PERCRO Perceptual Robotics Laboratory

RECOGNITION OF HAND GESTURES TRACKED BY A DATAGLOVE

Discriminative Training and Environment Description to Improve Recognition Performance

Vittorio Lippi, Emanuele Ruffaldi, Carlo Alberto Avizzano, Massimo Bergamasco

PERCRO laboratory, Scuola Superiore Sant'anna, Pisa, Italy

v.lippi@sssup.it, e.ruffaldi@sssup.it, carlo@sssup.it, bergamasco@sssup.it

ANNIIP Workshop
Saturday, July 4

PERCRO Perceptual Robotics Laboratory

ABSTRACT

PERCRO Perceptual Robotics Laboratory

ABSTRACT

- This is a gesture recognition system exploiting discriminative training algorithm based on Maximal Mutual Information (MMI) and the integration of environment information.
- The environment is described through a set of fuzzy clauses, on the basis of which a priori probabilities are computed.
- Adaptive systems such as unsupervised neural networks are used to build a codebook of symbols representing the hand's states.

PERCRO Perceptual
Robotics Laboratory

SOLUTION OVERVIEW

ANNIIP Workshop
Saturday, July 4

PERCRO Perceptual Robotics Laboratory

SOLUTION OVERVIEW

SYSTEM WITH TWO ANALYSIS BLOCKS

Sequence Recognition

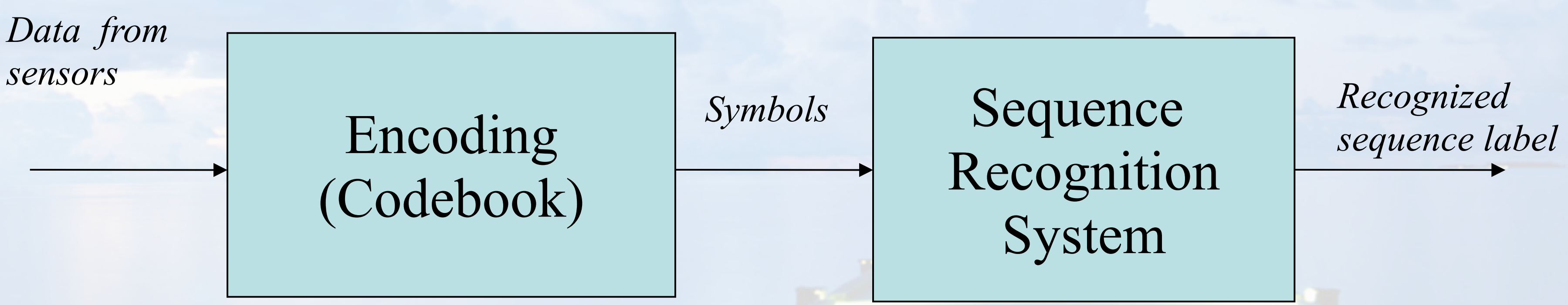
based on
HMM / MMI

Environment Description

Fuzzy logic
Inference Motor

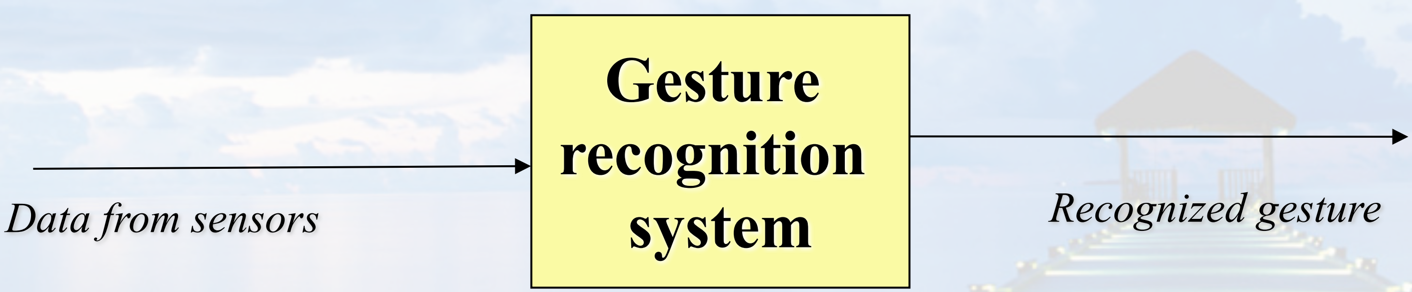
Sequence Recognition

- The sequence recognition is implemented through a classifier made of two components arranged in cascade:
  - A symbolic encoding of input data
  - A sequence recognition system
- The Encoding stage transforms the input data stream into a sequence of symbols
- The recognition system is a classifier of sequence of symbols

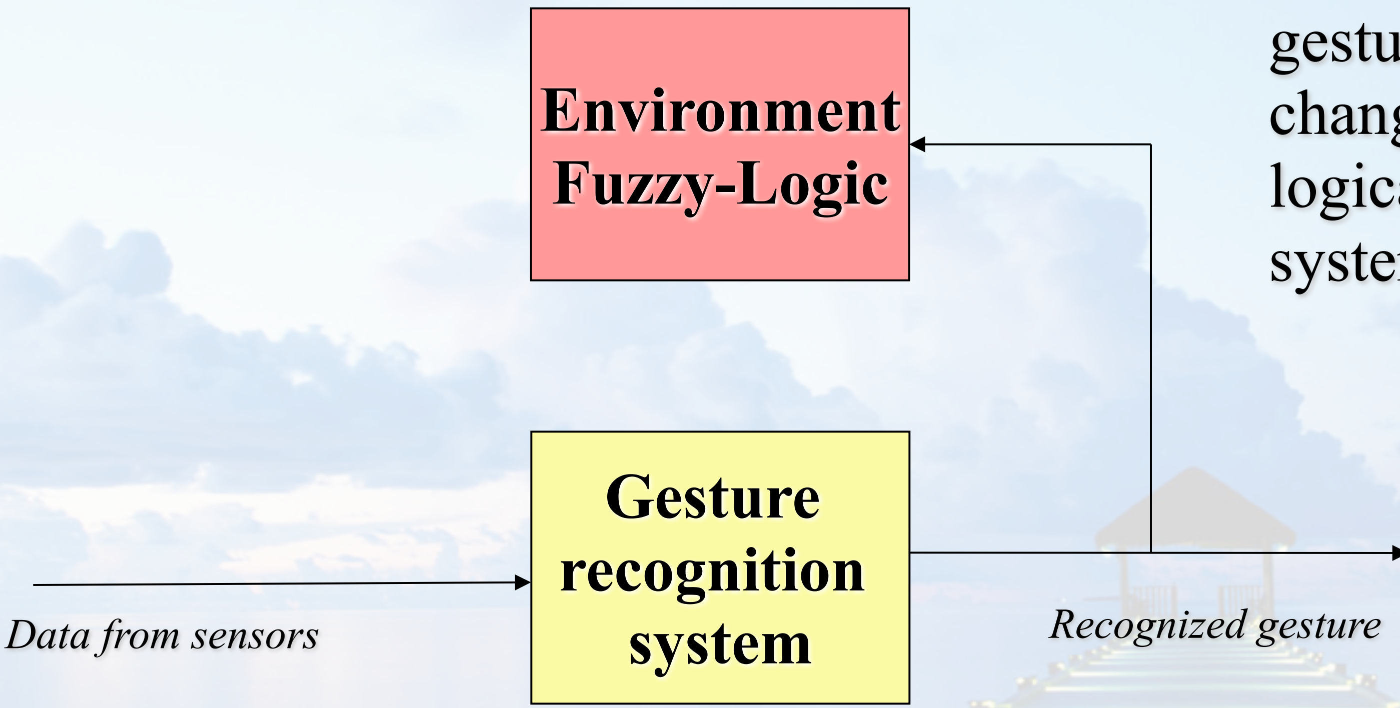


SOLUTION OVERVIEW

The recognition system takes data from sensors as input and returns a label identifying the recognized gesture

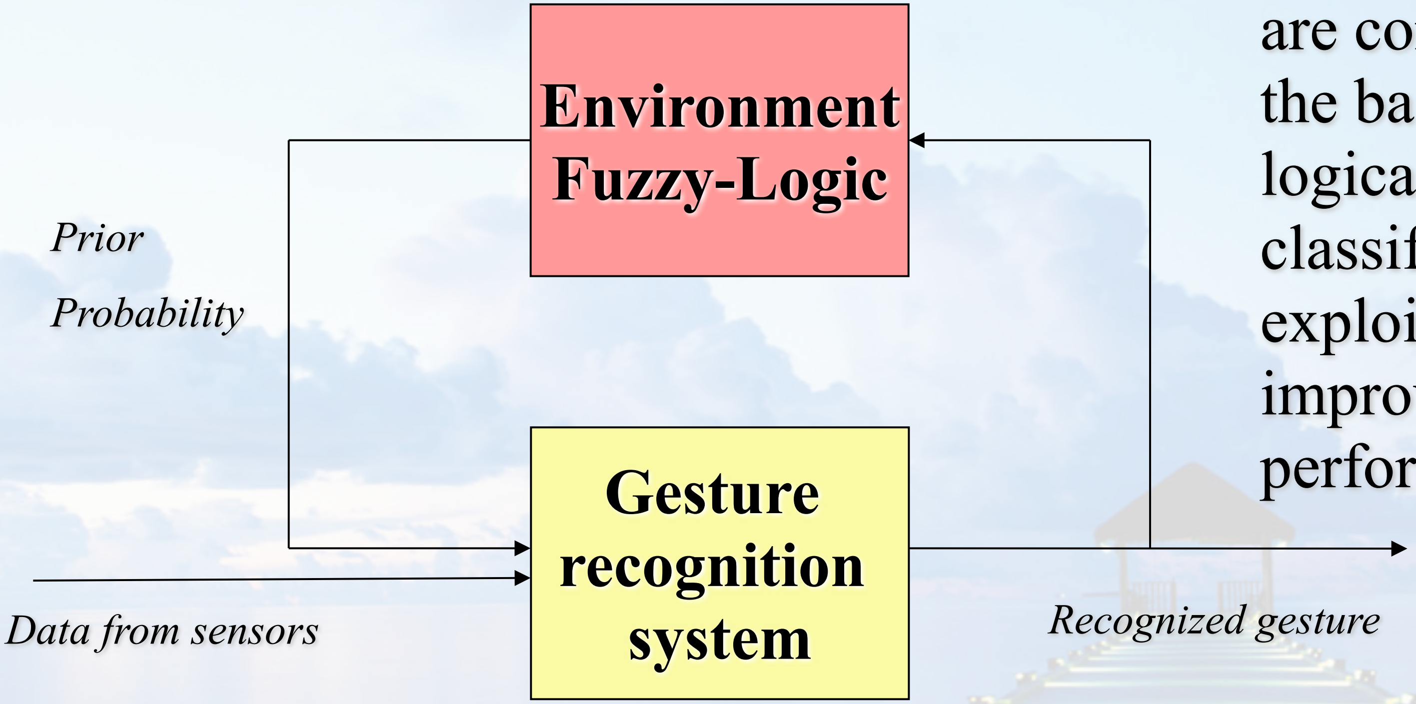


SOLUTION OVERVIEW



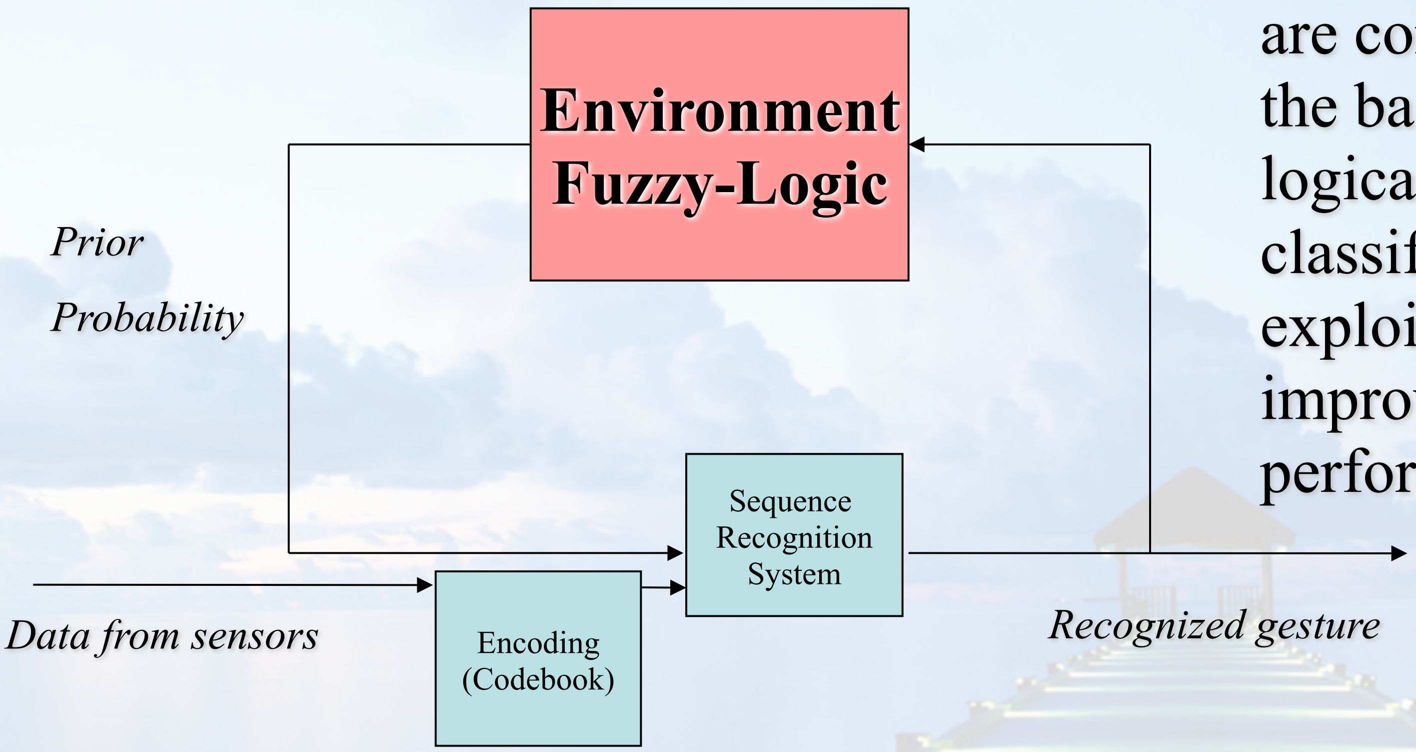
The recognized gesture produces a change in the logical state of the system

SOLUTION OVERVIEW



Prior probabilities are computed on the basis of the logical state: the classifier can exploit them to improve the performance

SOLUTION OVERVIEW



Prior probabilities are computed on the basis of the logical state: the classifier can exploit them to improve the performance

PERCRO Perceptual
Robotics Laboratory

COMPONENTS

The codebook

PERCRO Perceptual Robotics Laboratory

THE CODEBOOK

The encoding stage has the following purposes:

- Perform an early analysis extracting significant data.
- Reduce the quantity of data to be sent to the classifier .
- Adapt data representation to the selected classifier

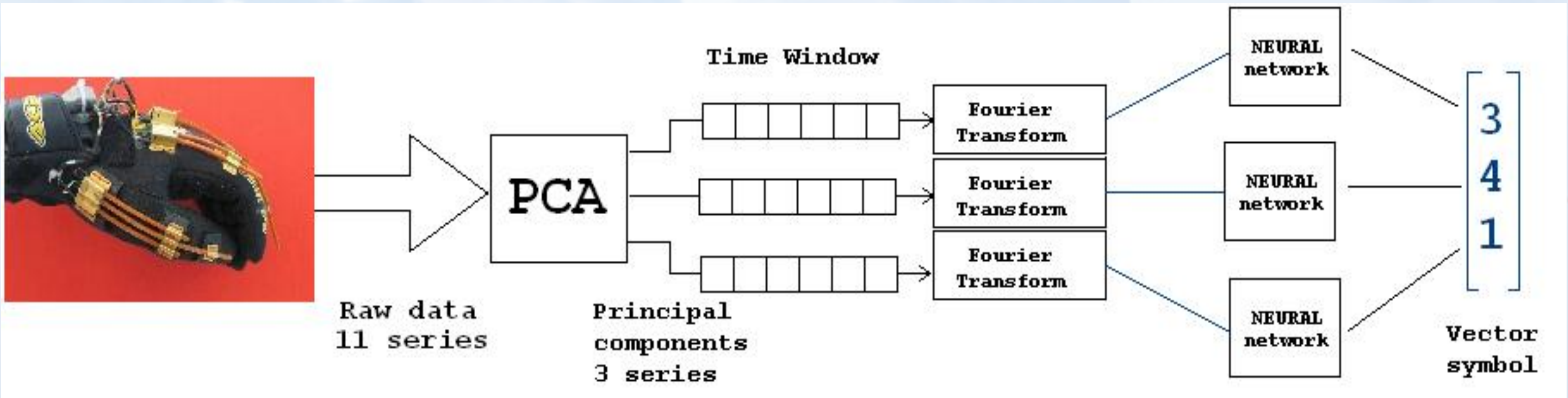
THE CODEBOOK

- Captured data are a sequence of vectors of the observed variables.
- The number of variables is generally high.
- The information gathered from sensor is often redundant

THE CODEBOOK

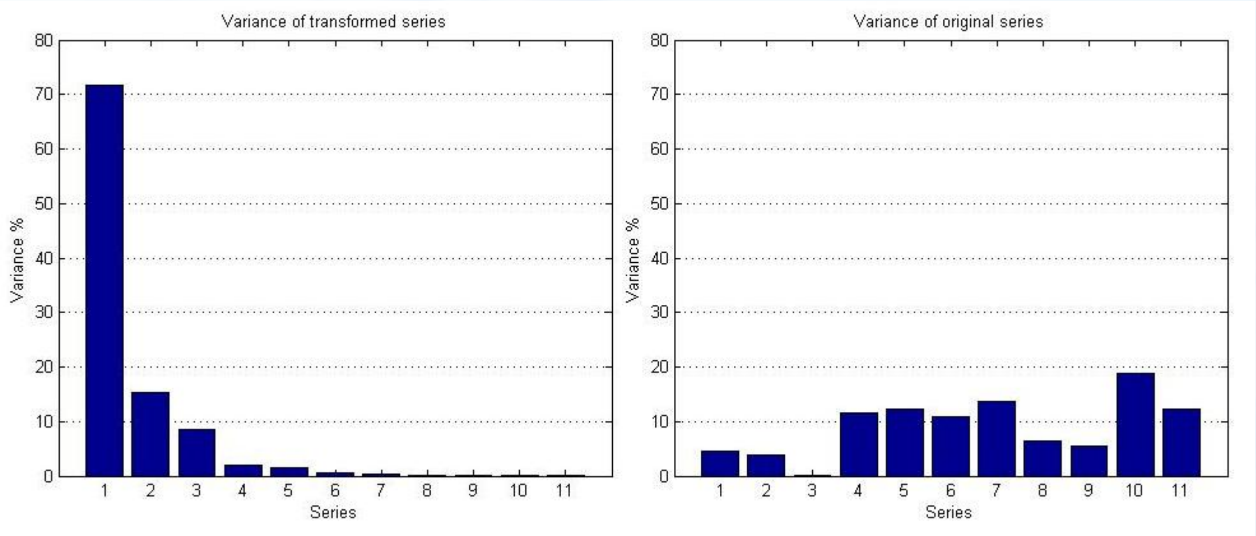
The encoding is performed through the following steps:

1. PCA: To reduce the number of components keeping the loss of information acceptable.
2. Building component vector series: it is a time window.
3. Fourier transform (with some filtering)
4. Process with neural networks to produce symbols



PCA OF DATA FROM DATAGLOVE

- 11 variables x 100Hz.
- Samples of 5 kind of gestures produced with an hand.



- Three series are enough to keep the 90% of the variance.

FREQUENCY ANALYSIS

- A discrete fourier transform DFT is applied to data sequence.
- The DFT

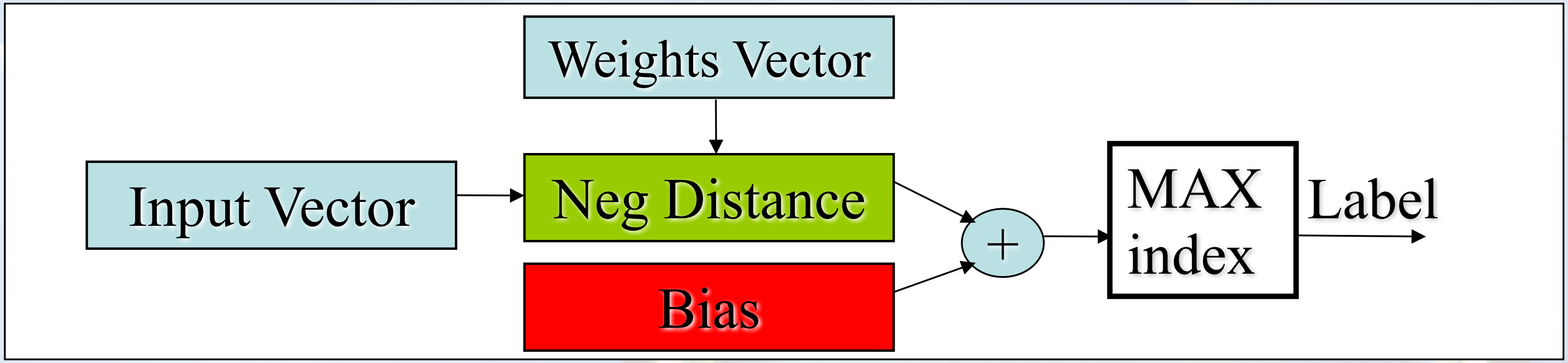
$$X_q = F(x_n) = \sum_{k=0}^{N-1} x_k e^{-j\frac{2\pi}{n}kq}$$

Produces a vector X with the same number of components of the vector x

- **The first component X_0 is dropped from the vector**.
- **It is a way to give to the movement more than to the position.**

NEURAL NETWORKS

- The last stage of the encoder is a set of competitive neural networks (Kohonen, T., Self-Organizing and Associative Memory, New York: Springer-Verlag, 1984).
- Each network takes one of the sequence produced in the previous steps as input.
- Neural networks are adapted with a non supervised algorithm.
- This stage works as a classifier, associating labels to sets of input vectors



PERCRO Perceptual Robotics Laboratory

COMPONENTS

Sequence Classifier

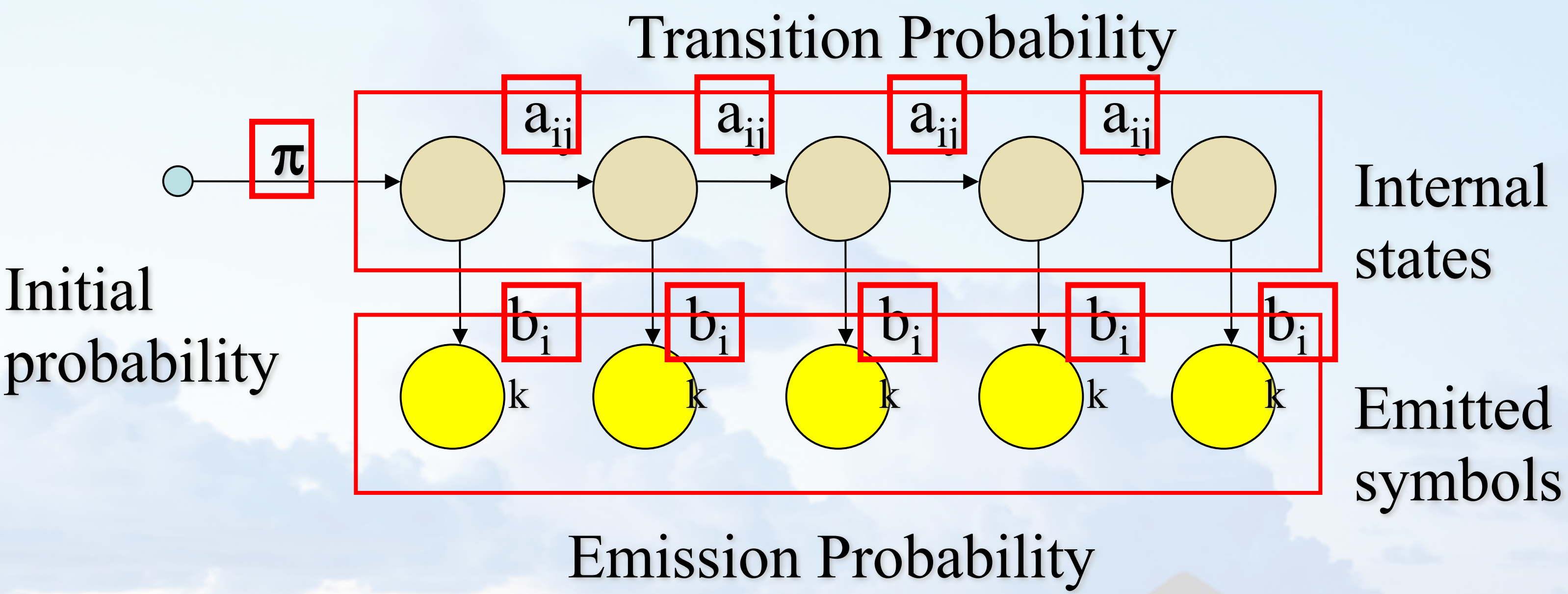
PERCRO Perceptual
Robotics Laboratory

SEQUENCE CLASSIFIER

- HMM are trained through examples in order to model the process to recognize
- Every HMM represent a stochastic process that could produce a sequence as output
- A given input sequence is assigned to the class associated to the HMM with the highest probability to emit it

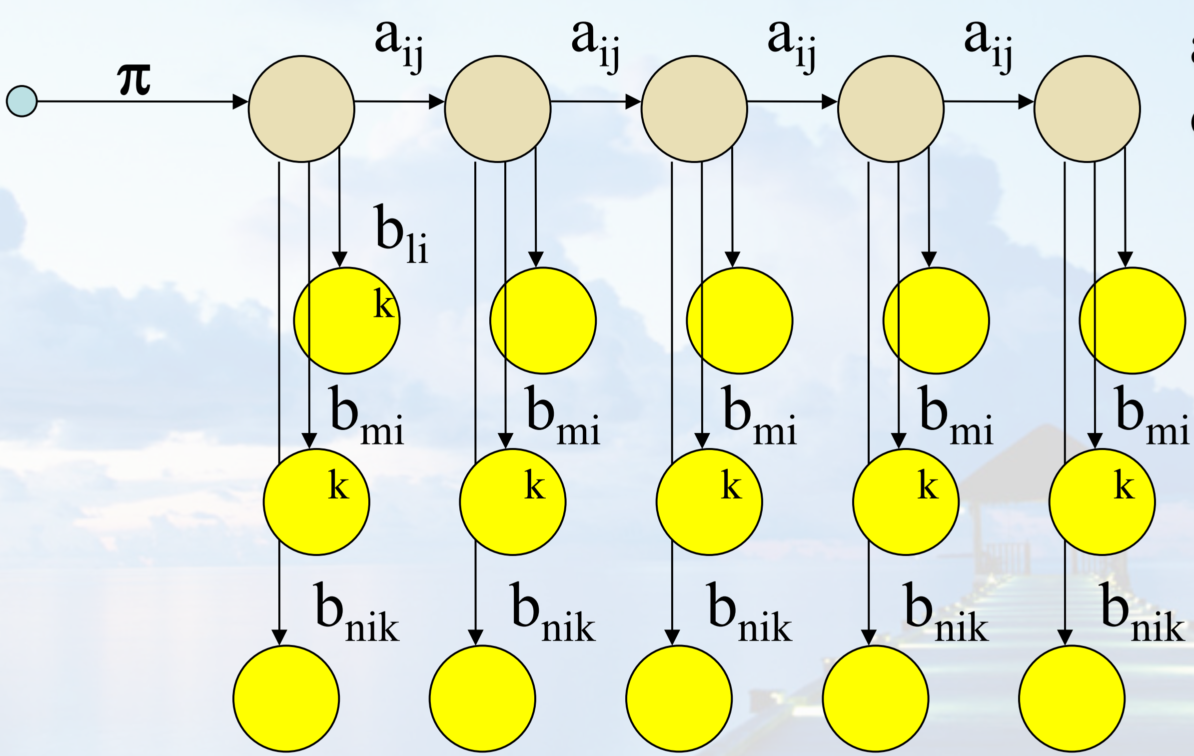
PERCRO Perceptual
Robotics Laboratory

HIDDEN MARKOV MODELS



Parameters are probability distributions and hence subject to the proper contstraints

HMM EXPLOITED



This model allows multiple observables

PARAMETER ADAPTATION

- Adapting the parameters of an HMM to maximize the emission probability of a given sequence is a well known problem for which several algorithms have been proposed in literature.
- **Maximization of likelihood** is the basis of the most of them.
- Although representing a well-known and (relatively) easy principle to match the model to the distribution of examples is not optimal for classification problems in the general case

PERCRO Perceptual Robotics Laboratory

DISCRIMINATIVE PARAMETERS ADAPTATION

- The Maximum Mutual information principle or MMI is applied
- The method has been successfully applied to speech recognition problems

PERCRO Perceptual Robotics Laboratory

MAXIMUM MUTUAL INFORMATION

- MI between the model M and the sequence S is

$$I(M_i;S) = \log \frac{P_\theta(S,M_i)}{P_\theta(S)P(M_i)} = \log P_i(S|M_i) - \log(P(S))$$

- Terms marked with θ are dependent on system parameters

PERCRO Perceptual Robotics Laboratory

MAXIMUM MUTUAL INFORMATION

- Expressing it as

$log(P_i(M_i|S))$- $log(P(M))$,

- *and* considering **P(M)** independent from parameters makes evident that maximizing the MI means to maximize the probability ***$log(P_i(M_i|S))$ that the observed sequence*** *S*, has been produced by M_i

MAXIMUM MUTUAL INFORMATION

- There is not an EM method such as baum-welsch for MMI
- The gradient algorithm is hence applied
- The maximized function is

$$F_\theta = \sum_i \sum_k I_\theta(M_i; S_k)$$

- The gradient is

$$\nabla_\theta F_\theta = \sum_i \nabla_\theta \sum_k I_\theta(M_i; S_k)$$

- With the single components, given a sequence S

$$\frac{\partial I(M;A)}{\partial \theta_i} = \frac{\frac{\partial P_\theta(S|M)}{\partial \theta_i}}{P_\theta(S|M)} - \frac{\sum_{\hat{M}} \frac{\partial P_\theta(S|\hat{M})}{\partial \theta_i} P(\hat{M})}{P_\theta(S)}$$

PERCRO Perceptual Robotics Laboratory

MAXIMUM MUTUAL INFORMATION

$$\frac{\partial I(M;A)}{\partial \theta_i} = \frac{\frac{\partial P_\theta(S|M)}{\partial \theta_i}}{P_\theta(S|M)} - \frac{\sum_{\hat{M}} \frac{\partial P_\theta(S|\hat{M})}{\partial \theta_i} P(\hat{M})}{P_\theta(S)}$$

- The first term is the derivative of the Likelihood function
- The gradient computed for a model is independent from other models parameters
- The second term produces a penalization for the probability of the sequence S to be emitted by the model M.
- *This is applied to Negative Examples*

PERCRO Perceptual Robotics Laboratory

NEGATIVE EXAMPLES

- Negative examples are defined on the basis of the probability of being emitted by the wrong model
- A parameter α between 0 and 1 is set
- A sequence S_j, associated to the model M_k becomes a negative example for the model M_i when

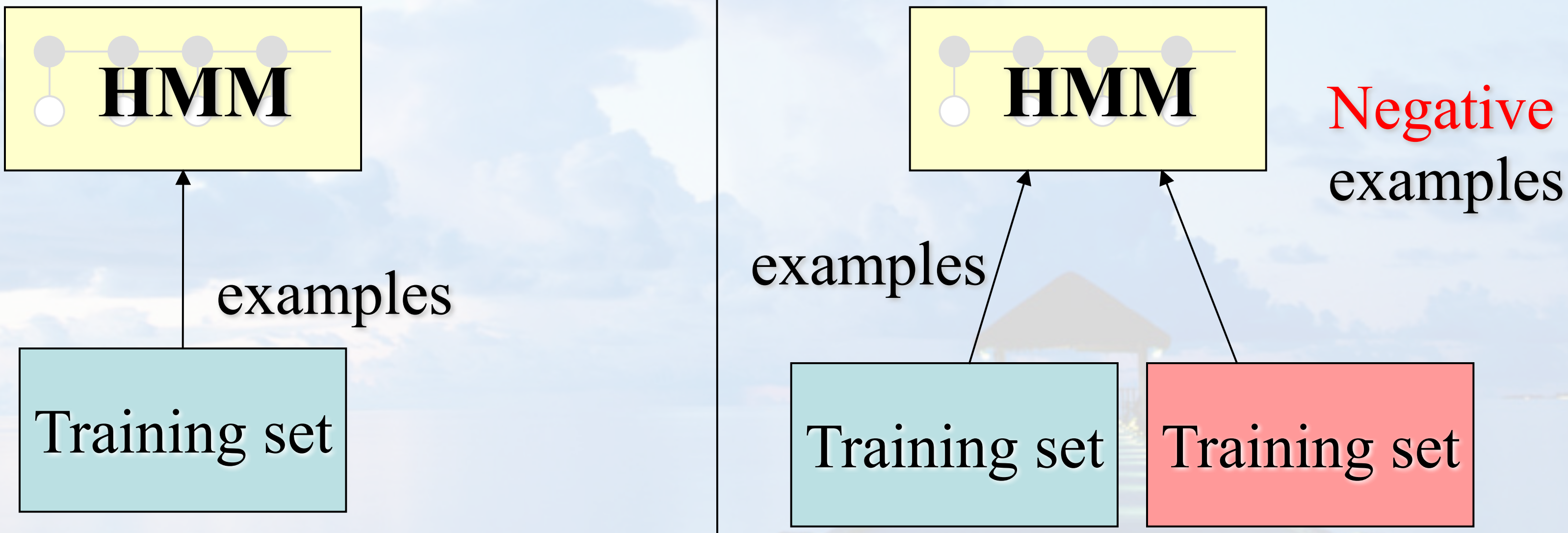
$$P(S_j \mid M_i) > \alpha P(S_j \mid M_k)$$

- The negative examples change at every step of the algorithm.

DISCRIMINATIVE PARAMETERS ADAPTATION

Maximum Likelihood

Maximum Mutual Information



PERCRO Perceptual Robotics Laboratory

COMPONENTS

Fuzzy logic model for the environment

ENVIRONMENT LOGICAL DESCRIPTION

- An ad hoc defined language is used to describe a logical model of the environment.
- Objects,classes of objects and proposition about both objects and classes can be defined using a simplied of predicative logic.
- Fuzzy logic is particularly suitable to represent physical concepts like proximity or alignment

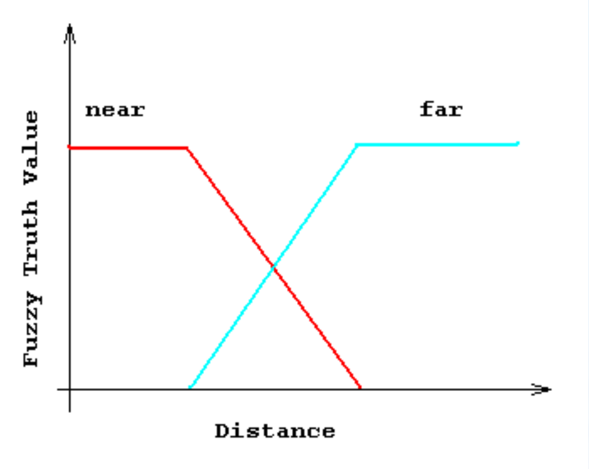
SYNTAX

The available syntax allows to describe the environment through:

- set of statements with their truth values, needed to describe environment state i.e., ”the hand is near the screw, with truth value 0.5”.
- A set of rules to compute the truth values of statements i.e., ”if a screw is screwed on a bolt then that screw is blocked”.
- A set of rules to compute the probability of a gesture to be performed.
- A set of rules to define the consequences of an action performed by the user i.e. ”if the user performs the action 'Pick' and the hand is near an object that object is then in-hand with truth value 1”;

SYNTAX

- Statement have a truth value that ranges from 0 to 1



- Logical operators are extended to fuzzy values by the following rules

A AND B = min(value(A),value(B))

A OR B = max(value(A),value(B))

SYNTAX

Example:

```
function screwed_on(2)
function blocked_on(2)

object bolt
bolt bolt1
object screw
screw screw1
fact screwed_on(screw1,bolt1)  1
rule screwed_on(screw #1,bolt #2) => blocked_on(screw #1,bolt #2)
```

PERCRO Perceptual Robotics Laboratory

SYNTAX

Example:

action screwing:

set_on(screw #1,bolt #2) AND near(hand,screw #1)=>screwed_on(screw #1,bolt #2)

action unscrewing:

near(hand,screw #1)=>~screwed_on(screw #1,bolt #2)

priors screwed_on(bolt1,screw2)

Screwing 0.8

Unscrewing 0.2

priors near(hand,screw)

Picking 0.5

Screwing 0.5

PERCRO Perceptual Robotics Laboratory

ENVIRONMENT LOGICAL DESCRIPTION

The inference engine computes truth values of arbitrary statements from the logical description. When the truth value of a statement is required, the following actions are performed:

1. The statement is searched in the description, if it is present its truth value is returned and the research is ended;
2. A rule having the state as effect is searched. If found the procedure is repeated for all the terms in the cause. The value of truth is than computed and returned.
3. If a value is not found in previous steps, 0 is returned

NOTES

- Function in this system are intended from objects to truth values, not from object to object as usually intended in predicative logic. The set of facts is hence a set of proposition or *datalog*
- Facts not represented and not inferable from other facts are considered 0
- The same rule is never used twice in the same demonstration three (anti-cycle rule)

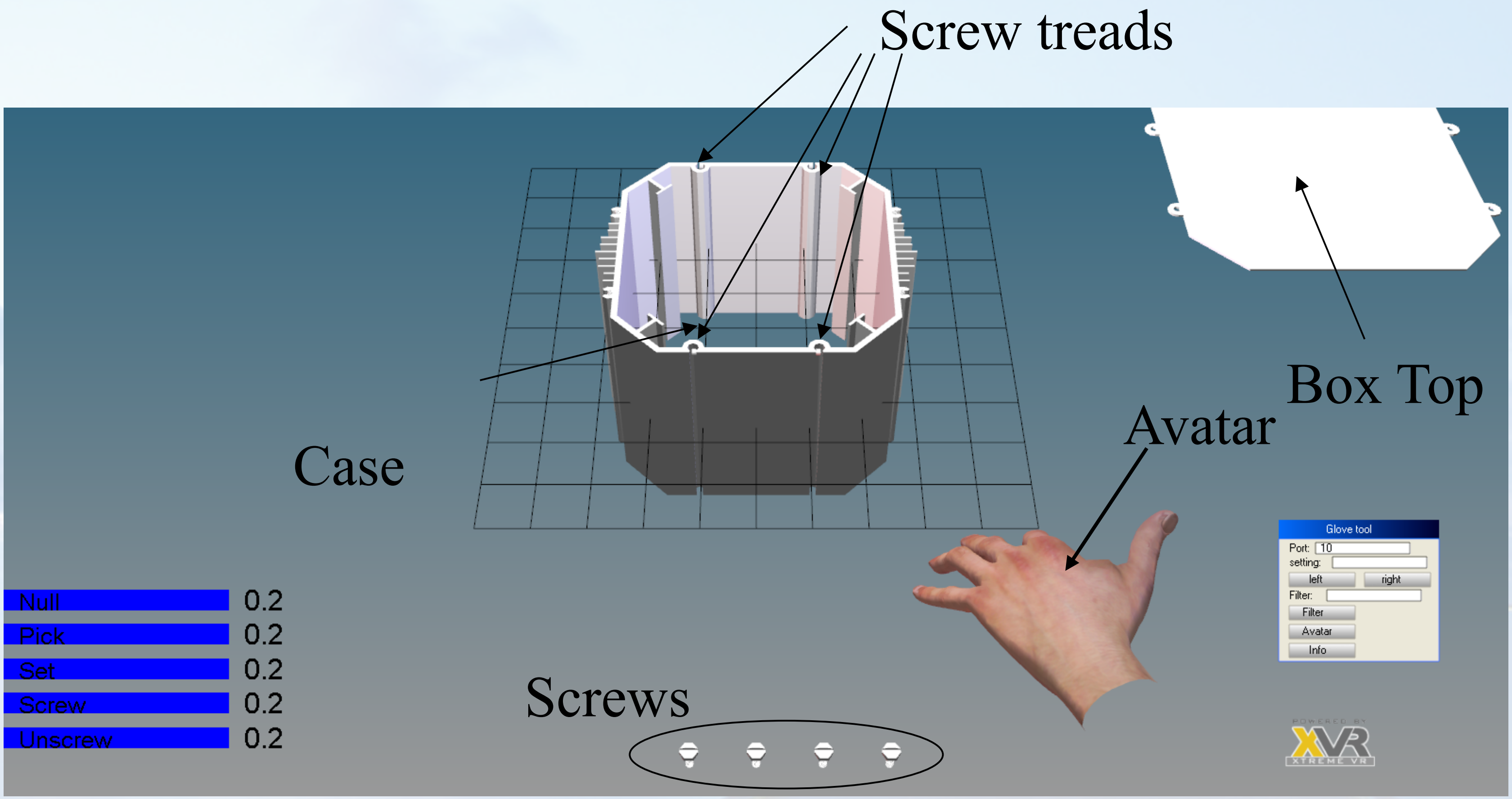
- **This assures that a truth value is always computed in a finite time**

PERCRO Perceptual Robotics Laboratory

AN APPLICATION EXAMPLE

PERCRO Perceptual
Robotics Laboratory

VIRTUAL REALITY ELEMENTS



PERCRO Perceptual
Robotics Laboratory

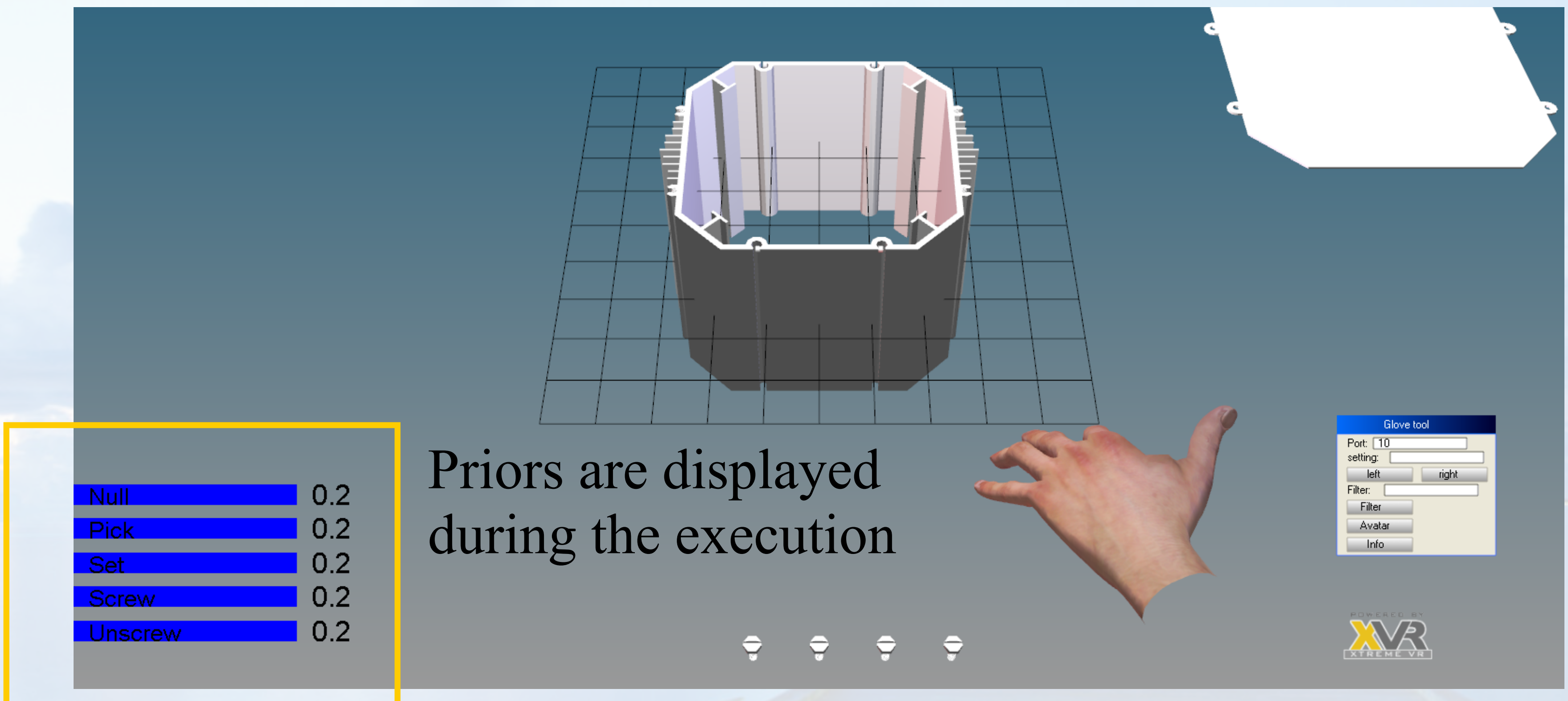
CAPTURE SYSTEM

Percro Dataglove

Hall effect sensors

Sensors end

PRIORS


Null 0.2
Pick 0.2
Set 0.2
Screw 0.2
Unscrew 0.2
Priors are displayed during the execution
Glove tool
Port: 10
setting:
left
right
Filter:
Filter
Avatar
Info
XVR

PERCRO Perceptual
Robotics Laboratory

ENVIRONMENT LOGICAL DESCRIPTION

Functions

function screwed_on(2)
function set_on(2)
function near(2)
function dummy(1)
function inhand(1)
function blocked_on(2)

ENVIRONMENT LOGICAL DESCRIPTION

Objects

object hand
hand lefthand
object screw
screw screw1
screw screw2
screw screw3
screw screw4

object box
box box1
object base
base base1
object hole
hole hole1
hole hole2
hole hole3
hole hole4

ENVIRONMENT LOGICAL DESCRIPTION

Actions and rules

rule screwed_on(screw #1,hole #2) AND set_on(base1,box1) => blocked_on(base1,box1)

action null: dummy(lefthand) => dummy(box1)

action picking: near(lefthand,screw #1) AND set_on(screw #1,hole #2) => ~set_on(screw #1,hole #2)
: near(lefthand,ANY #2) => inhand(ANY #2)
: near(lefthand,ANY #2) AND inhand(ANY #2) => ~inhand(ANY #2)

action setting: near(lefthand,screw #1) AND near(screw #1,hole #2) => set_on(screw #1,hole #2)

action screwing: near(lefthand,screw #1) AND set_on(screw #1,hole #2) => screwed_on(screw #1,hole #2)

action unscrewing: near(lefthand,screw #1) AND screwed_on(screw #1,hole #2) => ~screwed_on(screw #1,hole #2)

PERCRO Perceptual Robotics Laboratory

ENVIRONMENT LOGICAL DESCRIPTION

Prior probability vectors

priors dummy(lefthand)
null 0.2
picking 0.2
setting 0.2
screwing 0.2
unscrewing 0.2

priors inhand(screw) AND ~near(lefthand,hole)
null 0.5
picking 0.5

priors inhand(screw) AND near(lefthand,hole)
null 0.3
picking 0.2
setting 0.5

priors inhand(base1) AND near(lefthand,box1)
null 0.5
setting 0.5

priors inhand(base1) AND ~near(lefthand,box1)
null 0.5
picking 0.5

priors near(lefthand,screw #1) AND ~inhand(screw #1) AND ~screwed_on(screw #1,hole #2) AND ~inhand(base1) AND ~set_on(screw #1,hole #2)
null 0.4
picking 0.6

priors near(lefthand,screw #1) AND ~inhand(screw #1) AND ~screwed_on(screw #1,hole #2) AND ~inhand(base1) AND set_on(screw #1,hole #2)
null 0.3
picking 0.2
screwing 0.5

priors near(lefthand,screw #1) AND ~inhand(screw #1) AND screwed_on(screw #1,hole #2) AND ~inhand(base1)
null 0.5
unscrewing 0.5

priors near(lefthand,base1) AND ~inhand(base1) AND ~blocked_on(base1,box1)
null 0.4
picking 0.6

PERCRO Perceptual Robotics Laboratory

EFFICIENCY

PERCRO Perceptual
Robotics Laboratory

EFFICIENCY

- The system has been tested using examples of the 5 gestures produced by 5 users.
- Every user has been asked to perform every gesture for a minute.
- To build the codebook the 11 angular values describing the fingerpositions, sampled at 1 MHz are reduced to 3 series of data through a PCA, then the Fourier transform is applied on overlapping time windows of 50 samples.
- The resulting sample series have been divided in sequences of 8 symbols. The system has been validated splitting the set of examples into 5 subset, used 4 by 4 as training set

PERCRO Perceptual Robotics Laboratory

EFFICIENCY

- Confusion matrix for recognized gestures using ML training:

	null	pick	set	screw	unscrew
null	91,50%	1,81%	1.01%	3,36%	2,32%
pick	8,00%	66,80%	7,10%	13,10%	5,00%
set	6,00%	10,10%	70,20%	4,80%	8,90%
screw	7,00%	8.00%	3,00%	66,00%	16,00%
unscrew	6,80%	6,40%	6,14%	13,16%	68,00%

Accuracy :73%

PERCRO Perceptual Robotics Laboratory

EFFICIENCY

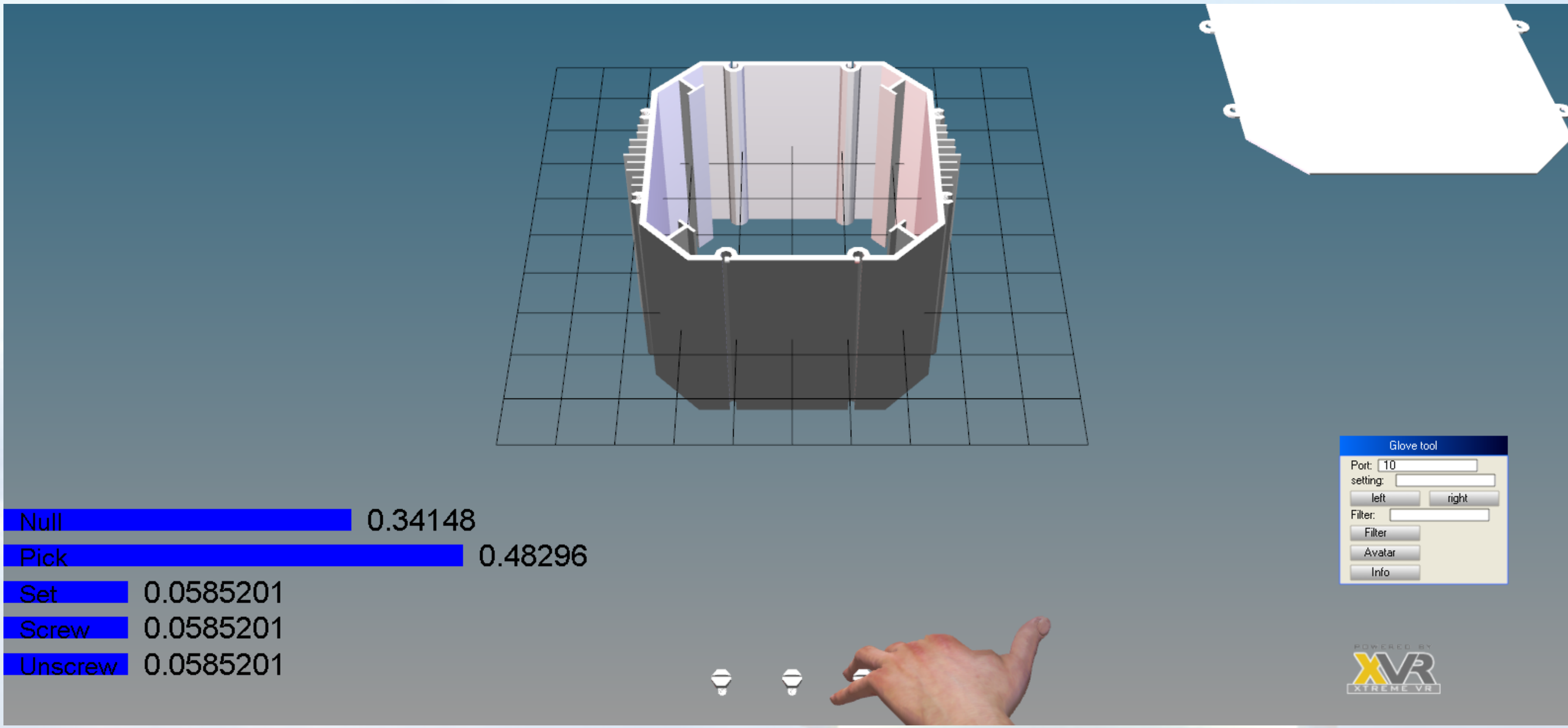
- Confusion matrix for recognized gestures:

	null	pick	set	screw	unscrew
null	94,65%	0,00%	5,35%	0,00%	0,00%
pick	0,00%	92,65%	0,00%	3,67%	3,67%
set	6,07%	1,62%	89,07%	2,02%	1,21%
screw	0,84%	4,20%	0,00%	83,19%	11,77%
unscrew	0,00%	0,80%	0,00%	9,56%	89,64%

Accuracy :89.9%

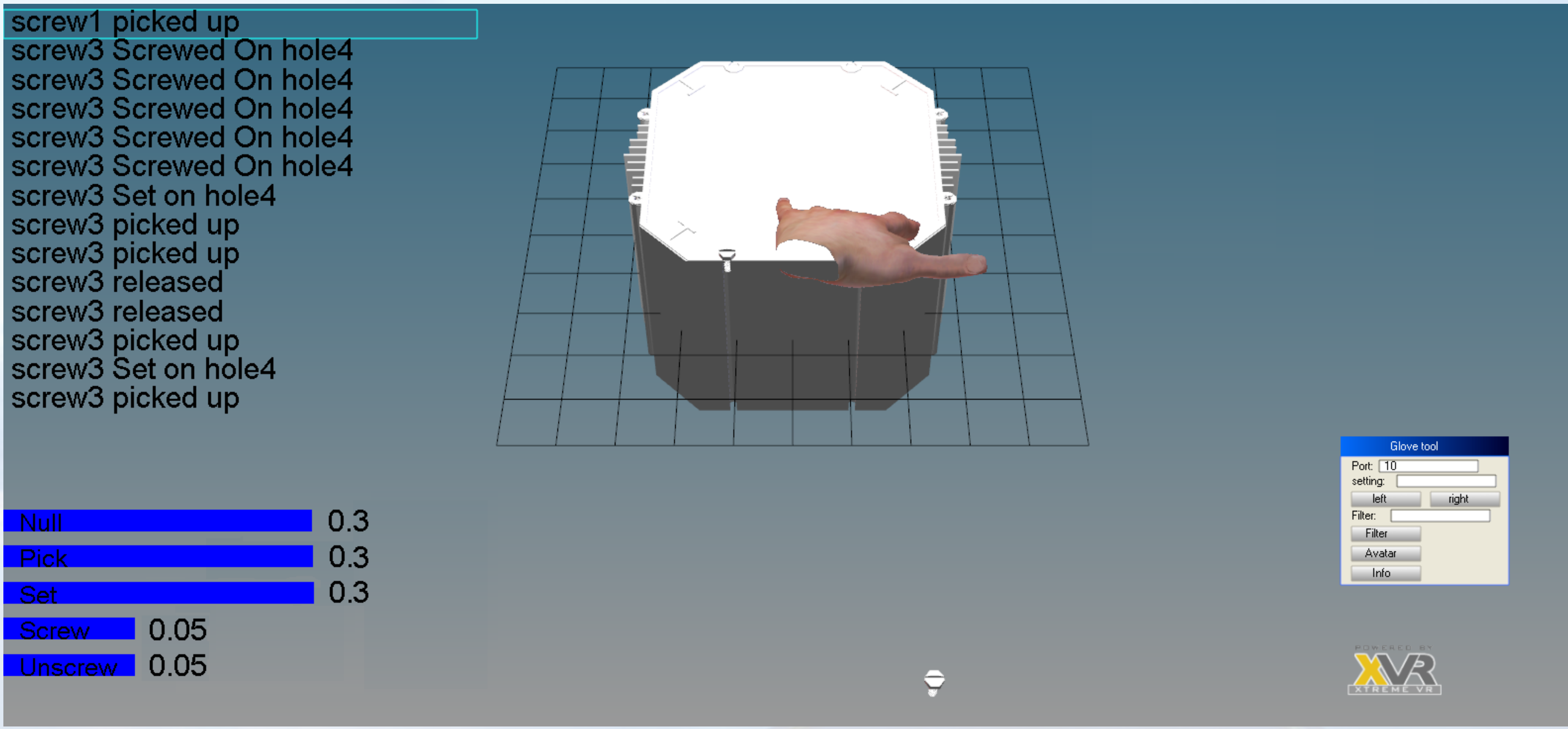
PERCRO Perceptual
Robotics Laboratory

EFFICIENCY



PERCRO Perceptual
Robotics Laboratory

EFFICIENCY



PERCRO Perceptual Robotics Laboratory

EFFICIENCY

Null/Pick/Screw – Confusion Matrix

(Hand near to a non screwed screw)

	Null	Pick	Unscrew
Null	100,00%	0,00%	0,00%
Pick	0,00%	93,47%	6,53%
Unscrew	0,84%	2,10%	97,06%

Accuracy: 96.83%

Null/Screw/Unscrew – Confusion Matrix

(Hand near to an half screwed screw)

	Null	Screw	Unscrew
Null	100,00%	0,00%	0,00%
Screw	1,68%	86,56%	11,77%
Unscrew	0,00%	9,96%	90,04%

Accuracy: 92.21%

PERCRO Perceptual Robotics Laboratory

SUMMARY

A test shown how adapting parameters with MMI produces a better classification performance compared with the usual ML-based training presented in literature, expecially in the classification of similar gestures.

The algorithm high computational requirements have been compensated selecting a subset of the training samples at every step, without affecting the recognition performance of the algorithm.

The introduction of environment information produced a further general performance improvement. The proposed fuzzy logic simplified inference system assures an efficient realtime update of environment description according to user actions.

PERCRO Perceptual
Robotics Laboratory

THANKS.