Microsoft

Microsoft® Research

Faculty Summit 2012

Riviera Maya, Mexico | May 23-25 | In partnership with CONACYT

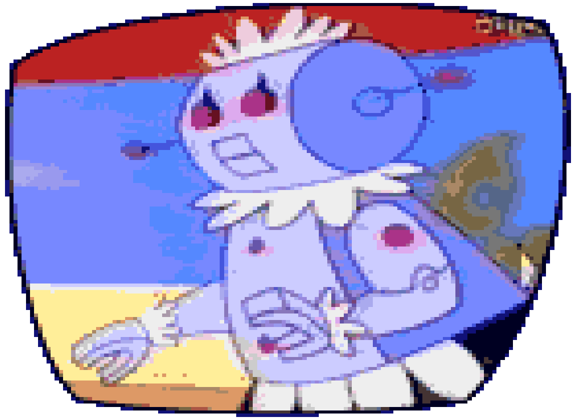
Teaching A Robot How To Perform New Tasks

Eduardo F. Morales
Instituto Nacional de Astrofísica, Óptica y Electrónica

May 24th, 2012

Outline

- Motivation
- Learning framework
- Teaching options:
  - The user controls the robot
  - The user tells the robot
  - The user shows the robot
- Experimental results
- Conclusions and future work



Service Robots

Guides – museums, exhibitions, …

Home – lawn, vacuum cleaner, eldely people, …

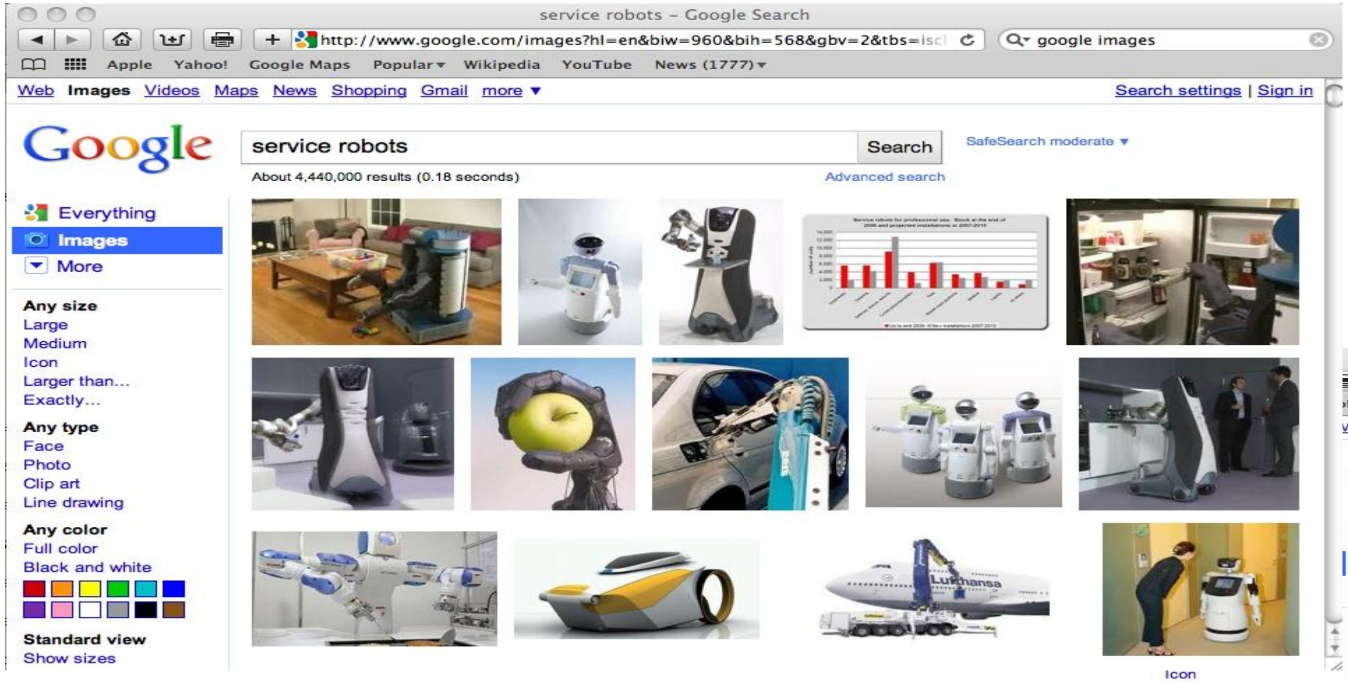
Rescue – locate survivals in natural disaste

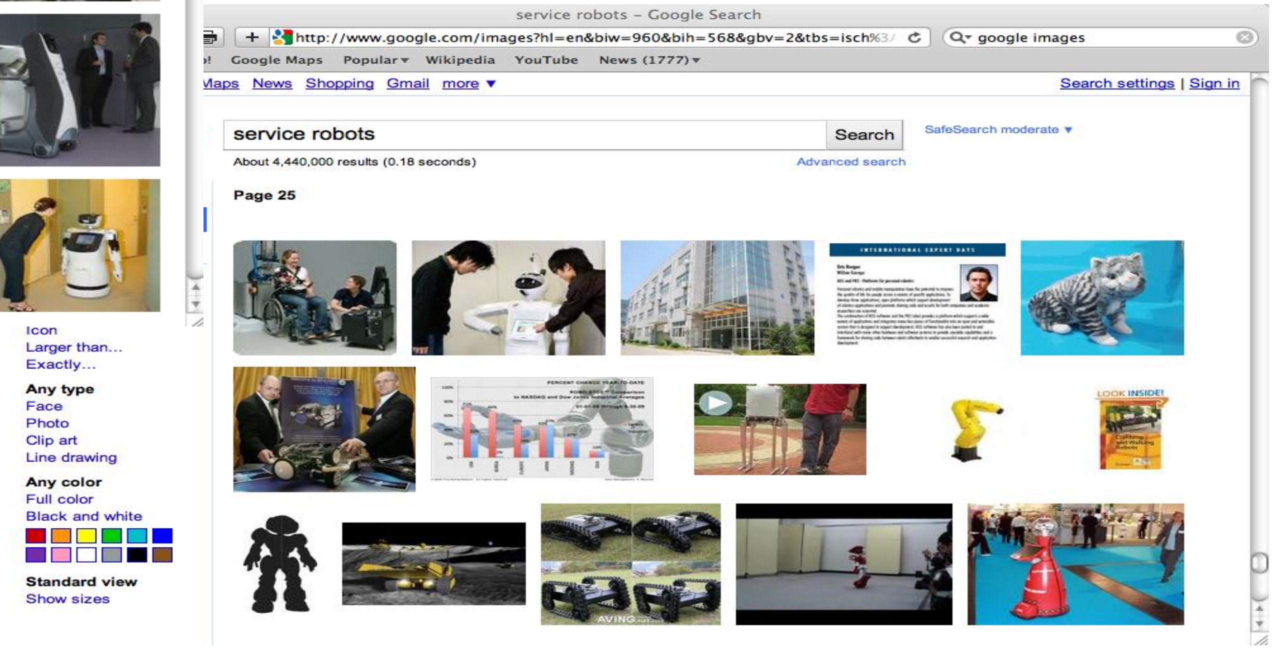
…

Exploration – volcanoes, sea reefs, planets, …

Some Examples

More Examples





Why Teach A Robot New Tasks?

To prevail robots will have:

- To adapt to their environment
- To satisfy user's needs
- *Extend their capabilities to tasks for which they were not programmed*
- *Users will have to teach them in a natural way*

General Learning Scheme

The user provides traces (state-action sequences) of how to perform a task

The traces may be noisy, sub-optimal, represent different strategies, diff. users,…

A reinforcement learning algorithm (with some adjustments) is used to find a *suitable* policy in a *reasonable* time

Teaching Options

We are considering three teaching options:

The user controls the robot (*joystick*) [Julio]

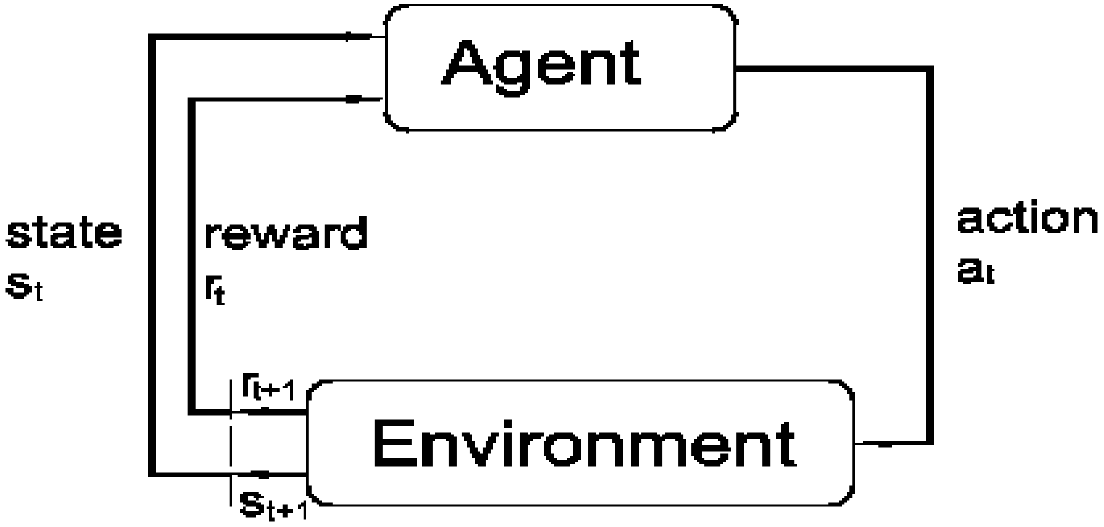
The user commands the robot (*voice*) [Ana]

The user performs the task (*vision*) [Luis Adrián]

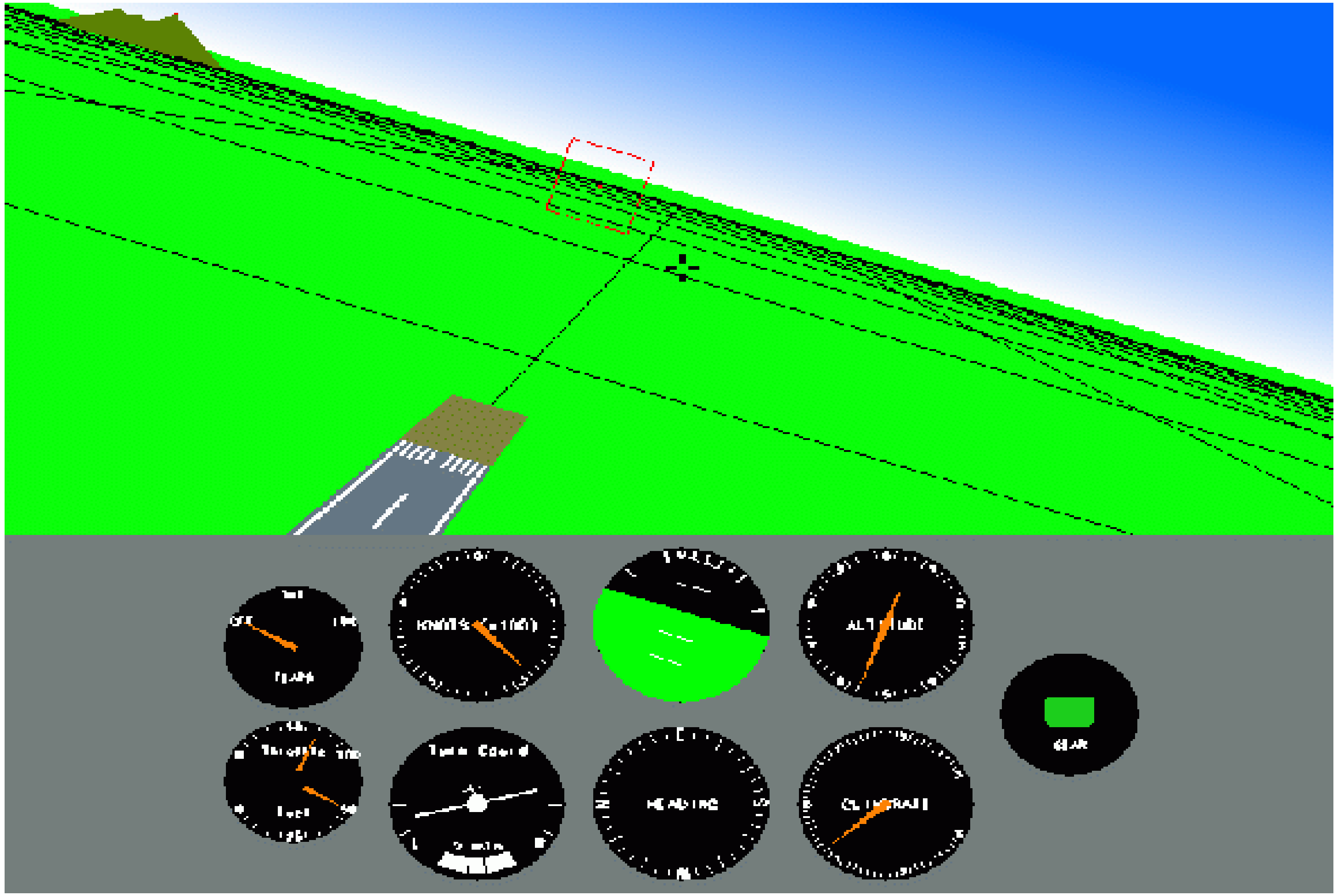
Reinforcement Learning

States (S), actions (A), immediate rewards (R), discounted infinite horizon, …

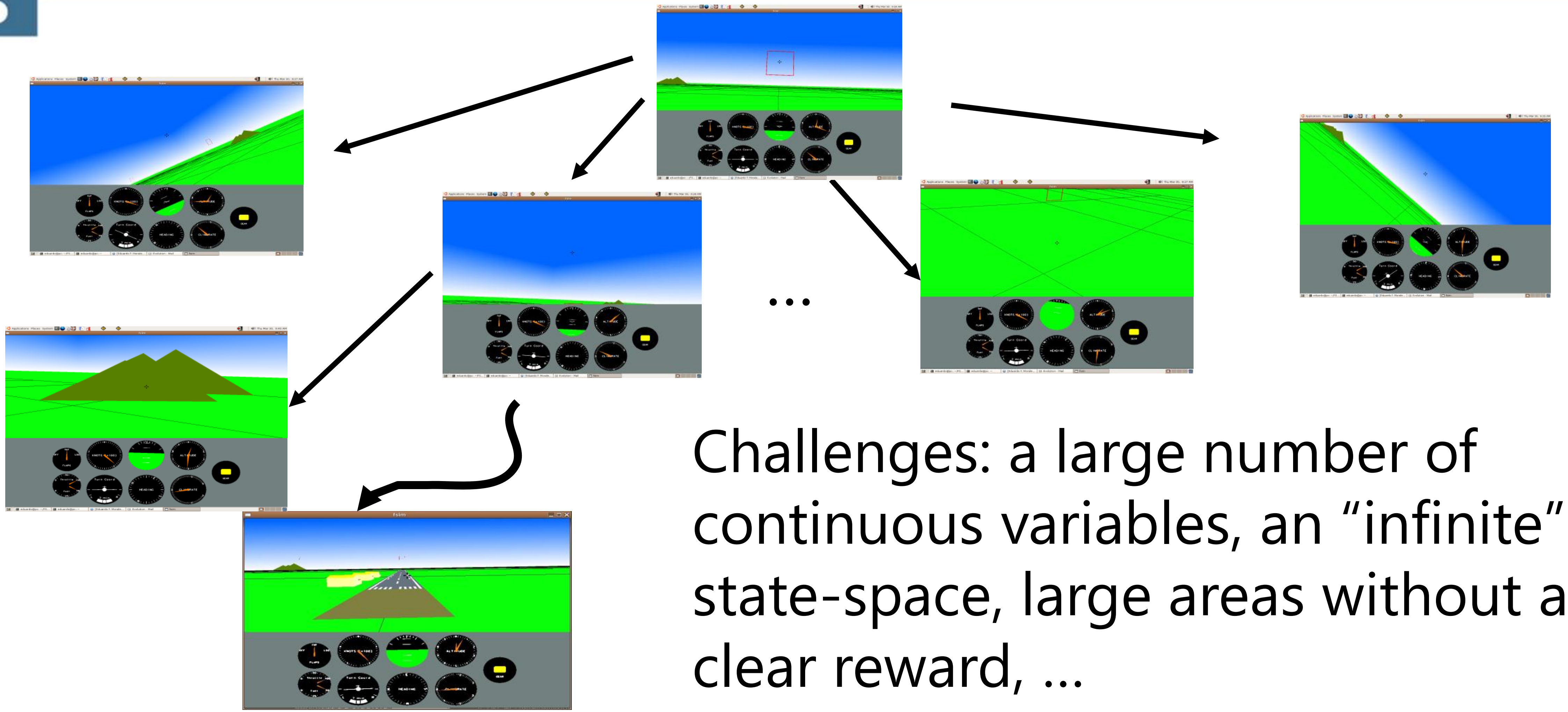
Learn from interactions with the environment how to map states to actions to maximize the total expected accumulated reward



Can We Use It To Learn How To Fly?



Learning To Fly With RL



Challenges: a large number of continuous variables, an "infinite" state-space, large areas without a clear reward, …

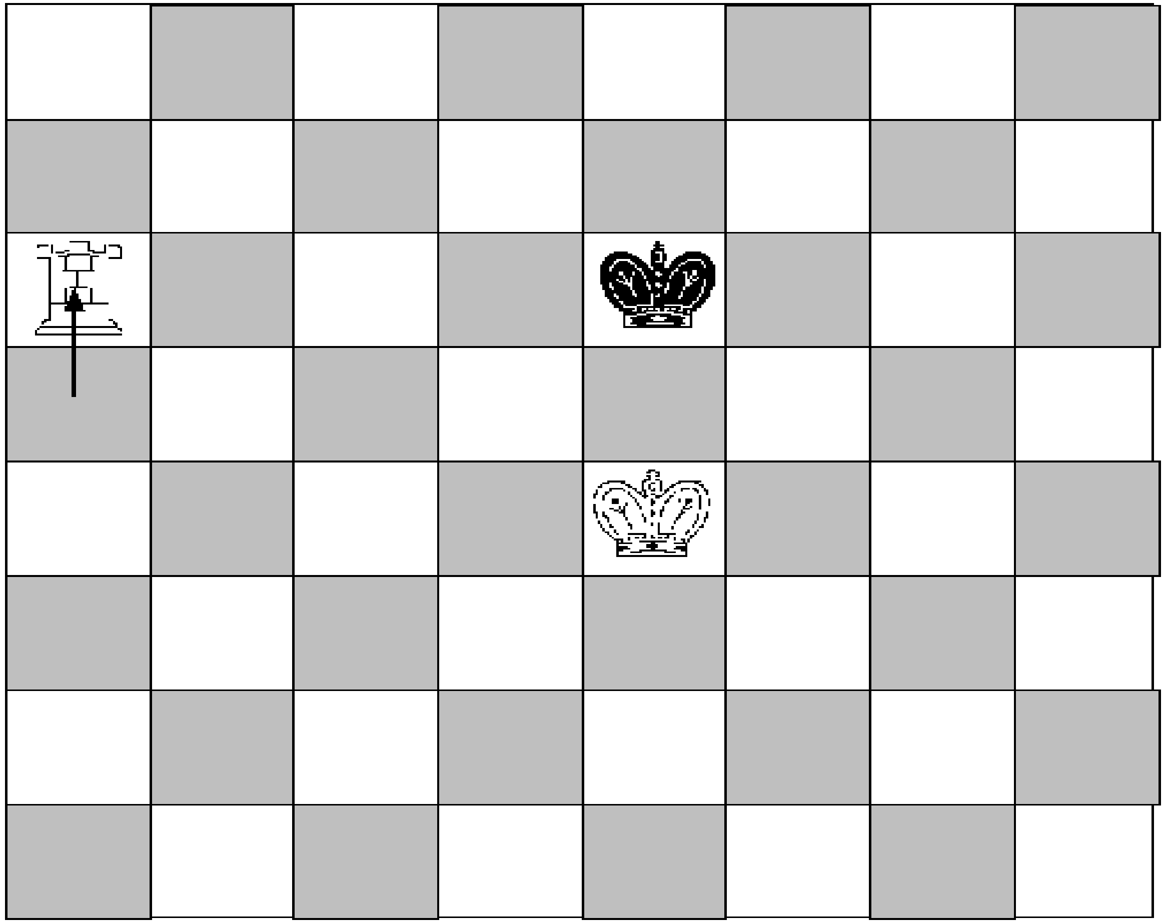
Two Main Ideas

Use a relational representation:

- Easy to express powerful abstractions
- Can incorporate background knowledge
- The learned policies can be re-used in similar problems

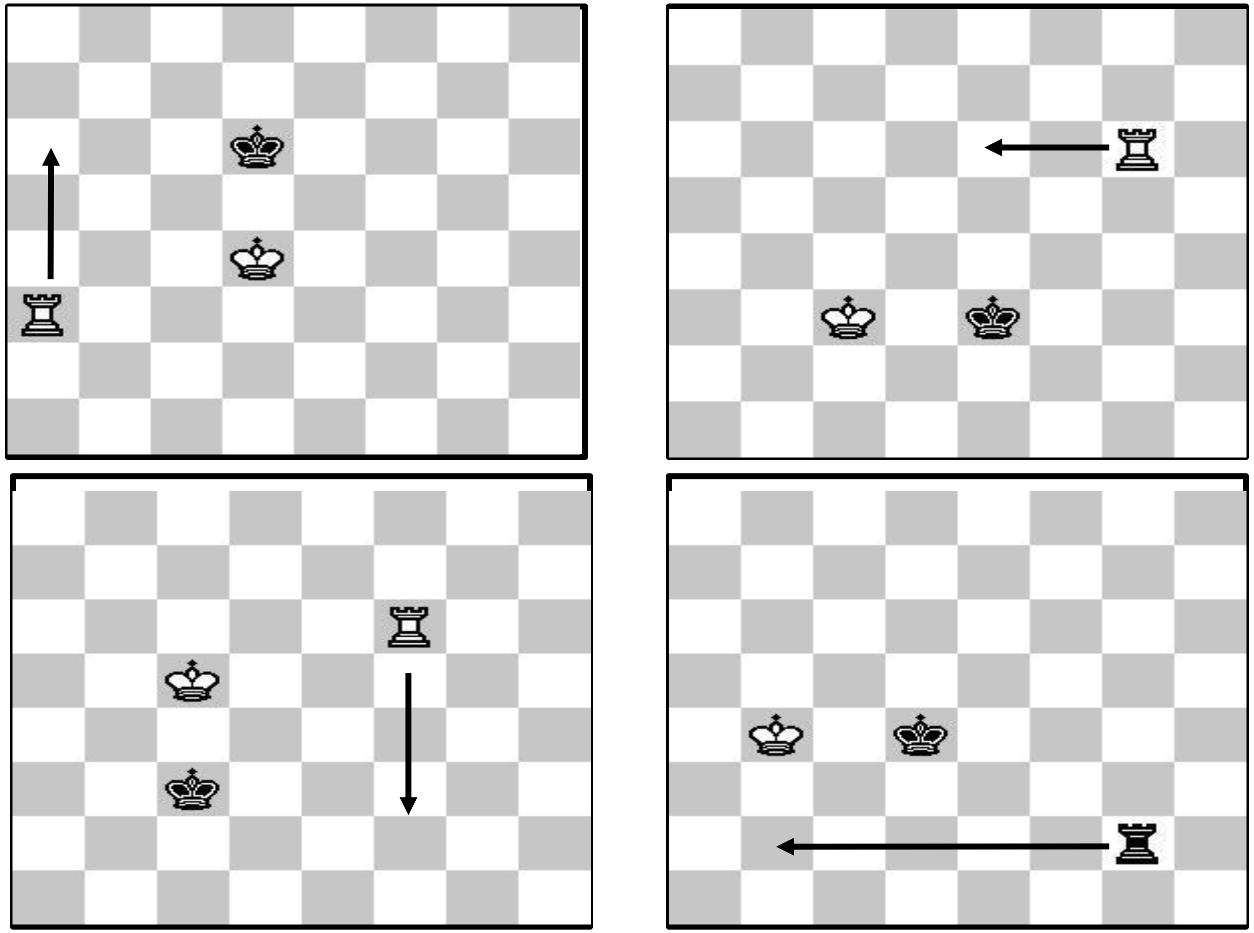
Learn/consider a subset of relevant actions from user-provided traces

Relational Representation



> 150,000 (positions) states & up to 22 actions per state

Equivalent State-Action Pairs



STATE:

kings_in_oppos(*S*) and

not threatened(*S*) *and* …

ACTION:

IF *kings_in_oppos*(*S1*) and

not threatened(*S1*) and …

THEN *move*(rook,*S1*,*S2*)

Induce Actions from Traces

Learn a subset of relevant actions per state from human traces

For each frame of a trace-log:

- Transform the information of the frame into a relational representation (rS)
- Construct, if new, an action with the conjunction of the predicates (rS) and a predicate-action (rA)

rQ-Learning Algorithm

Initialize $Q(s_r, a_r)$ arbitrarily

repeat (for each episode)

Initialize $s, s_r \leftarrow$ rels(s)

repeat (for each step in episode)

Choose a_r from s_r using policy derived from Q

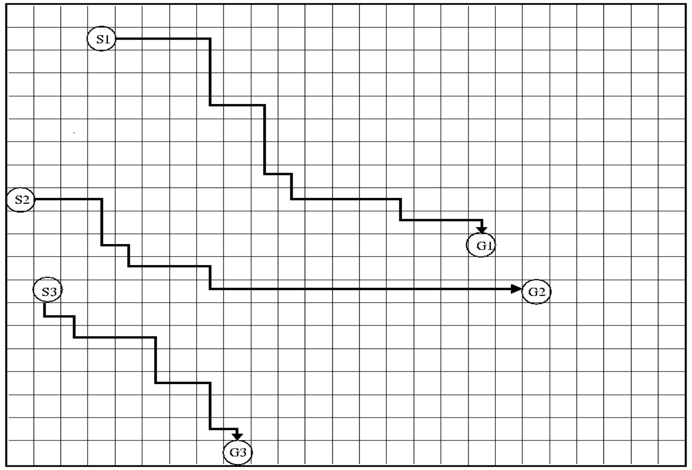
Randomly take a from a_r; observe r, s', $s_r' \leftarrow$ rels(s')

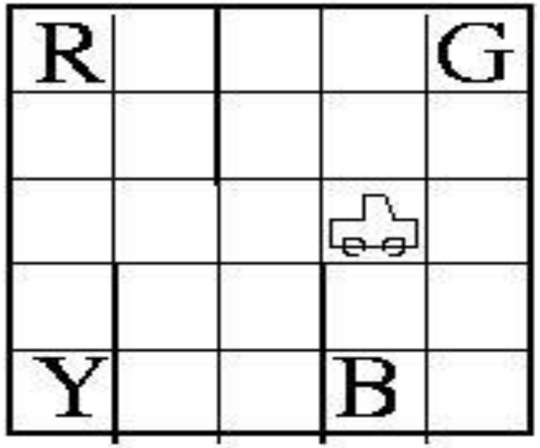
$Q(s_r, a_r) \leftarrow Q(s_r, a_r) + \alpha[r + \gamma \max_{ar'} Q(s_r', a_r') - Q(s_r, a_r)]$

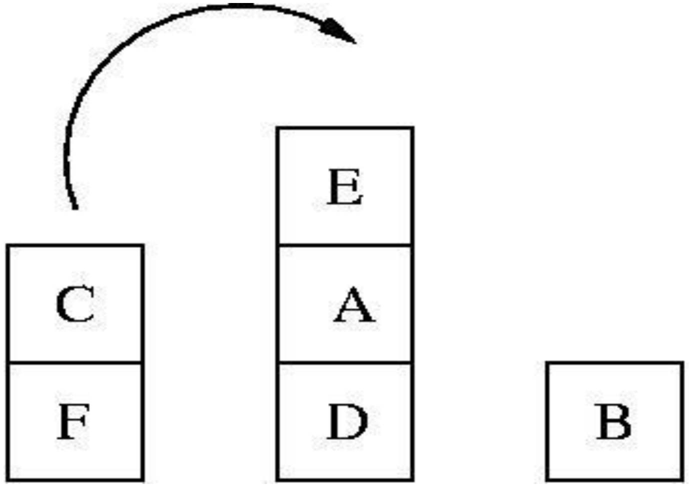
$s \leftarrow s'$, $s_r \leftarrow s_r'$

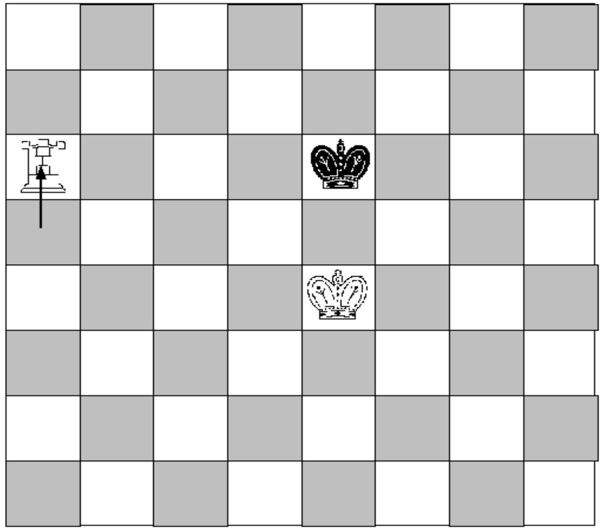
Until s is terminal

Previous Experiments

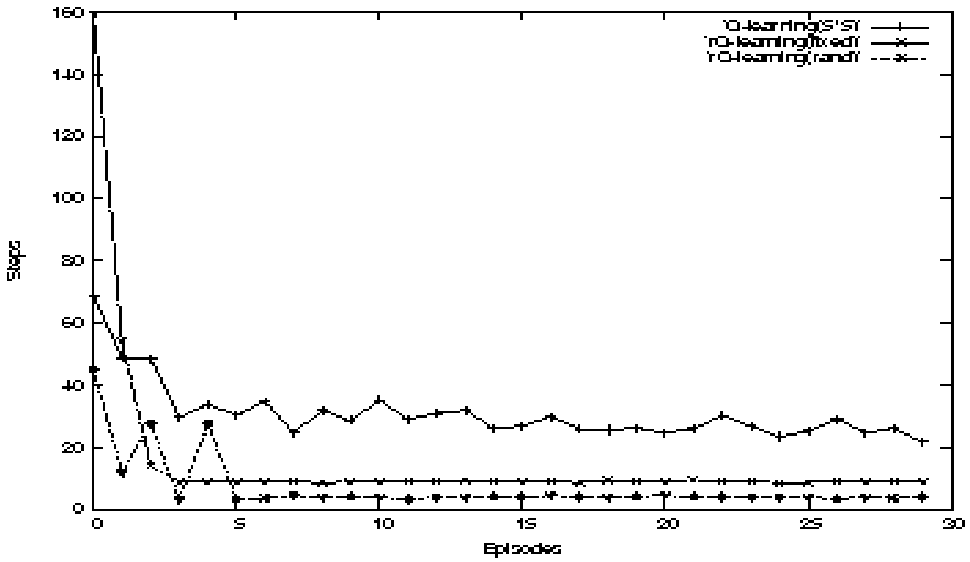




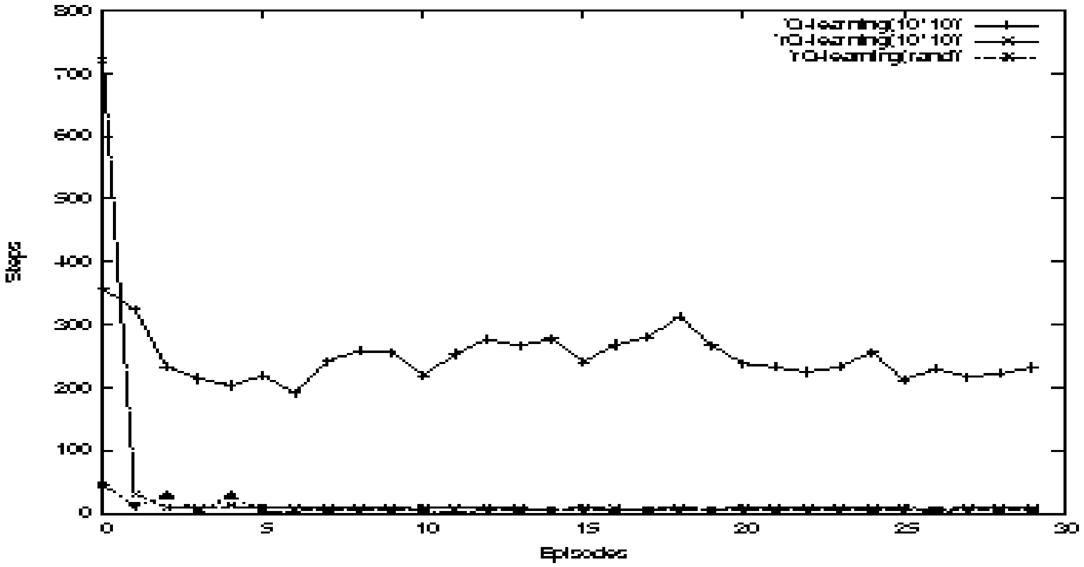




Faster Convergence

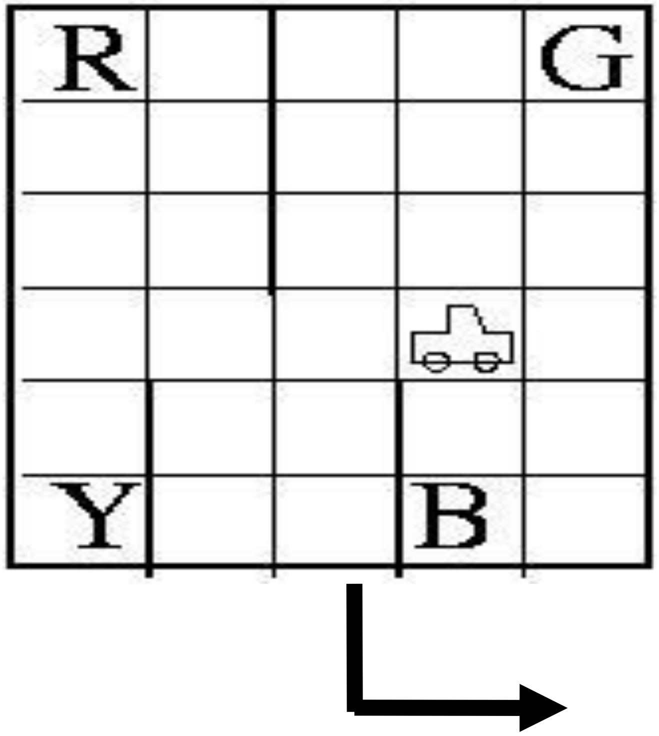


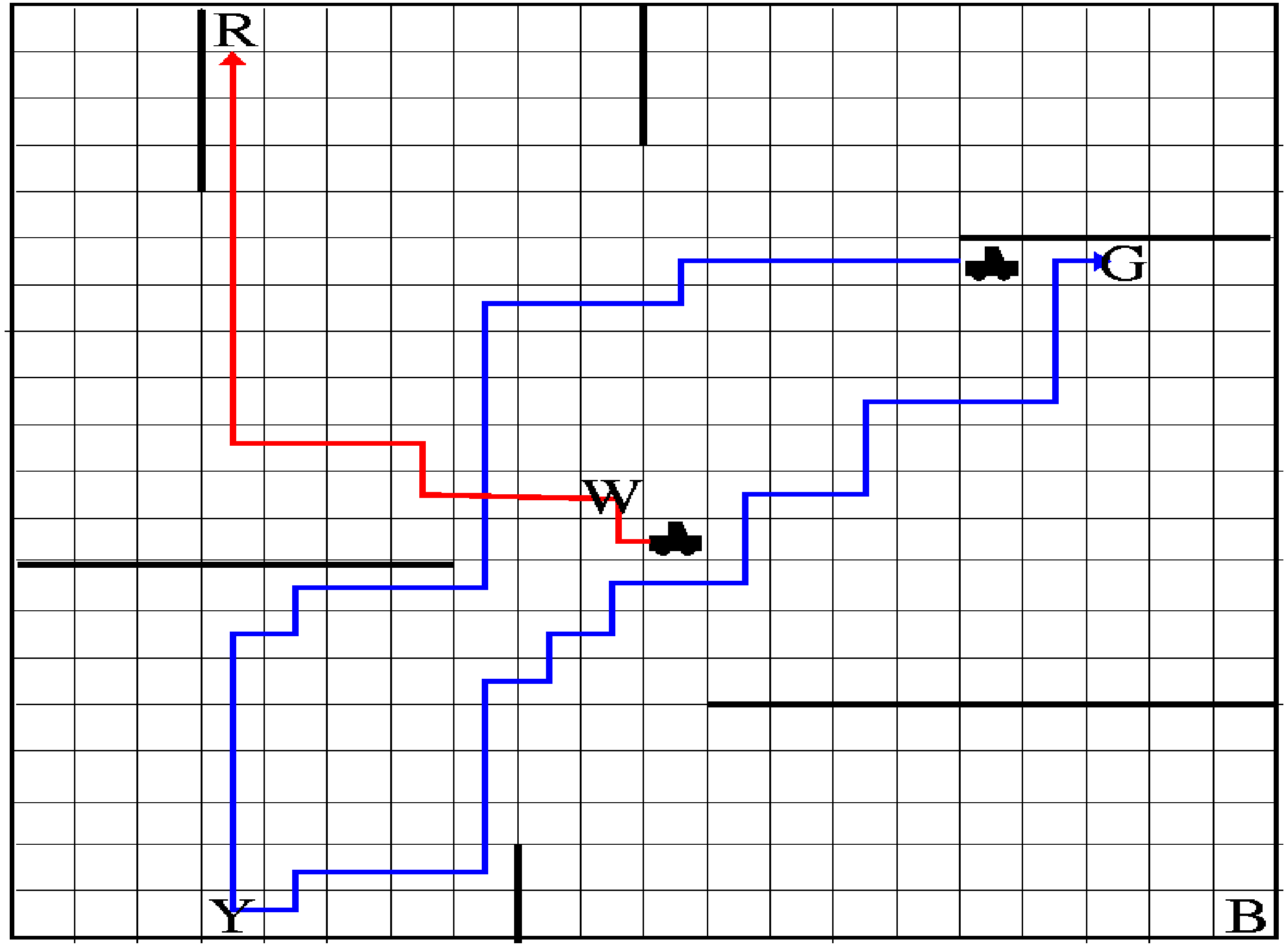
5 x 5 grid



10 x 10 grid

Re-Usability Of Policies





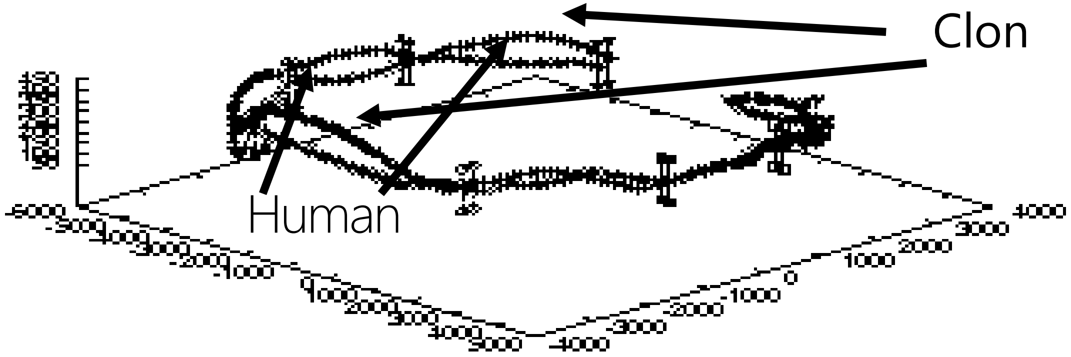
Learning To Fly

Assume the aircraft is in the air, with constant throttle, flat flaps and retracted gear

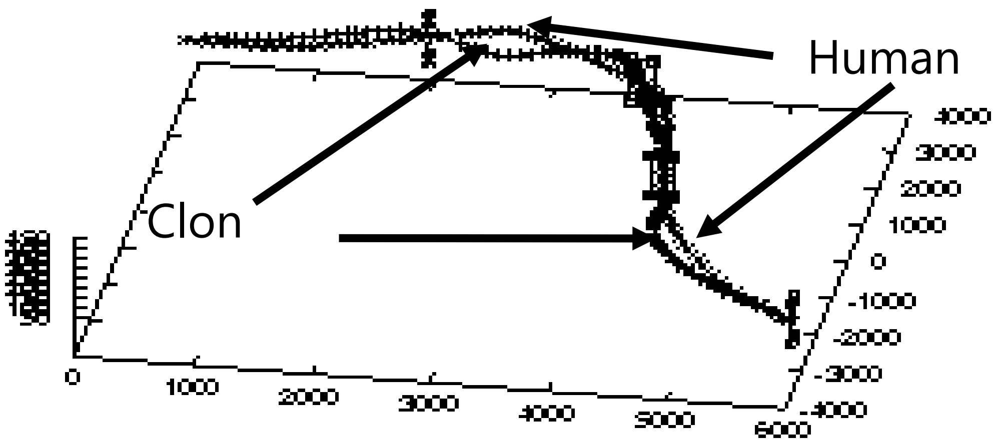
Two stages:

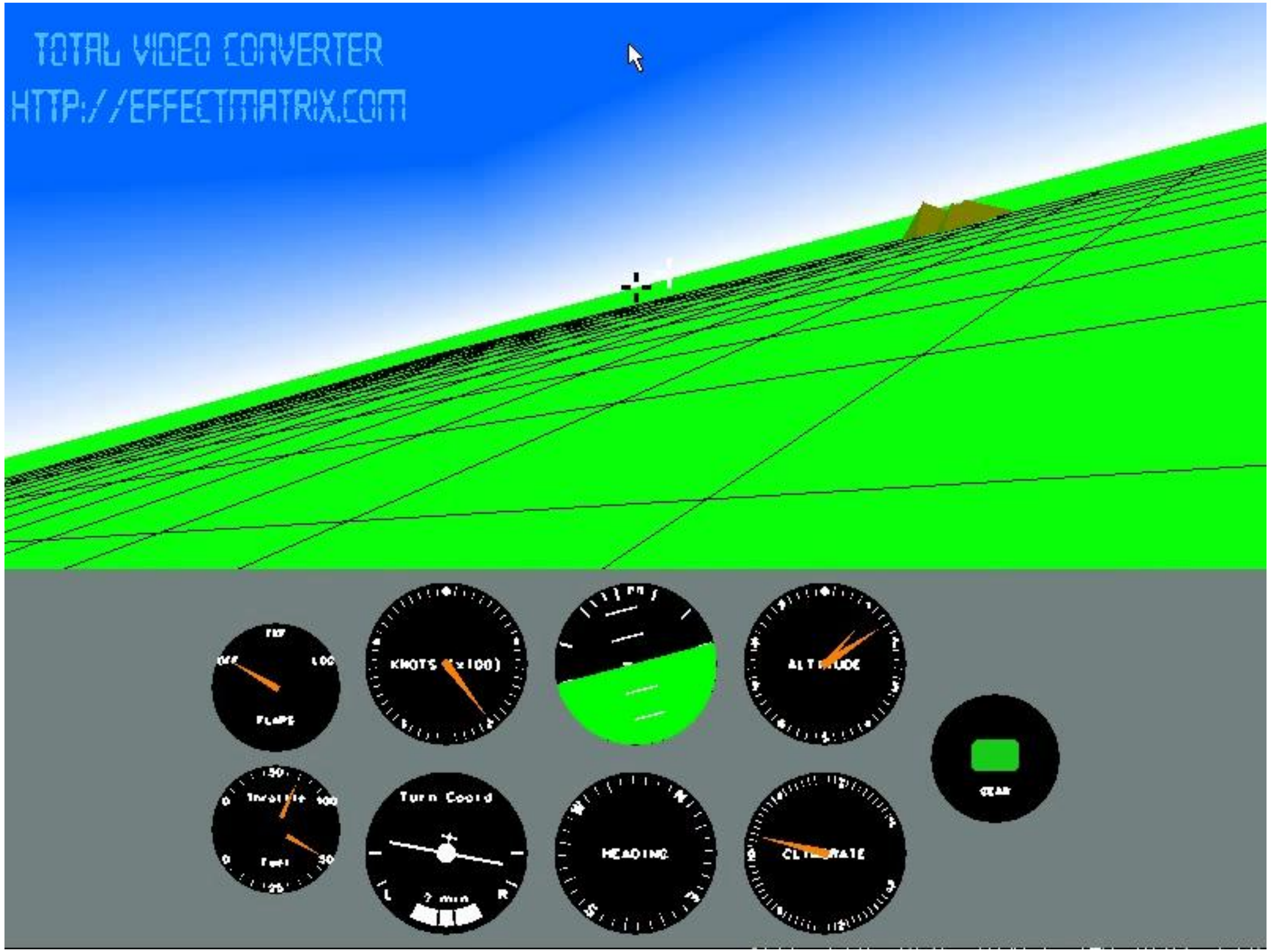
- Induce actions from traces of flights (5)
- Use the learned actions to explore and learn new actions until (almost) no more learning (20 trials)
- 32% (359) aileron 1.6 (of 5) per state
- 64% (180) elevation 3.2 (of 5) per state

Results With High Turbulence



Results On Different Flight Plans



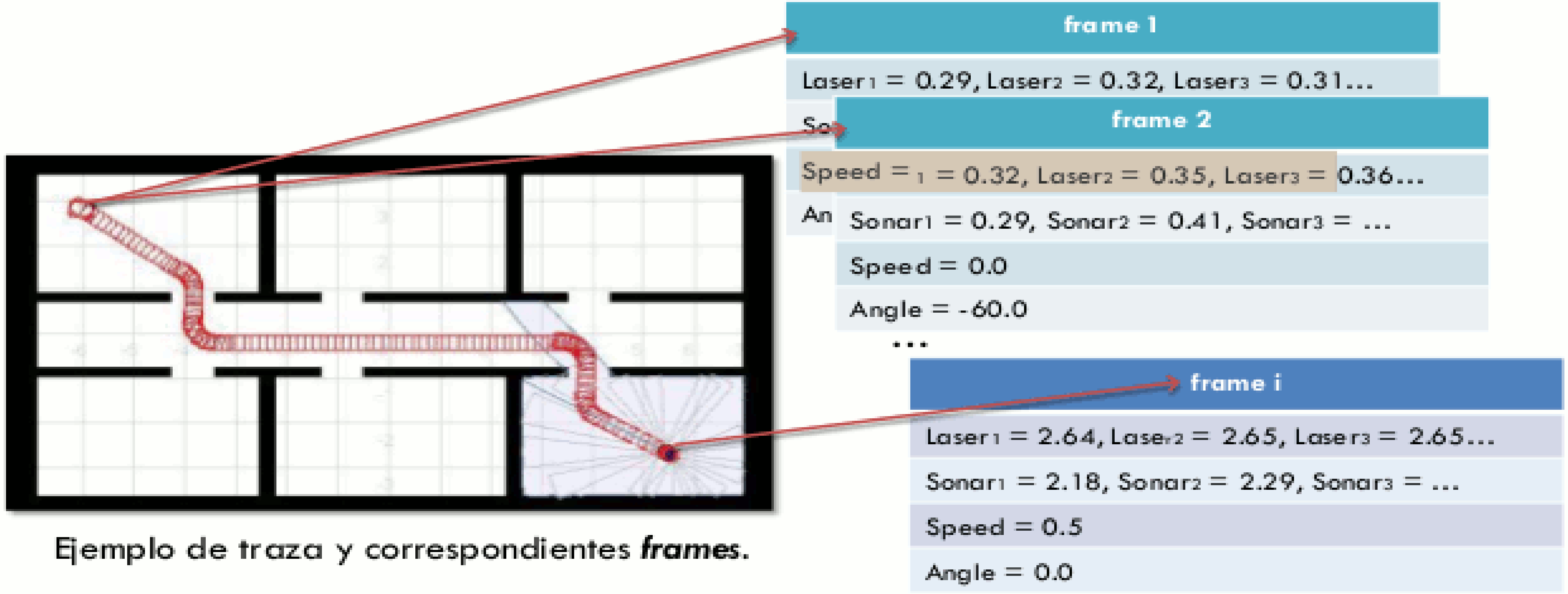


(1) The User Controls (Joystick/Keyword)

Steps:

- The user provides traces
  - *Transform the low-level sensor information into a relational representation*
- Learn a policy
  - *Transform on-line the discrete-actions policy into a continuous-actions policy*

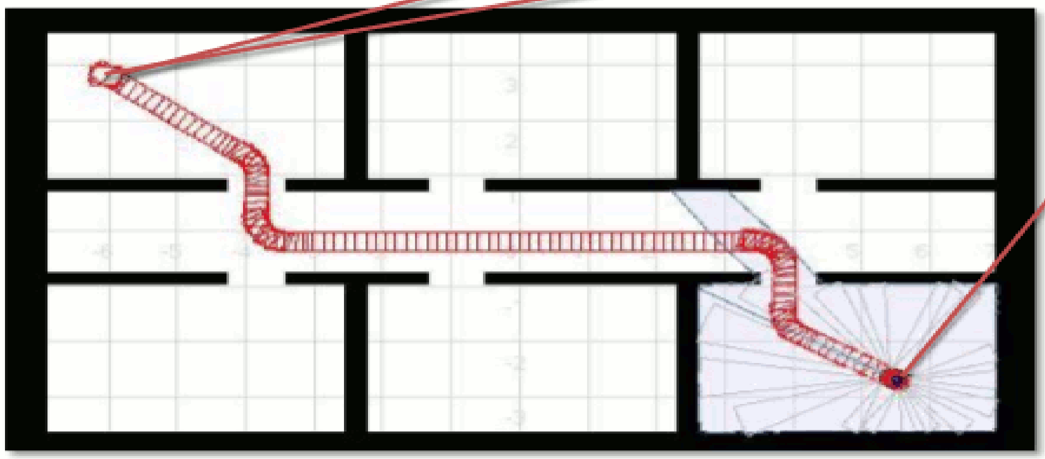
Original Traces



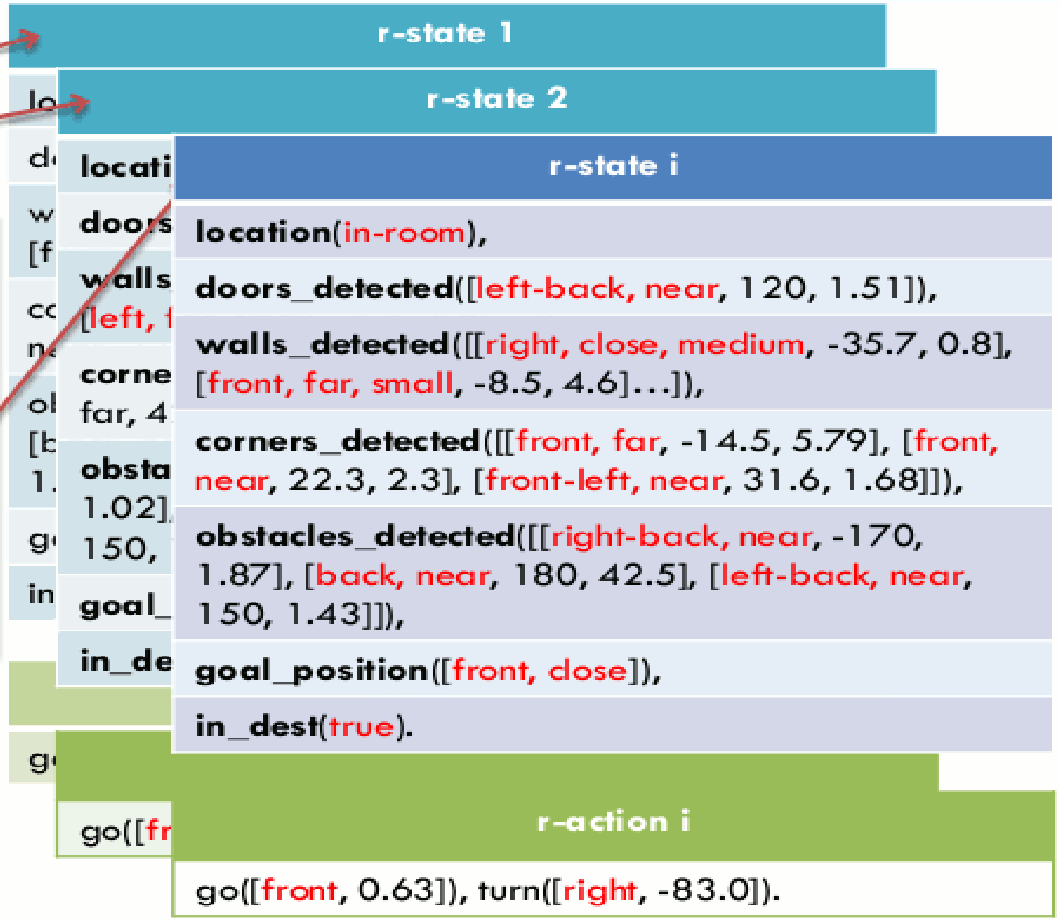
Ejemplo de traza y correspondientes *frames*.

Transformed Traces

- Los *frames* se convierten en pares r-estado-r-acción.

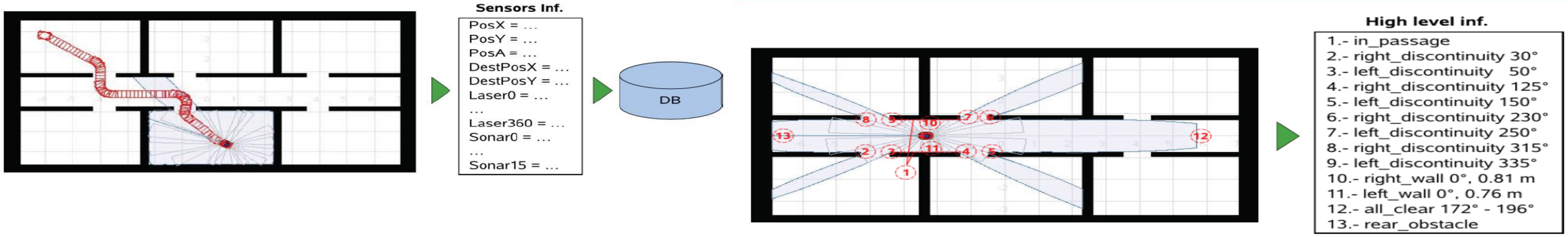


Ejemplo de traza y correspondientes pares r-estado-r-acción.

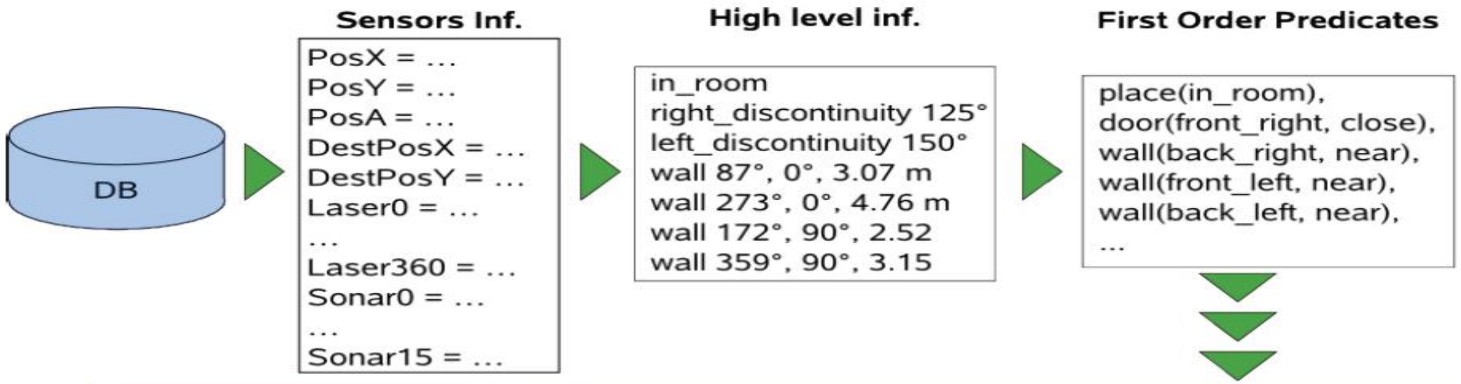


Learn a policy with this representation (as in the flight simulator)

Learning From Traces …



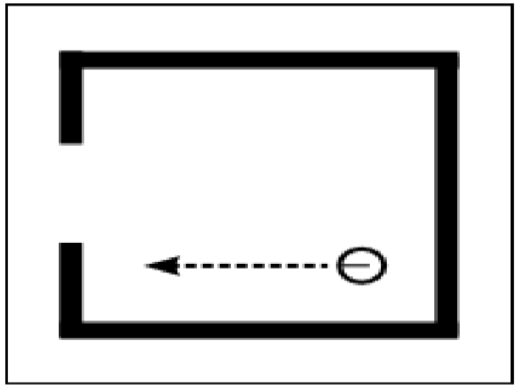
Transform sensor's information into a more "natural" and transferible representation



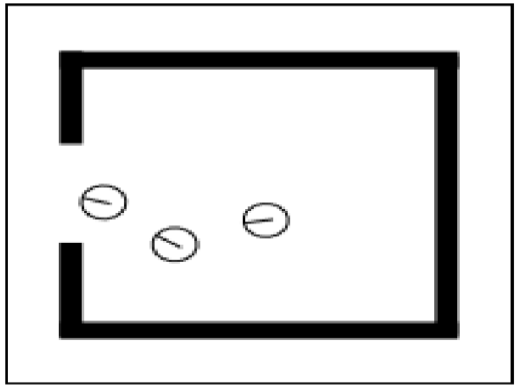
state	place(State,in_room), doors_detected(State, front_right, close), walls_detected(State, back_right, close, front_left, near, back_left, near), passages_detected(State, nil) goal_orientation(front_right), goal_distance(far), rear_obs(true), goal_reached(false),
r-action	turn_right

On-line Transformation To A Continuous-Actions Policy

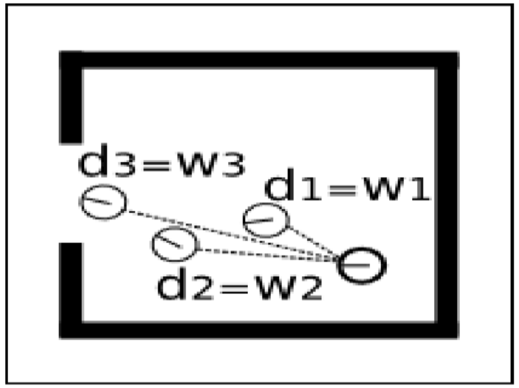
Discrete-actions policy



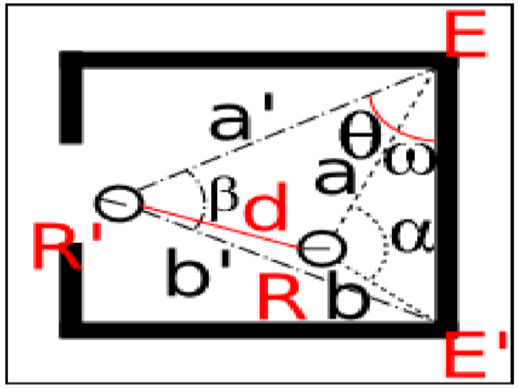
St-Ac pairs from traces



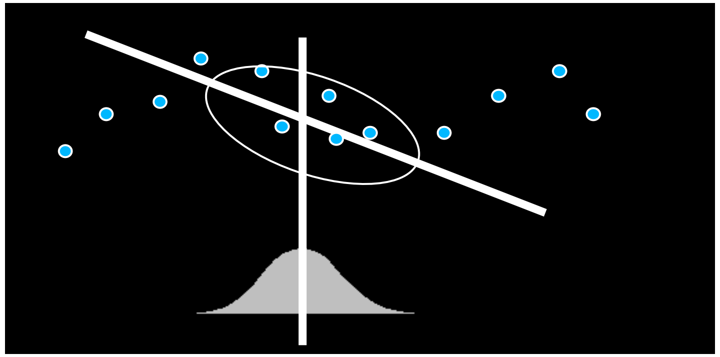
Relate similar States-Actions



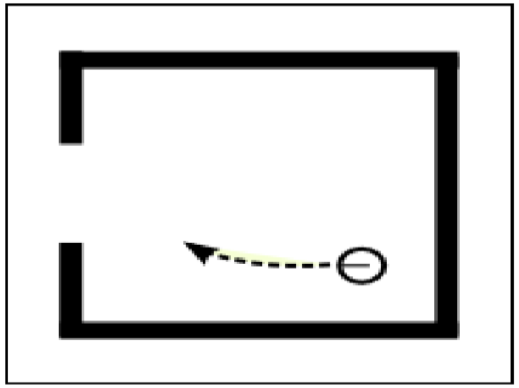
Consider natural landmarks



Weighted with a Gaussian (LWR)

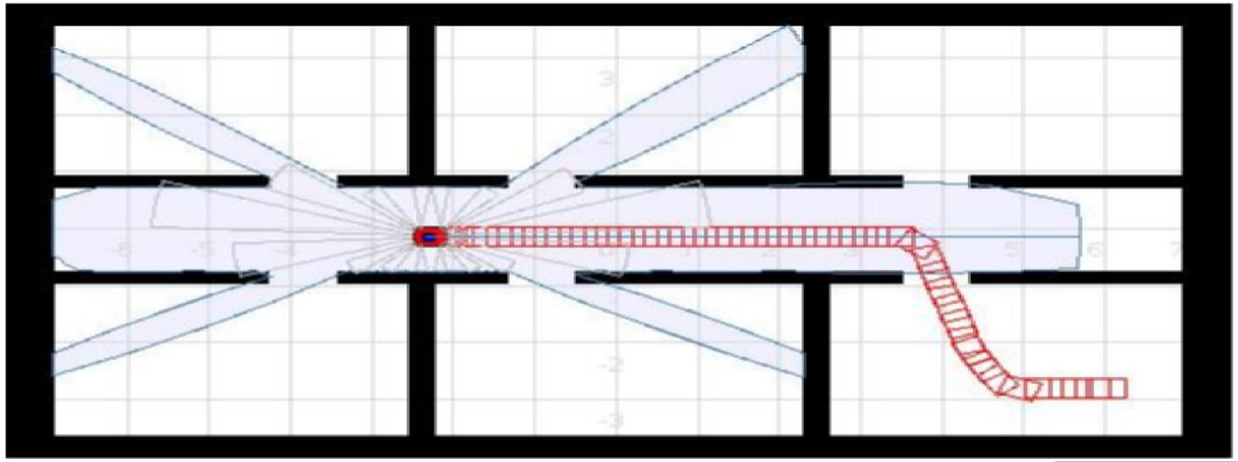


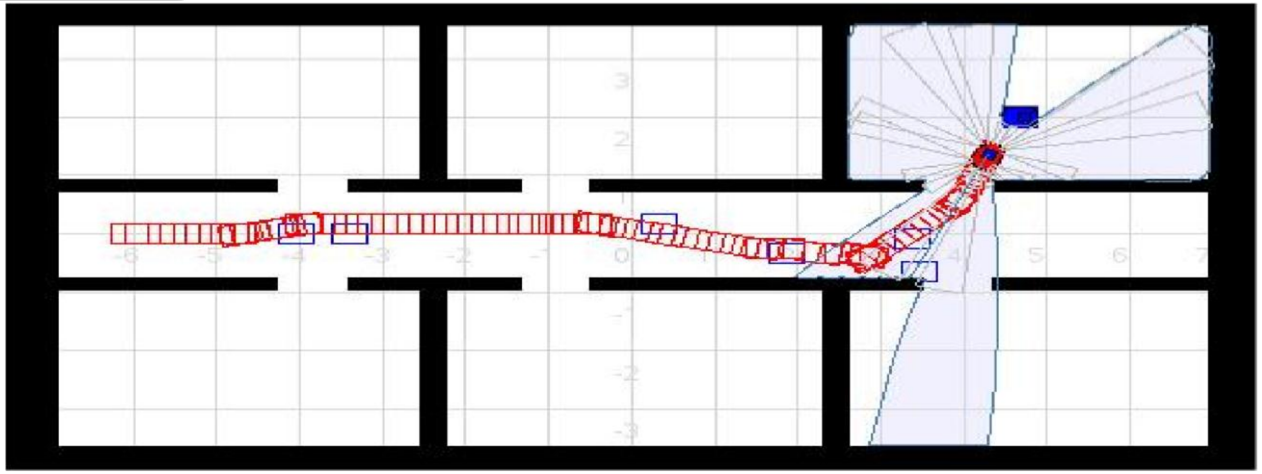
Combined weighted action



Experiments (Training)

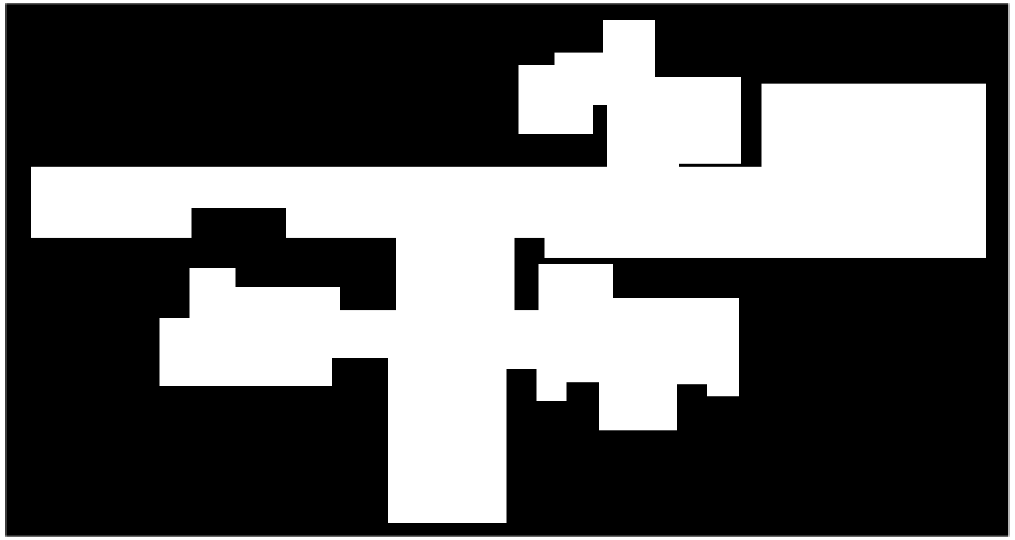
20 *navigation* traces and 20 *following* traces

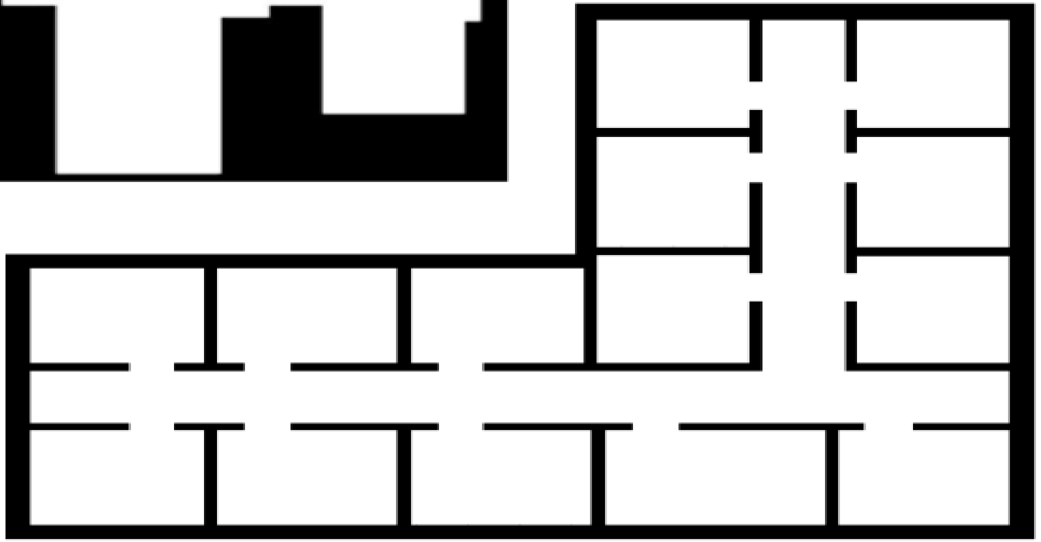




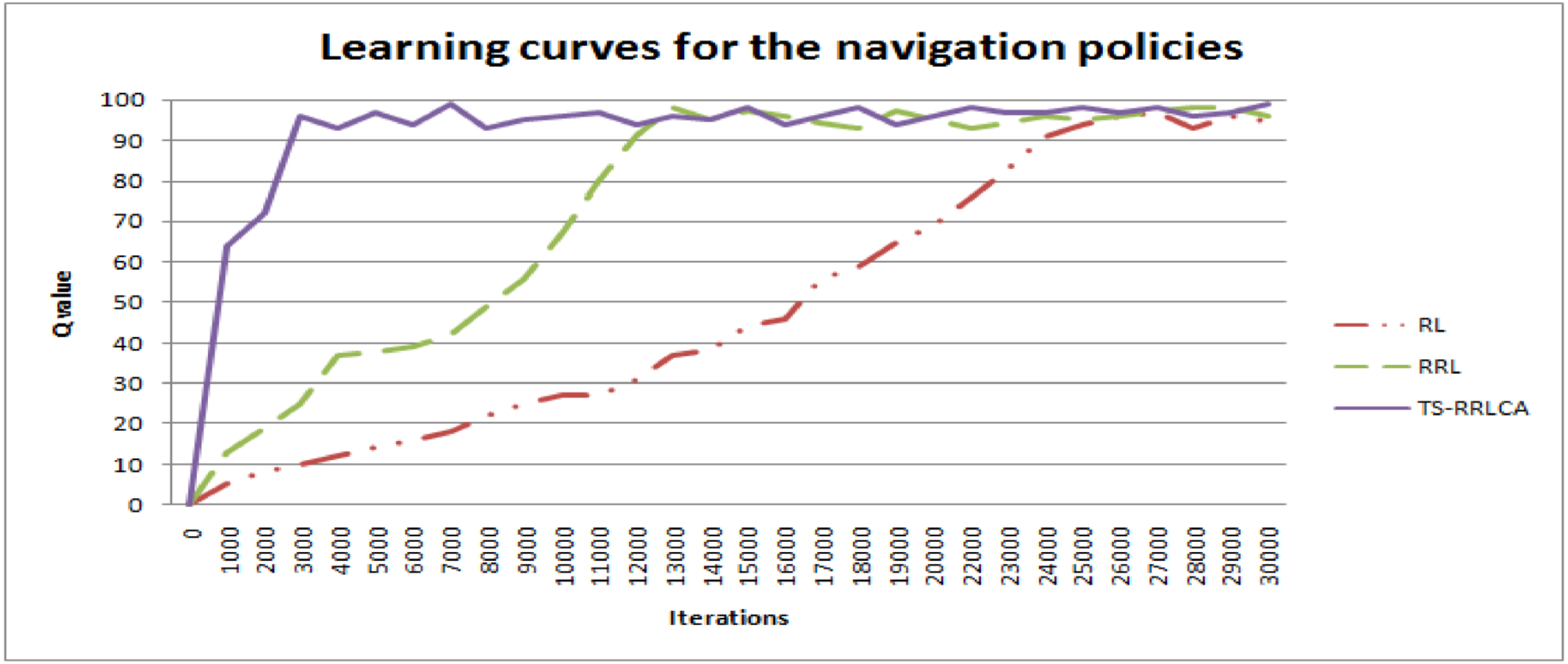
Experiments (Testing)

10 *navigation* and 10 *following* tasks with different maps and goals

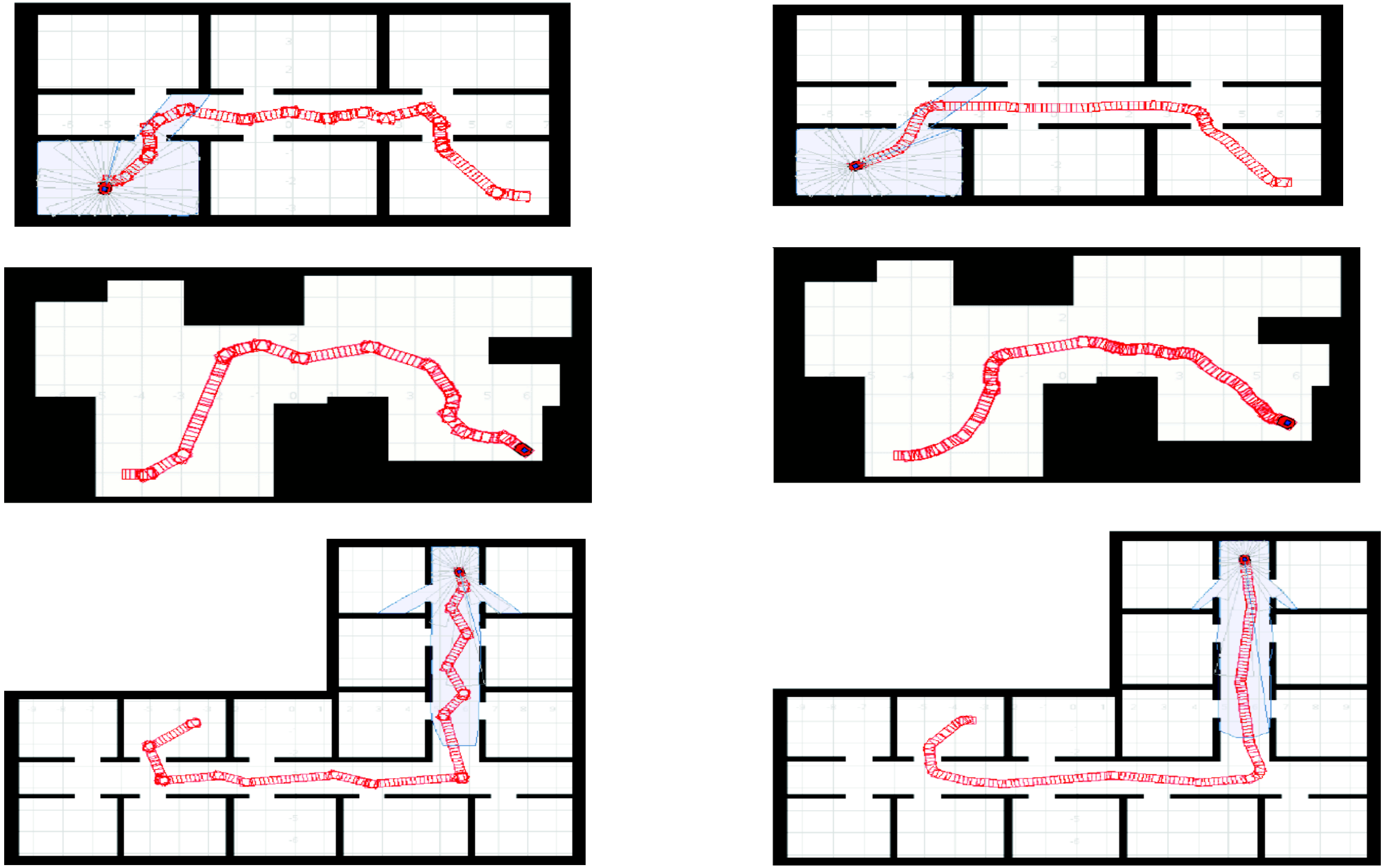




Learning Curves

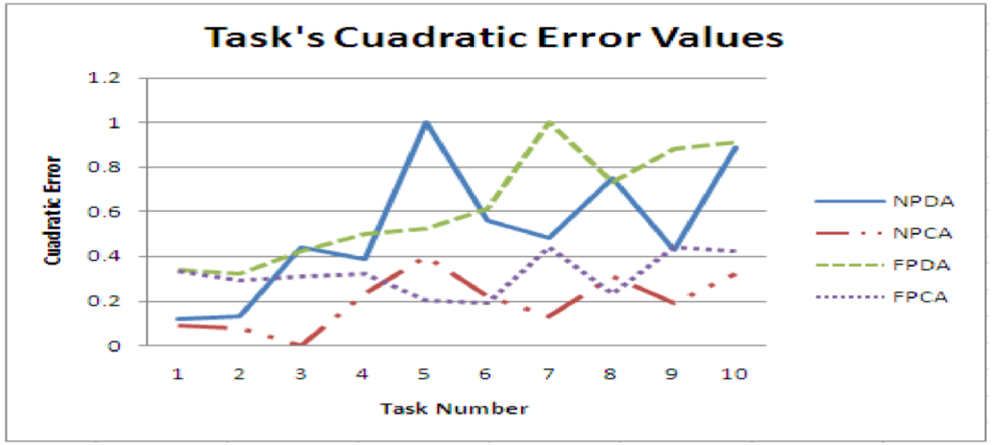


Discrete vs. Continous Policies

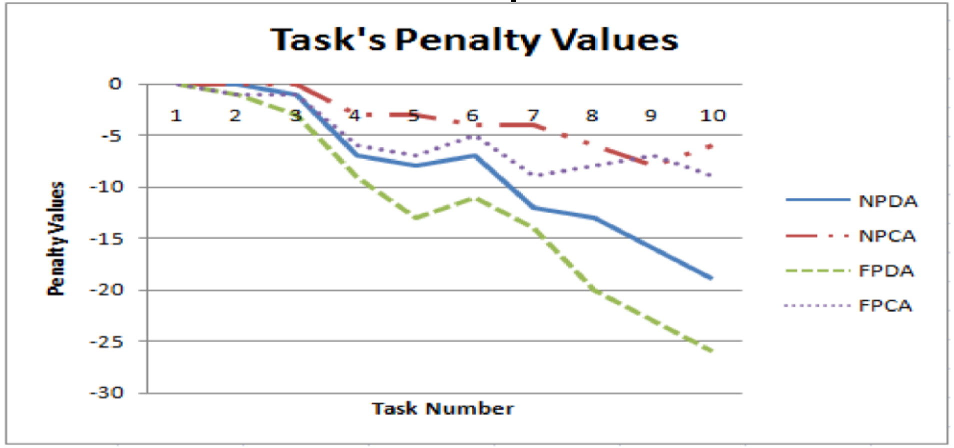


Discrete vs. Continous Policies

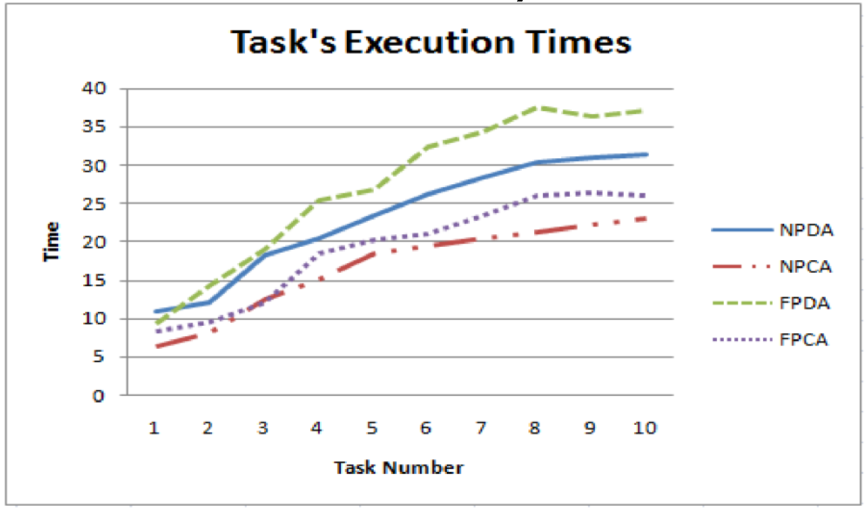
Similar to human traces



Safer paths



Faster trajectories



Discrete Navig.
Continuous Navig.
Discrete Follow.
Continuous Follow.

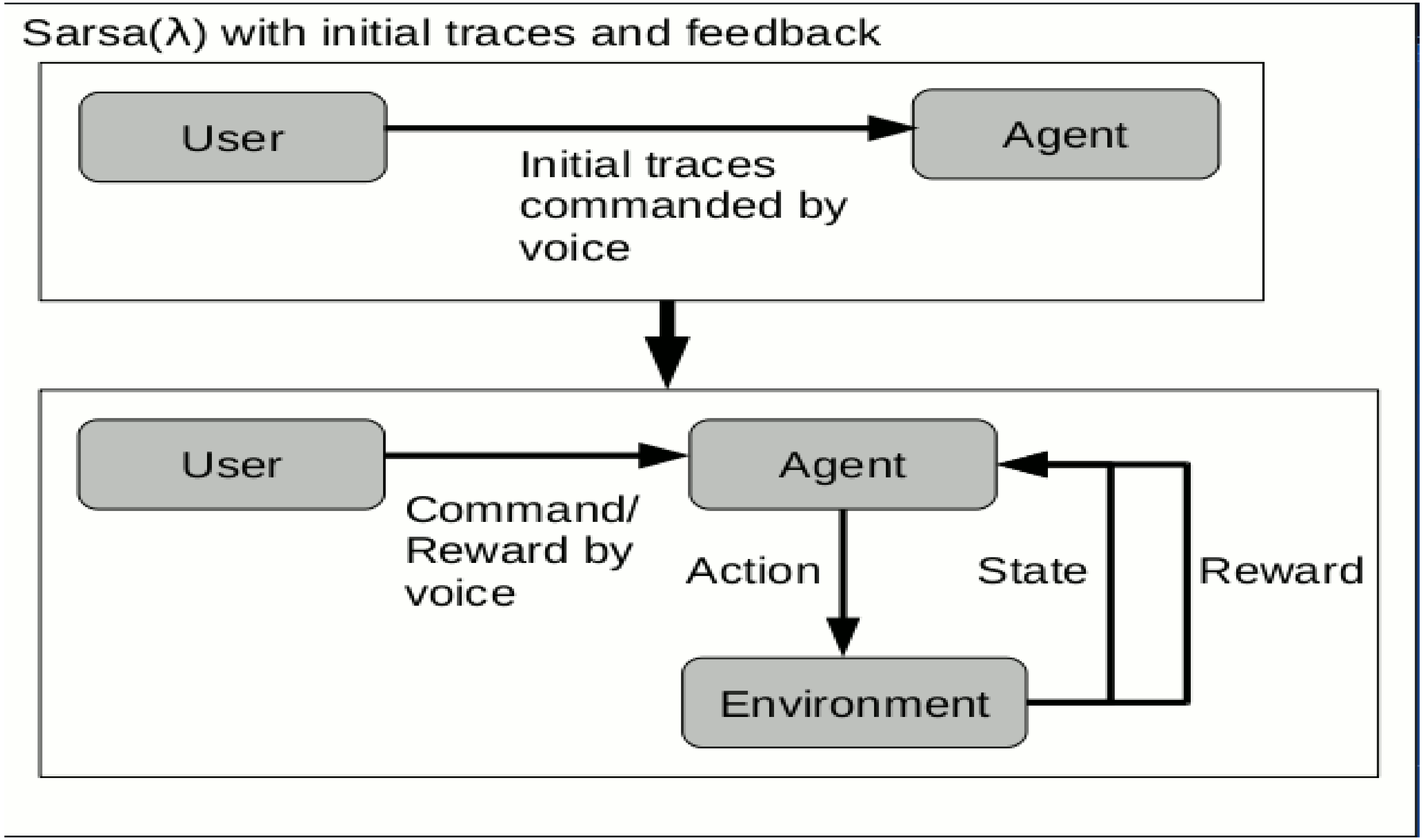
(2) The User Instructs (Voice)

Generate traces with voice commands

New issues:

- Errors in the speech-recognition system
- Try current policy and provide voice feedback during the learning process (a.k.o. *dynamic/on-line reward shaping*)

The User Instructs (Voice)



Dynamic Reward Shaping

Feedback can directly change temporarily the reward ($R = R_{rl} + R_u$) and the actions suggested by the policy

Other issues:

- Delayed feedback
- Inconsistent feedback over time

Dynamic Reward Shaping

Some feedback cases:

- Continuous (can change policy and create new sub-goals)
- Sporadic (how can it affect the result?)
- Noisy (how robust is the strategy to noise?)

Vocabulary

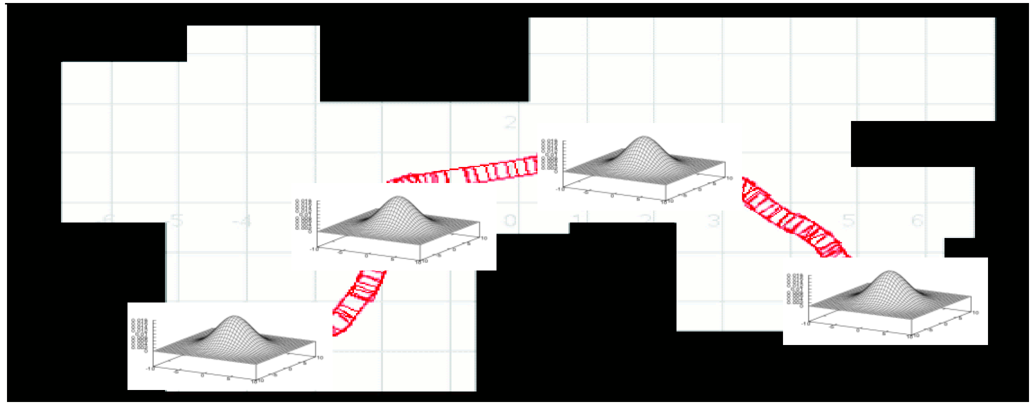
Izquierda	Derecha
Gira a la izquierda / derecha	Es para el otro lado
Gira hacia la izquierda / derecha	Avanza
Hacia tu izquierda / derecha	Adelante
A tu izquierda / derecha	Sigue avanzando
A la izquierda / derecha	Sigue caminando
Hacia la izquierda / derecha	Sigue derecho
Da vuelta a la izquierda / derecha	Camina derecho
Ve a la izquierda / derecha	Ve todo derecho
A mano izquierda / derecha	Ve derecho
Vuelta a la izquierda / derecha	Ahí derecho
Ve hacia la izquierda / derecha	Todo derecho
Dobla a la izquierda / derecha	Síguele
Dobla hacia la izquierda / derecha	Atrás
Hacia atrás	Hacia adelante
Para atrás	Para adelante
Hey regresa	Mejor regresa
Regresa	

Hasta ahí	Ya hasta ahí
Hasta ahí nada mas	Para ahí
Excelente	Bien
Así como vas	Como vas
Tu síguele	Así sigue
Sigue así	Sí así
Muy bien	Bien hecho
Buen trabajo	Vas bien
Mal	Terrible
Así no	Muy mal
Estás mal	Hey para allá no
Hacia allá no	Para allá no
Allá no	Ya no hagas eso
Que no	Que eso no
Que ahí no	Que así no
Por ahí no	Ya te equivocaste
Ya la regaste	No te vayas por ahí
No era por ahí	Por ahí no era

We used Sphinx3 and Dimex (UNAM) ≈ 250 words

States and Actions

States are incrementally generated from the traces

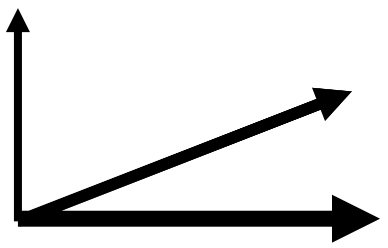


Highly correlated states (Pearson) are considered equal

$$r = \frac{N\Sigma xy - \Sigma x \Sigma y}{(\sqrt{N\Sigma x^2 - (\Sigma x)^2})(\sqrt{N\Sigma y^2 - (\Sigma y)^2})}$$

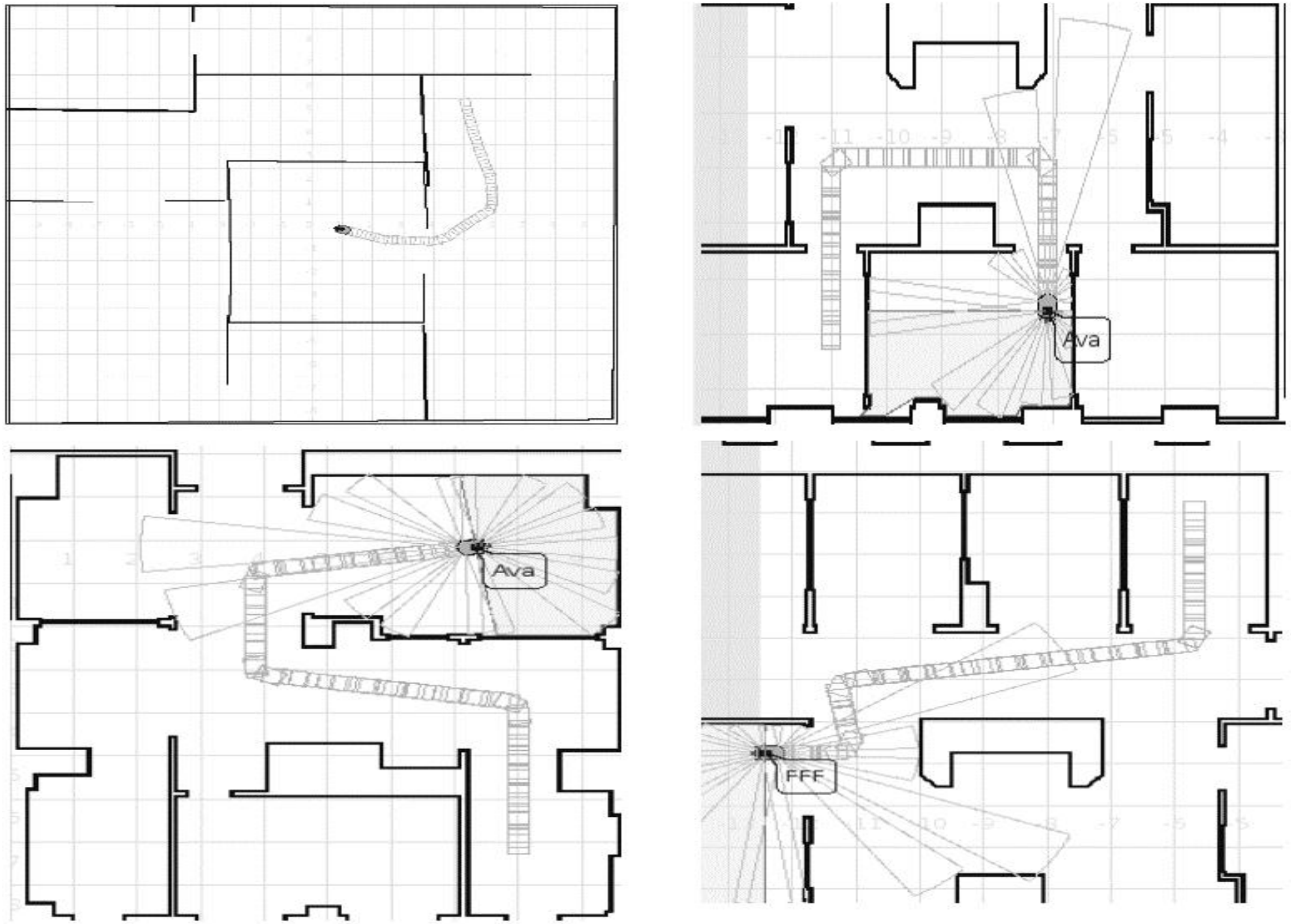
Continuous Actions

We used Sarsa(λ) with discrete actions however the resulting action is a combination of the dominated actions

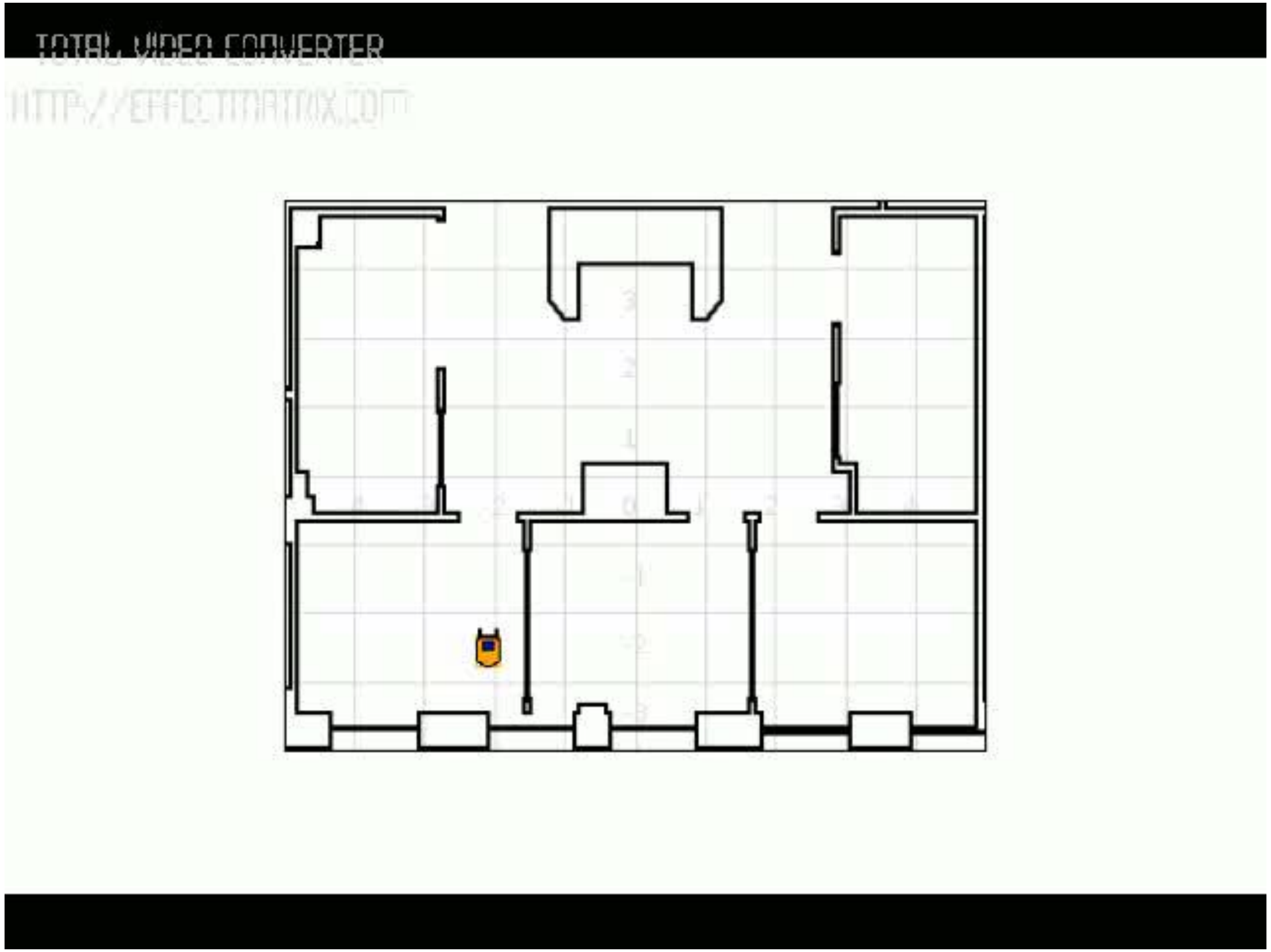


Update Q-values proportionally to the Q values of the used actions

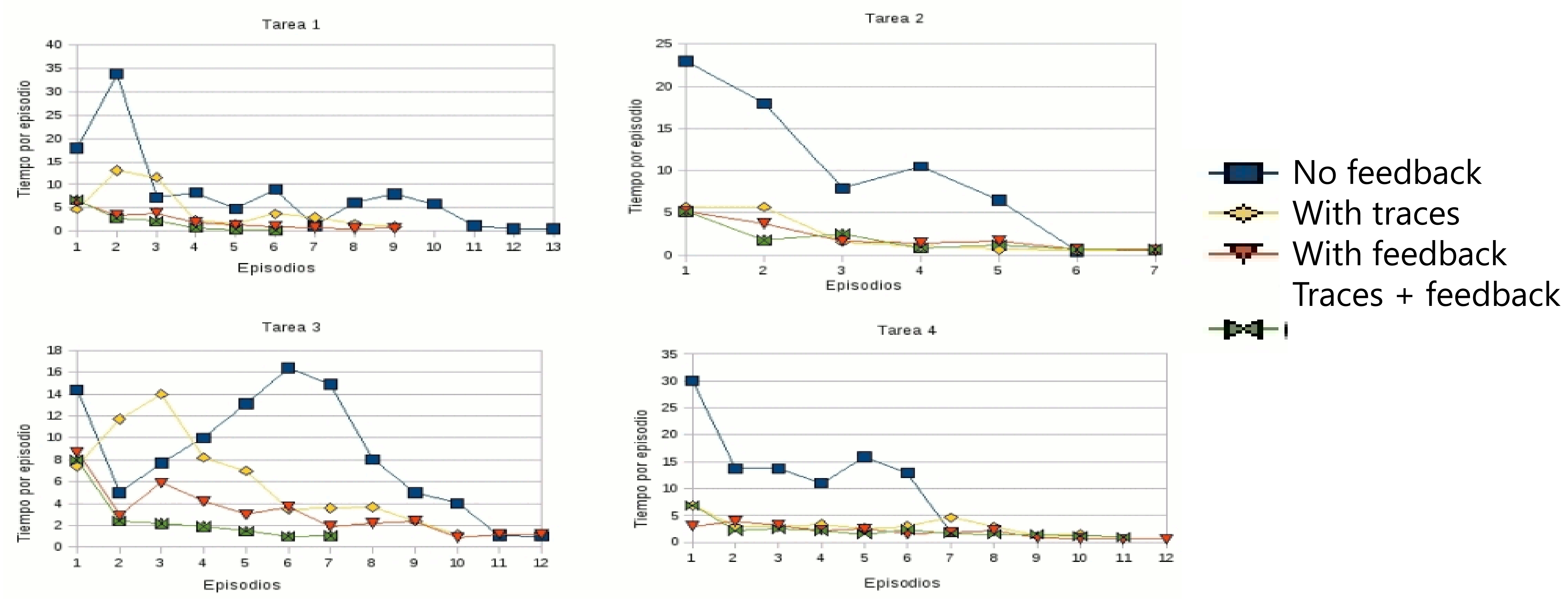
Navigation Tasks



Example Of Initial Trial



Results



Tarea 1
Tarea 2
Tarea 3
Tarea 4
Tiempo por episodio
Episodios
No feedback
With traces
With feedback
Traces + feedback

Results

Number of Episodes

Task	RL	RL+T	RL+ F	RL+T+F
T1	13	9	9	6
T2	6	7	7	7
T3	12	10	12	7
T4	7	10	12	11
Aver.	9.5	9	10	7.75

Time (min)

Task	RL	RL+T	RL+ F	RL+T+F
T1	103.93	41.8	19.59	12.856
T2	66.4	16.07	15.1	13
T3	100.65	62.66	38.2	18.09
T4	99.1	31.9	23.43	24.61
Aver.	92.54	38.11	24.08	17.13

Results

	Time	Interventions
T2 “nomal feedback”	13	34
T2 Perfect feedback	7.6	39
T2 No feedback	20.51	N/A
T2 50% errors in feedback	66.4	187

(3) The User Shows (Vision)

Generate traces by showing how to do it

Transform them to possible robot traces

Learn and adjust with exploration (RL) and on-line feedback (voice)

New Issues:

- Estimate 3D positions from cameras
- Generate corresponding traces

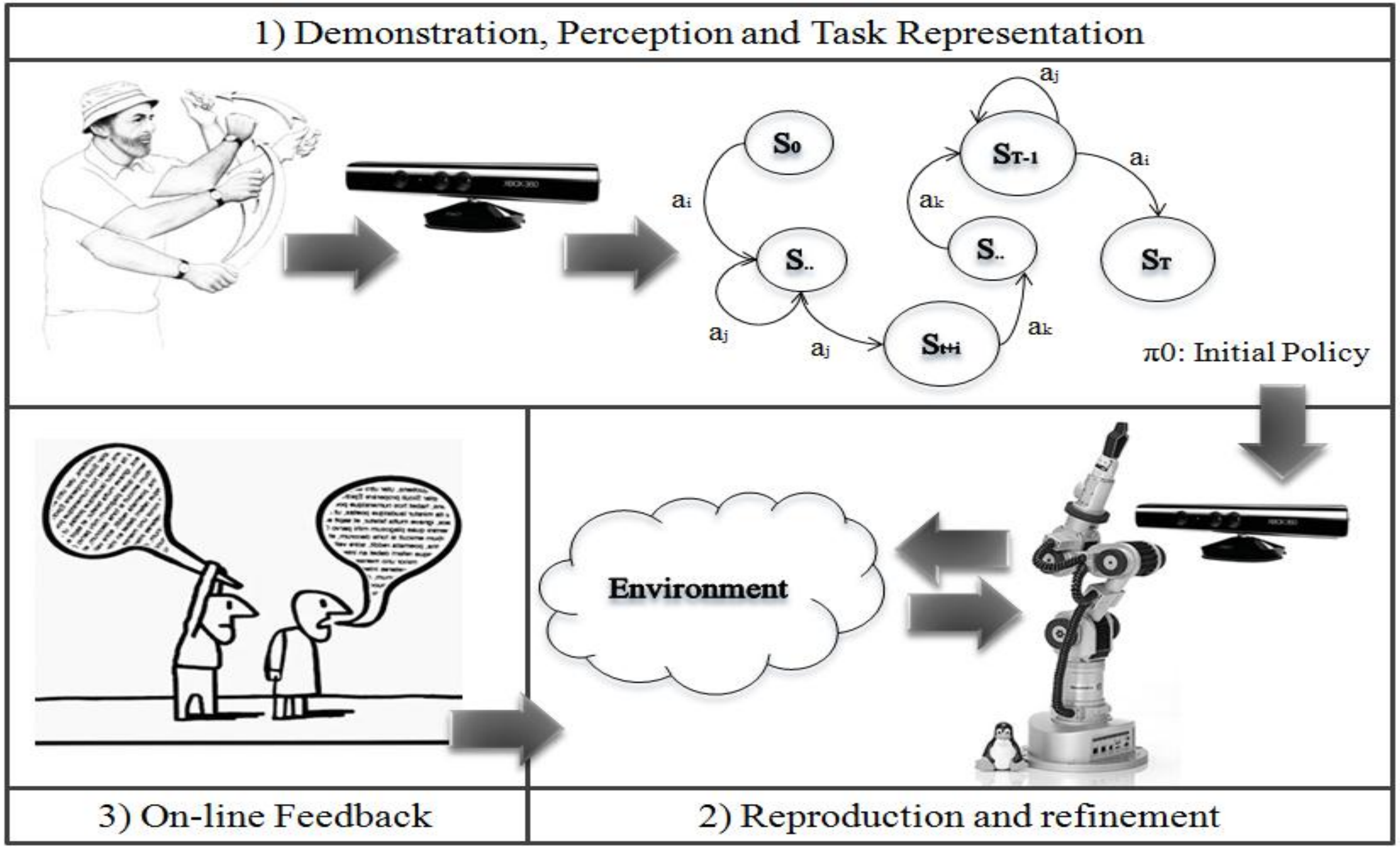
The User Performs The Task

Estimate and track the position of the hand and objects

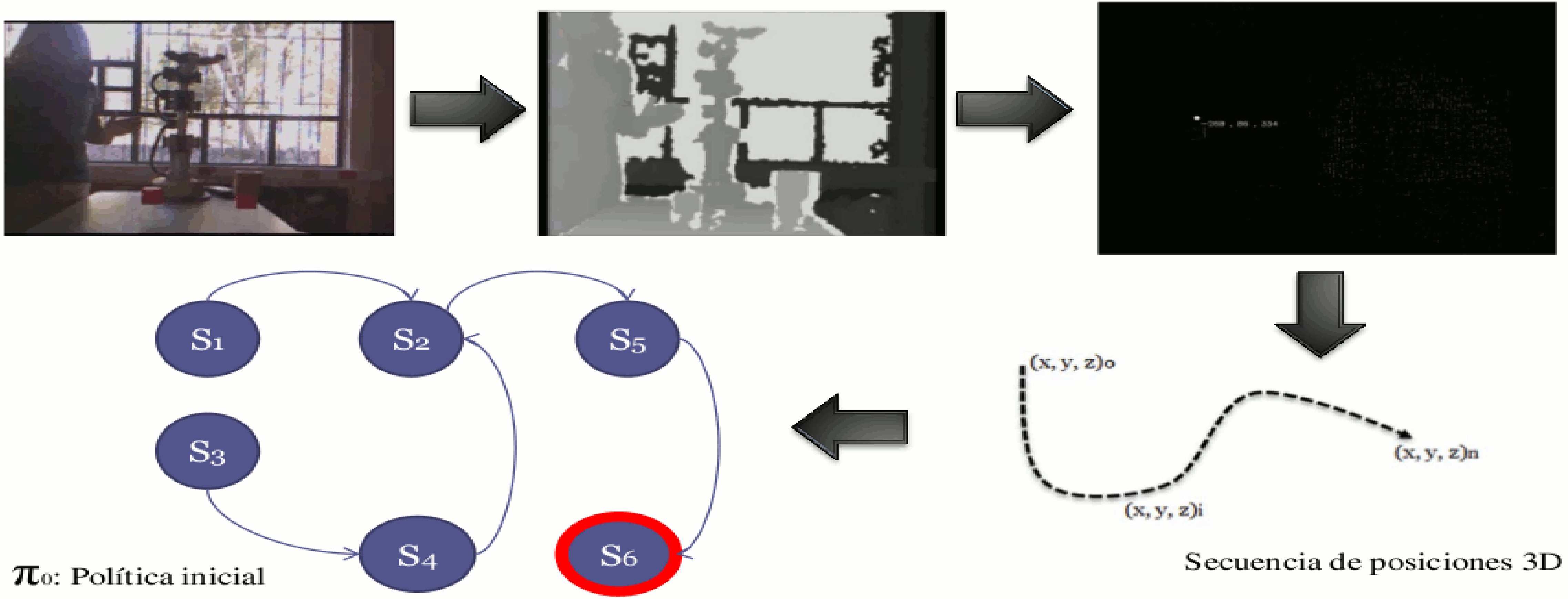
Use a *Kinect*

Representation: relative position and distance between the hand/manipulator and the target object/place

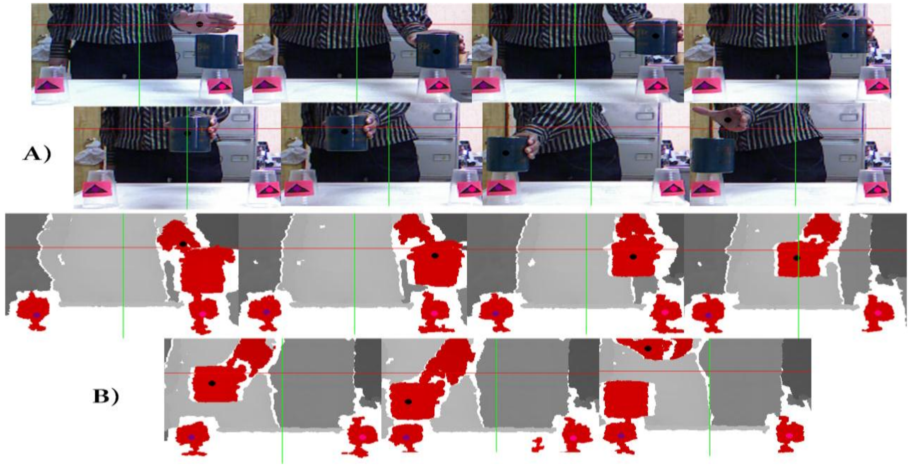
General Learning Framework

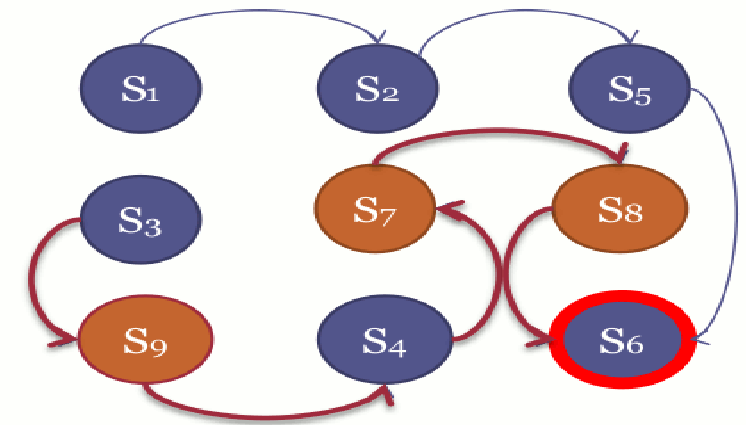


Initial Policy

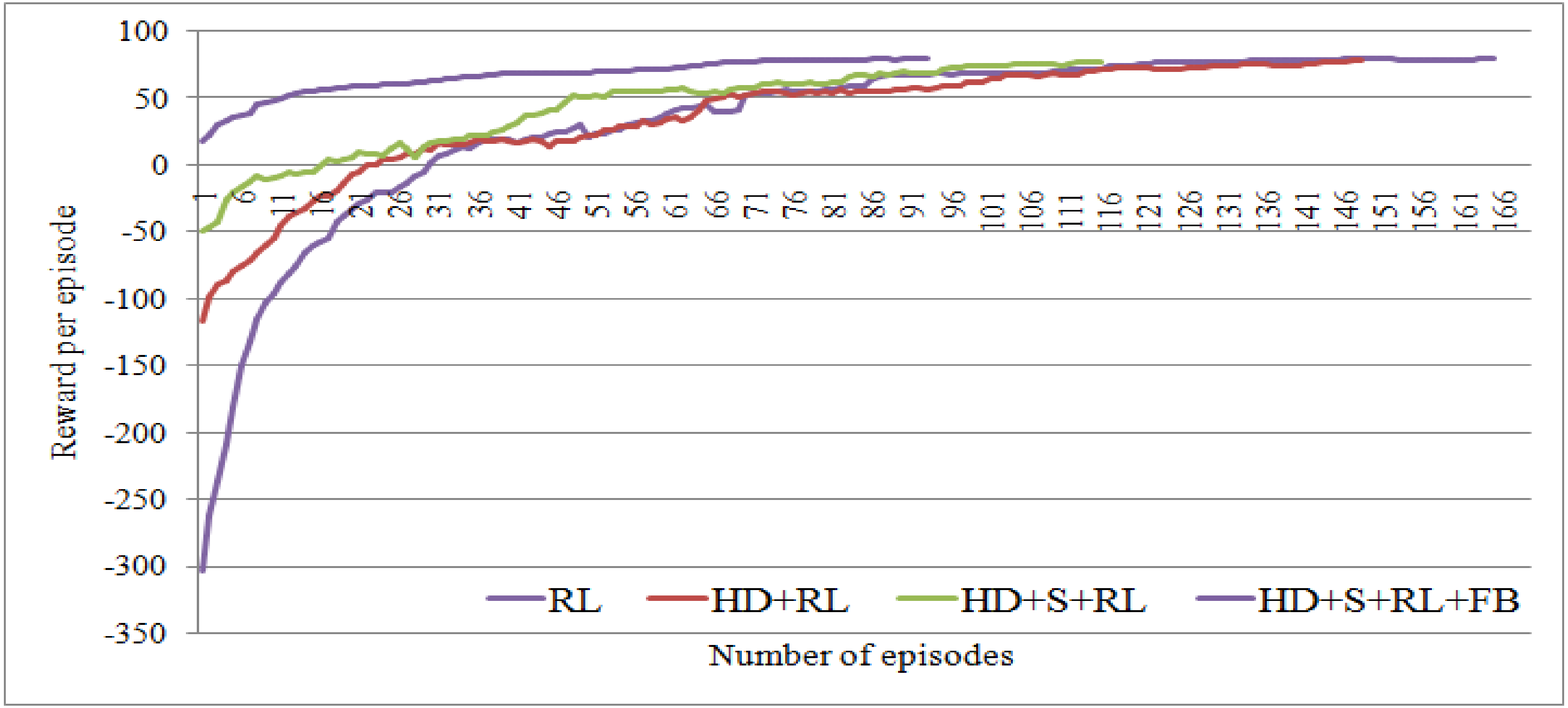


Experimental Setup

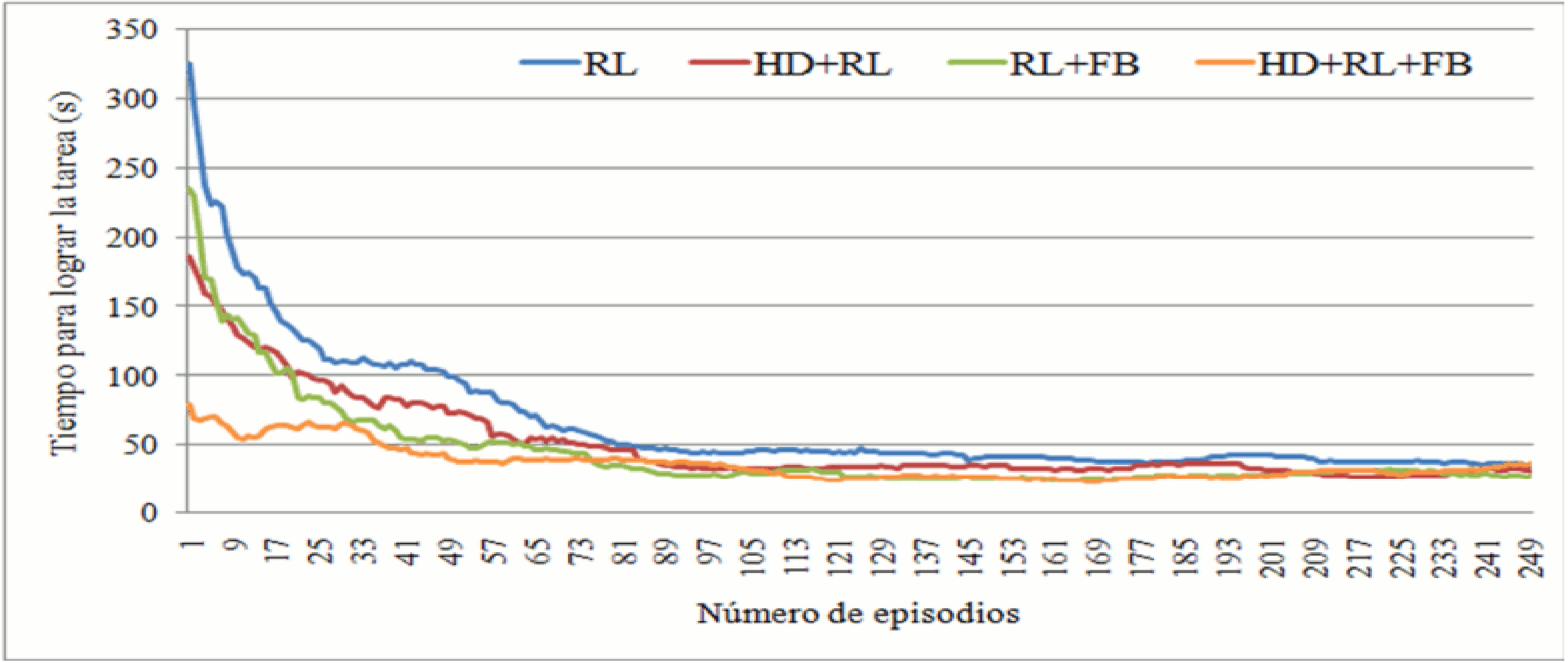




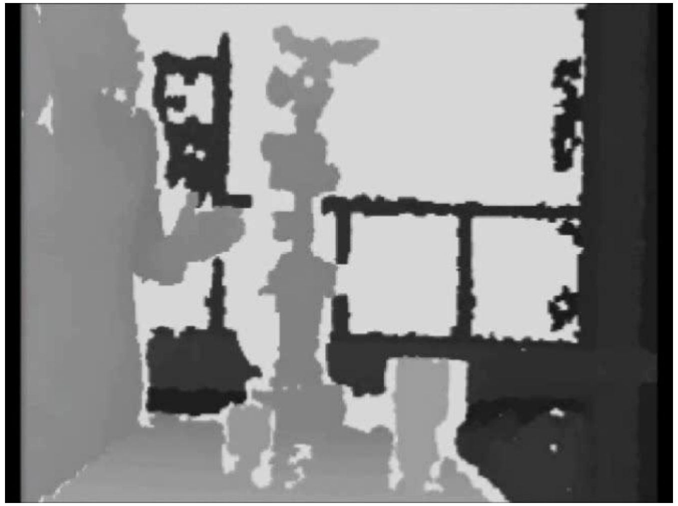
Convergence Results



Convergence Results



Example



Discussion

A relational representation offers more abstracted and natural representation and re-usability of the learned policies

User's traces focus the search space in potentially relevant actions

We loose optimality and completeness (know what to do in every state)

Exploration and voice feedback can help

Conclusions

- The inclusion of service robots into society requires flexibility/adaptability from the robots
- Teaching tasks in a “natural” (manipulate, command or show) way can offer such flexibility

Future Work

Better exploration strategy

Additional user's feedback

More study, tests and formal analysis on feedback during the learning process

Partially observable states

Identify when/how to change the representation

Thanks!

emorales@inaoep.mx