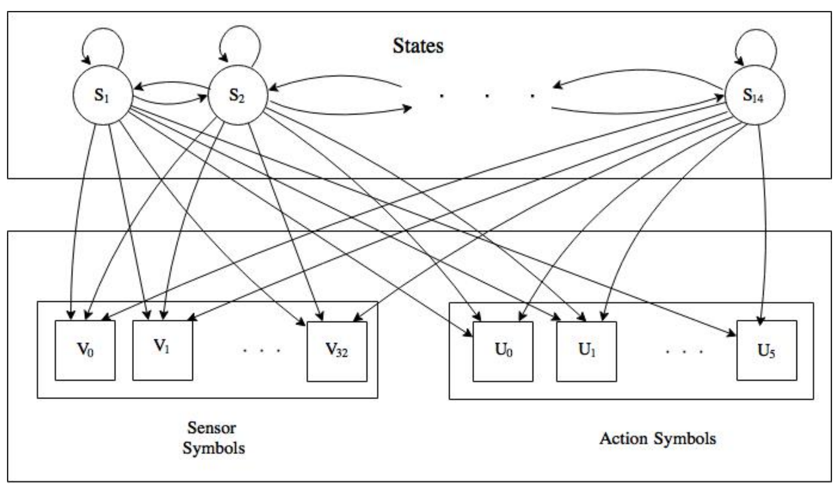
Discrete Hidden Markov Models for Robots Behaviors

At the end, the decision of moving to another state will also depend on the probabilities of the observation symbols V, obtained from the sensory data, as well as the observed symbol O_t at time t.



Action Symbols

In our approach, we incorporate the outputs of the FSM as action symbols attached to the states of the HMM. The action symbols are:
$U = \{U_0, U_1, U_2, U_3, U_4, U_5, U_6, U_7\}$, that is:
U $= \{$ stop, forward, backward, turn left (45°), turn right (-45°), turn left and go forward, turn right and go forward and turn right twice (-90°) and go forward$\}$
The following figure shows the structure of the HMM taking into consideration the sensor symbols (FSM inputs) and the action symbols (FSM outputs).



Robot Behavior Using Markov Decision Process (MDP)

The following figure shows the complete enviroment where the robot navigates and the observable enviroment with the robot's sensors:

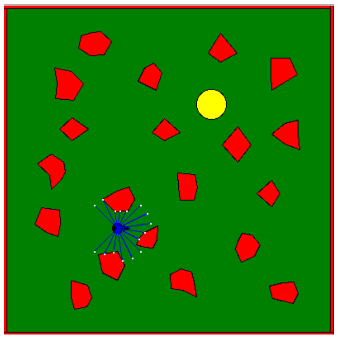


Figure: Complete environment.

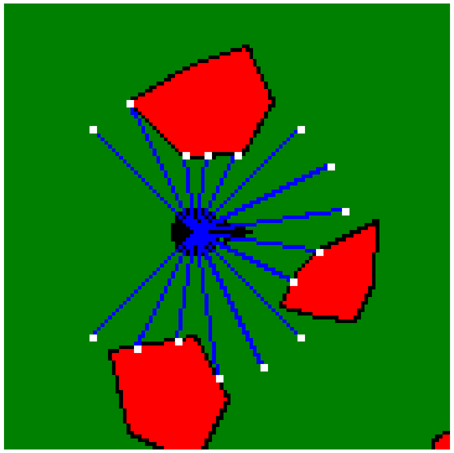
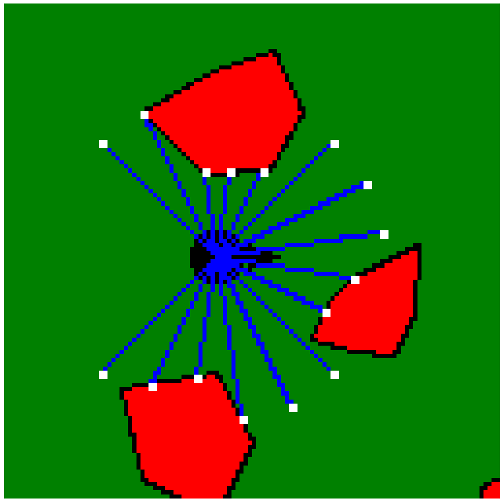


Figure: Observable environment.

In each state s the robot receives a reward for its actions, if the robot reaches an state where there is not obstacle the reward is -0.04, if there is an obstacle -1.0 and to reach a terminal state that represents the goal, it receives 2.00.

Robot Behavior Using Markov Decision Process (MDP)



Thus, for the previous observable environment, where the robot is positioned in row 3 and column 3, the following reward matrix is generated, and even the goal is outside this frame it is positioned in the upper right part of the matrix:

Table: Reward Matrix

	1	2	3	4	5	6
6	-0.04	$\otimes$	$\otimes$	$\otimes$	-0.04	2.00
5	-0.04	-0.04	$\otimes$	$\otimes$	-0.04	-0.04
4	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
3	-0.04	-0.04	0.00	-0.04	-0.04	-0.04
2	-0.04	-0.04	-0.04	-0.04	$\otimes$	$\otimes$
1	-0.04	$\otimes$	$\otimes$	$\otimes$	-0.04	$\otimes$

Robot Behavior Using Markov Decision Process (MDP)

For each state s the robot can perform the following actions:
$A(s) = \{$ stop, forward, backward, turn left (45°), turn right (-45°), turn left and go forward, turn right and go forward and turn right twice (-90°) and go forward$\}$
For each of the actions $A(s)$ there is a transition model probability matrix $P(s^{'}|s, a)$.
In the following example the robot is in the position $(2, 2)$ and facing north, if the command is to go forward the following table represents the transition model probability:

Table: Transition Model Probability from Going Forward

	1	2	3
3	0.100000	0.700000	0.100000
2	0.050000	0.000000	0.050000
1	0.000000	0.000000	0.000000

Robot Behavior Using Markov Decision Process (MDP)

There are 8 transition model probabilities corresponding to each of the actions that the robot can perform, with these an optimal policy is found:

$$\Pi^*_s = argmax_{\Pi} U^{\Pi}(s) = argmax_{(a \in A(s))} \sum_{s'} P(s'|s,a) U(s')$$

Where $U(s)$ is the utility function:

$$U(s) = R(s) + \gamma max_{(a \in A(s))} \sum_{s'} P(s'|s,a) U(s')$$

The best $U(s)$ is found solving the Bellman equation.

Robot Behavior Using Markov Decision Process (MDP)

The following table shows an example of a movements' policy Π:

Table: Movements' Policy Π

	1	2	3	4	5	6
6	$\searrow$	$\otimes$	$\otimes$	$\otimes$	$\rightarrow$	2.00
5	$\searrow$	$\searrow$	$\otimes$	$\otimes$	$\nearrow$	$\uparrow$
4	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\nearrow$	$\nearrow$	$\uparrow$
3	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\uparrow$
2	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\otimes$	$\otimes$
1	$\nearrow$	$\otimes$	$\otimes$	$\otimes$	$\nwarrow$	$\otimes$

Hidden Markov Models