

Visual Search and Recognition for Robot Task Execution and Monitoring

F. Puja, S. Grazioso, L. Mauro, V. Ntouskos,
M. Sanzari, **L. Freda** and F. Pirri

DEPARTMENT OF COMPUTER, CONTROL, AND
MANAGEMENT ENGINEERING ANTONIO RUBERTI



SAPIENZA
UNIVERSITÀ DI ROMA

ALCOR

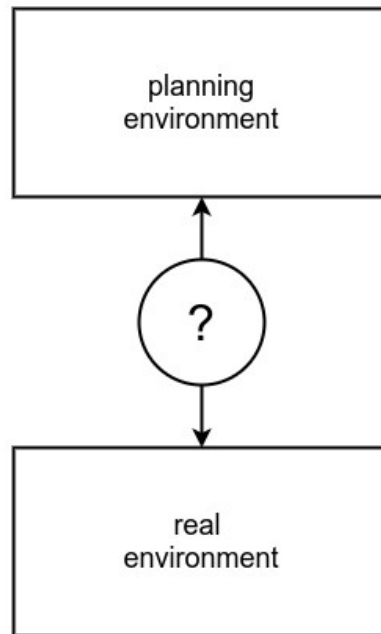
Vision, Perception and Learning Robotics Lab

APPIS 2018 - Las Palma de Gran Canaria, Spain

Problem description

We have a robot executing tasks in a dynamic uncharted environment

- How can we effectively **monitor** the execution of high-level robot actions?
- How can we interleave **perception** and **actions** in a coherent way?
- How can we **refocus** in case a failure occurs?

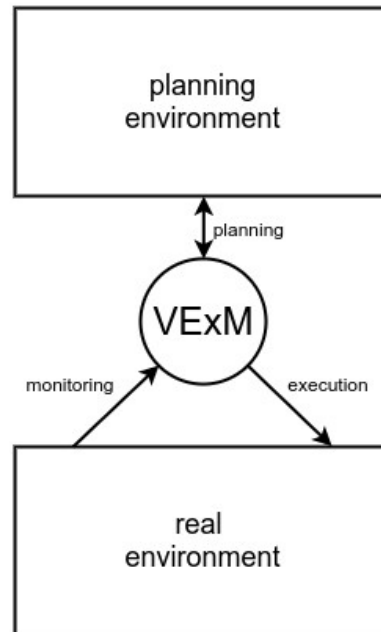


Focus

Visual Execution Monitoring (**VExM**) of high-level robot actions

Our approach

Our VExM acts as an orchestra conductor who gives directives for planning and execution, while at the same time coordinating and monitoring everything



Our approach

The state of the world is visually monitored by verifying **preconditions** and **postconditions** that hold before and after the execution of an **action**

In our visual execution monitoring, we embed a real-time visual system within a hybrid planning model

- The **visual system** recognizes *objects* and *relations* (the visual stream) in order to **focus** on task-relevant objects and **assess** discrepancy between the inferred states and the perceived states
- The **hybrid planning model** blends *deterministic planner* (durable actions) with a *visual search* (for recovering from failures)

Related Works

- Tackle *non-deterministic response* of the environment [1,2]
- Introduce *perception* in execution monitoring [3]
- Present approaches for *recovering* from errors at execution time [5,6,7]

There is a **lack** of a **comprehensive approach** to address all the problems concerning the interleaving of perception and actions in a **coherent** way

[1] O. Pettersson, "Execution monitoring in robotics: A survey," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73–88, 2005. 1

[2] F. Ingrand and M. Ghallab, "Deliberation for autonomous robots: A survey," *Artificial Intelligence*, vol. 247, pp. 10–44, 2017. 1

[3] R. J. Doyle, D. J. Atkinson, and R. S. Doshi, "Generating perception requests and expectations to verify the execution of plans." 11

[5] S. Guadarrama, L. Riano, D. Golland, D. Go, Y. Jia, D. Klein, P. Abbeel, T. Darrell, et al., "Grounding spatial relations for humanrobot interaction," in (IROS 2013), 2013, pp. 1640–1647. 2

[6] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in ECCV, 2016, pp. 852–869. 1, 2

[7] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: Do humans and deep networks look at the same regions?" arXiv preprint arXiv:1606.03556, 2016. 2

Main Contributions

- **Enhanced Execution Monitoring**, visually monitoring the state of the world in terms of action preconditions and postconditions
- **Extended relation recognition** using networks active features of the recognized objects
- **New minimal representation of the state of the world**, with the creation of “*mental maps*”, images containing objects, relationships and depths from the robot point of view

VExM

Task: take the Spraybottle and hand it to Person

The task is executed in several stages:

- search(robot,spraybottle)**: The robot searches for the spray bottle on the table.
- approach(robot,table)**: The robot moves towards the table.
- grasp(robot,spraybottle)**: The robot grasps the spray bottle.
- search(robot,spraybottle)**: The robot searches for the spray bottle on the floor.
- grasp(robot,spraybottle)**: The robot grasps the spray bottle from the floor.
- search(robot, person)**: The robot searches for a person in the room.
- approach(robot, person)**: The robot moves towards the person.
- handover(spraybottle, person)**: The robot hands the spray bottle to the person.

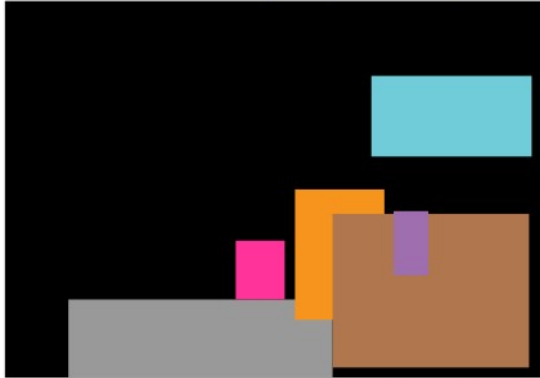
Legend:

- backpack
- blackboard
- chair
- floor
- hand
- person
- robot hand
- spray bottle
- table
- toolbox
- TV-monitor

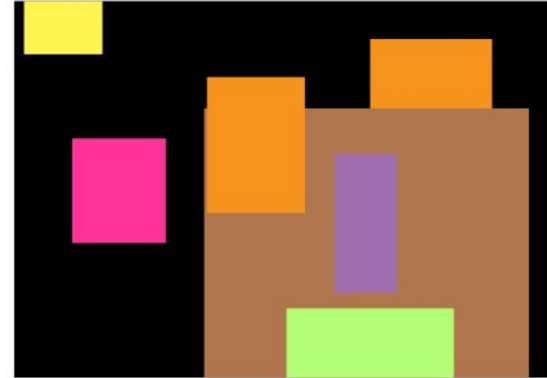
VExM

actions from planner -->

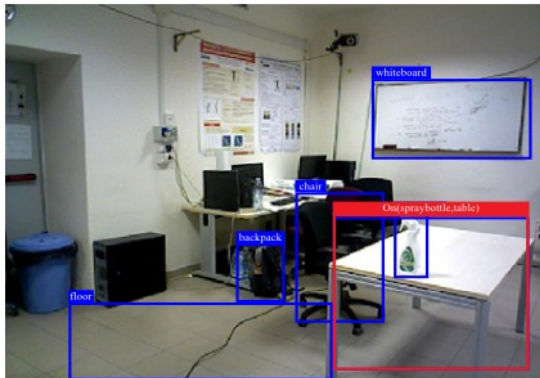
search(robot,spraybottle)



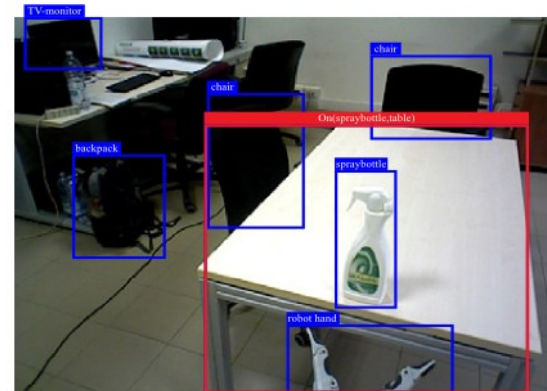
approach(robot,table)



- backpack
- blackboard
- chair
- floor
- hand
- person
- robot hand
- spraybottle
- table
- toolbox
- TV-monitor



Found(robot,spraybottle)
On(spraybottle,table)



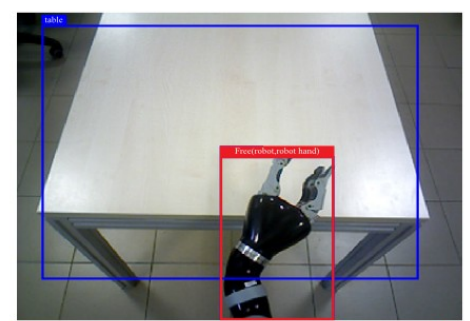
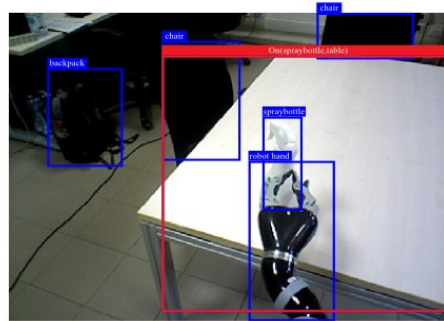
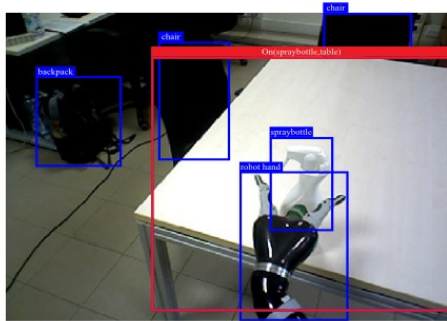
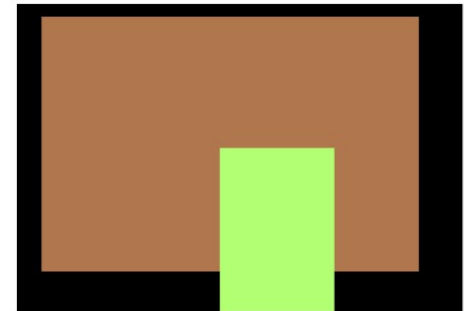
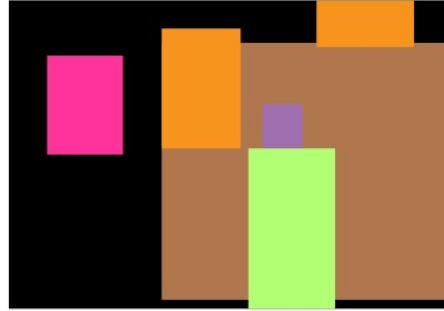
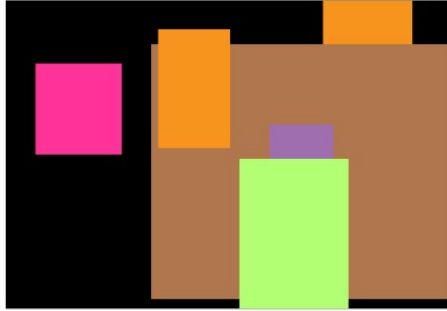
CloseTo(robot,table)

action postconditions -->

VExM

actions from planner -->

`grasp(robot,spraybottle)`



action postconditions -->

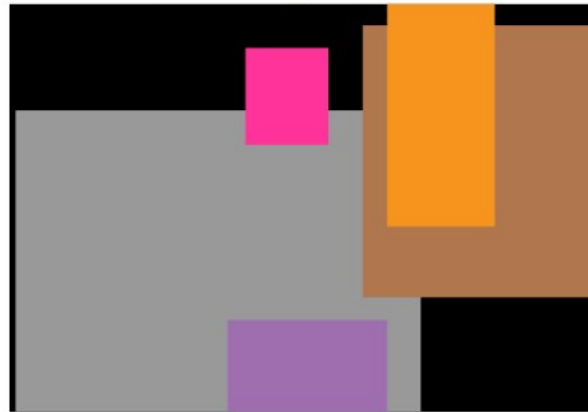
Holding(robot,spraybottle)
Free(robot,robot hand)

- backpack
- blackboard
- chair
- floor
- hand
- person
- robot hand
- spraybottle
- table
- toolbox
- TV-monitor

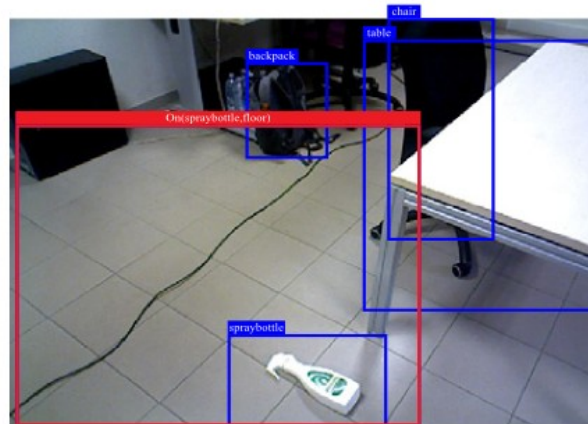
VExM

actions from visual search -->

search(robot,spraybottle)



- backpack
- blackboard
- chair
- floor
- hand
- person
- robot hand
- spraybottle
- table
- toolbox
- TV-monitor



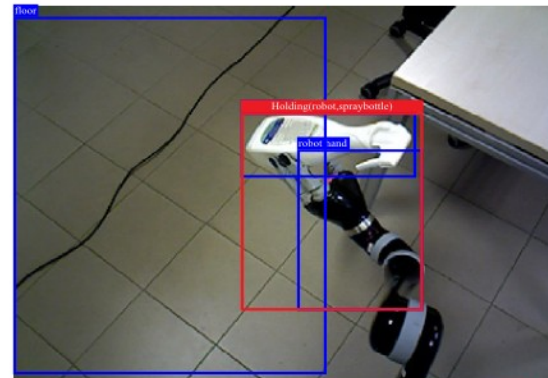
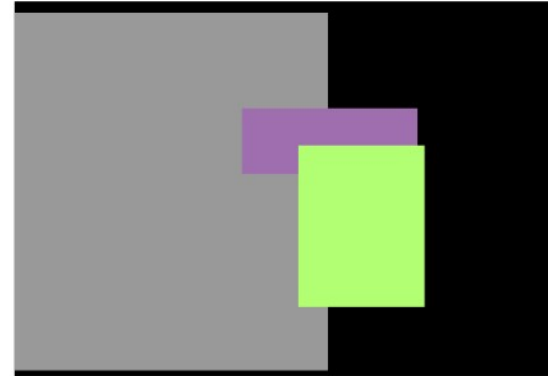
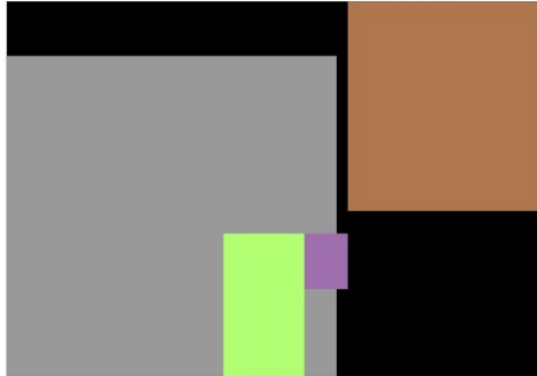
action postconditions -->

Found(robot,spraybottle)
On(spraybottle,floor)

VExM

actions from planner -->

grasp(robot,spraybottle)



action postconditions -->

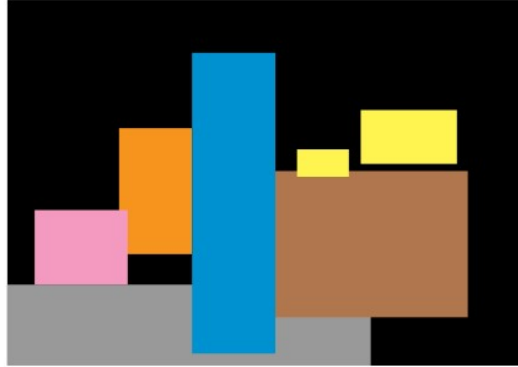
Holding(robot,spraybottle)

- backpack
- blackboard
- chair
- floor
- hand
- person
- robot hand
- spraybottle
- table
- toolbox
- TV-monitor

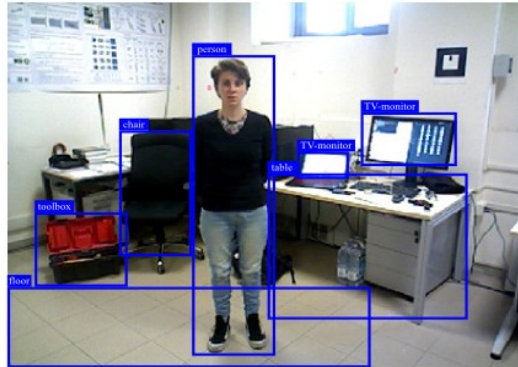
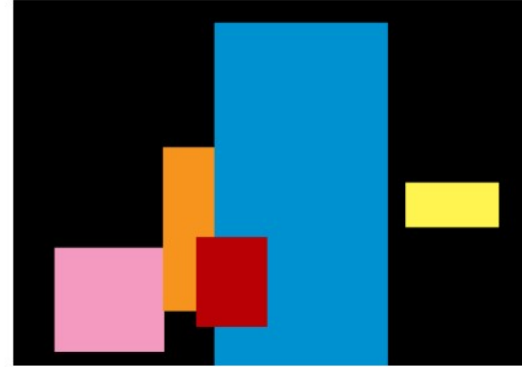
VExM

actions from planner -->

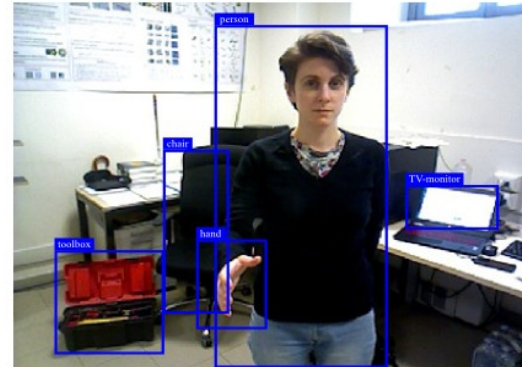
search(robot, person)



approach(robot, person)



Found(robot, person)



CloseTo(robot, person)

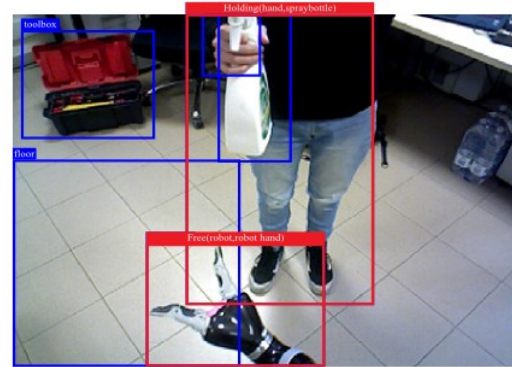
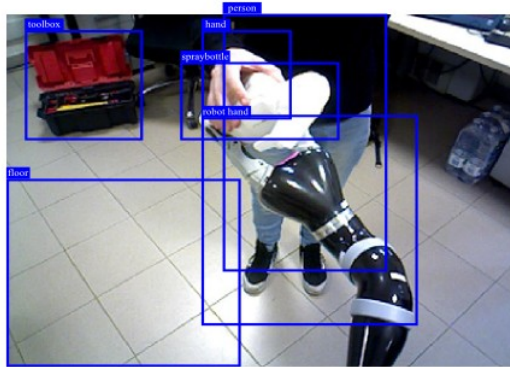
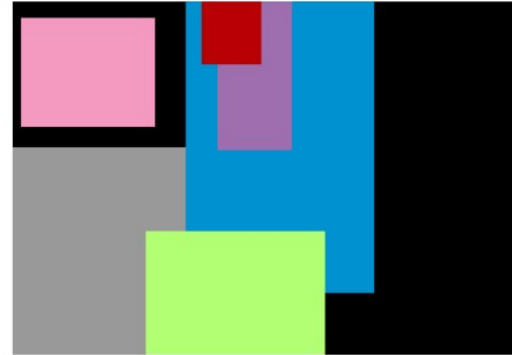
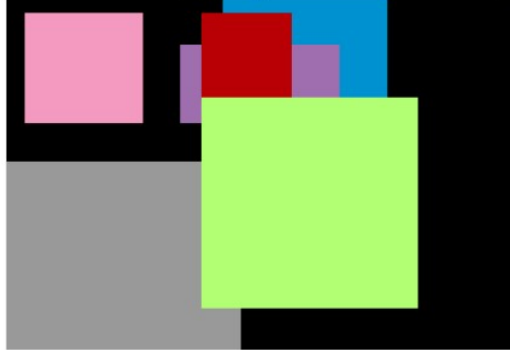
action postconditions -->

- backpack
- blackboard
- chair
- floor
- hand
- person
- robot hand
- spraybottle
- table
- toolbox
- TV-monitor

VExM

actions from planner -->

handover(spraybottle, person)



action postconditions -->

Free(robot, robot hand)
Holding(person, spraybottle)

- backpack
- blackboard
- chair
- floor
- hand
- person
- robot hand
- spraybottle
- table
- toolbox
- TV-monitor

Visual Execution Monitoring

The **VExM**:

- **connects** the planning environment and the real world environment
- **generates** and **coordinates** the Planning and Execution information flow
- **monitors** the current state and the executed action

The system is **hybrid** because comprises:

- **Deterministic** *planner* based on FastDownward [10]
- **Non deterministic** *visual interpretation* of preconditions and postconditions of actions
- **Non deterministic** *visual search policy* for recovering from failures

Deterministic Component

The **deterministic Planner** is designed in **PDDL**, a series of objects, relations and actions, with their pre- and post- conditions are defined

The robot **task** gets split in several goals, represented as problems with specific initial conditions

FastDownward planner [10] is used to infer **sequence of actions** leading to the goals

[10] M. Helmert, "The fast downward planning system." (JAIR), vol. 26, pp. 191–246, 2006. 5

Non-Deterministic Component

It's composed of **two sub-components**:

- The **visual system**, a real time system for *recognition of objects and relations* (visual stream) designed to monitor the state of execution
- The **visual search policy**, required to focus the robot toward the item of interest, so as to let the visual stream to deal with the preconditions of the action to be executed and to recover from failure

Deterministic Planner

designed in PDDL

- **domain** composed of objects, relations and actions
- **actions** definition, specifying the preconditions and postconditions of each action
- robot **task** is split in several goals, represented as problems with specific initial conditions
- for each **goal**, if a **plan** leading from initial state to the goal exists, the *FastDownward* planner infers a **sequence of actions** leading to the goal
- the planning and **ordered execution** of **plans** constitute the roadmap for completing a whole task

Non-Deterministic Component

It's composed of **two sub-components**:

- The **visual system**, a real time system for *recognition* of *objects* and *relations* designed to monitor the state of execution
- The **visual search policy**, required to *focus* the robot toward the item of interest, so as to let the visual stream to deal with the preconditions of the action to be executed and to *recover* from failure

Non-Deterministic Component

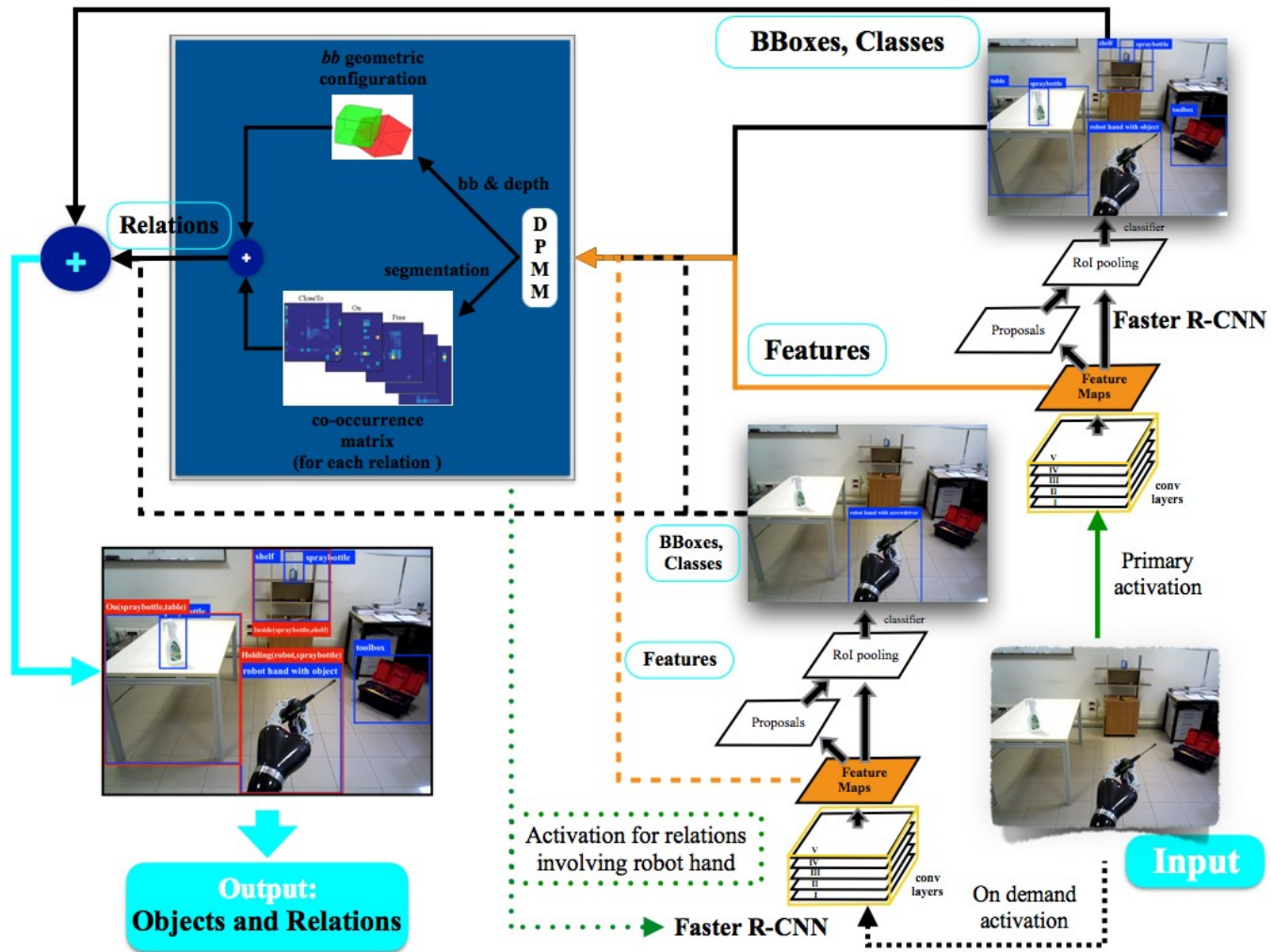
Visual System

- **Two DCNNs** for **detecting** the **objects** that the robot has to deal with and manipulate
- A **non parametric Bayes** estimation to **discover** the **relation** amid objects from DCNNs features

Visual Search

- **Mental Maps** that **represents** the **state of the world** from the robot point of view as input to the deep reinforcement Learning
- An adapted **Deep Q-Network** [9] to learn **visual search policies** to recover from failure or lack of focus trained on a virtual environment

Visual System structure



The Visual search policy

The **visual search policy**:

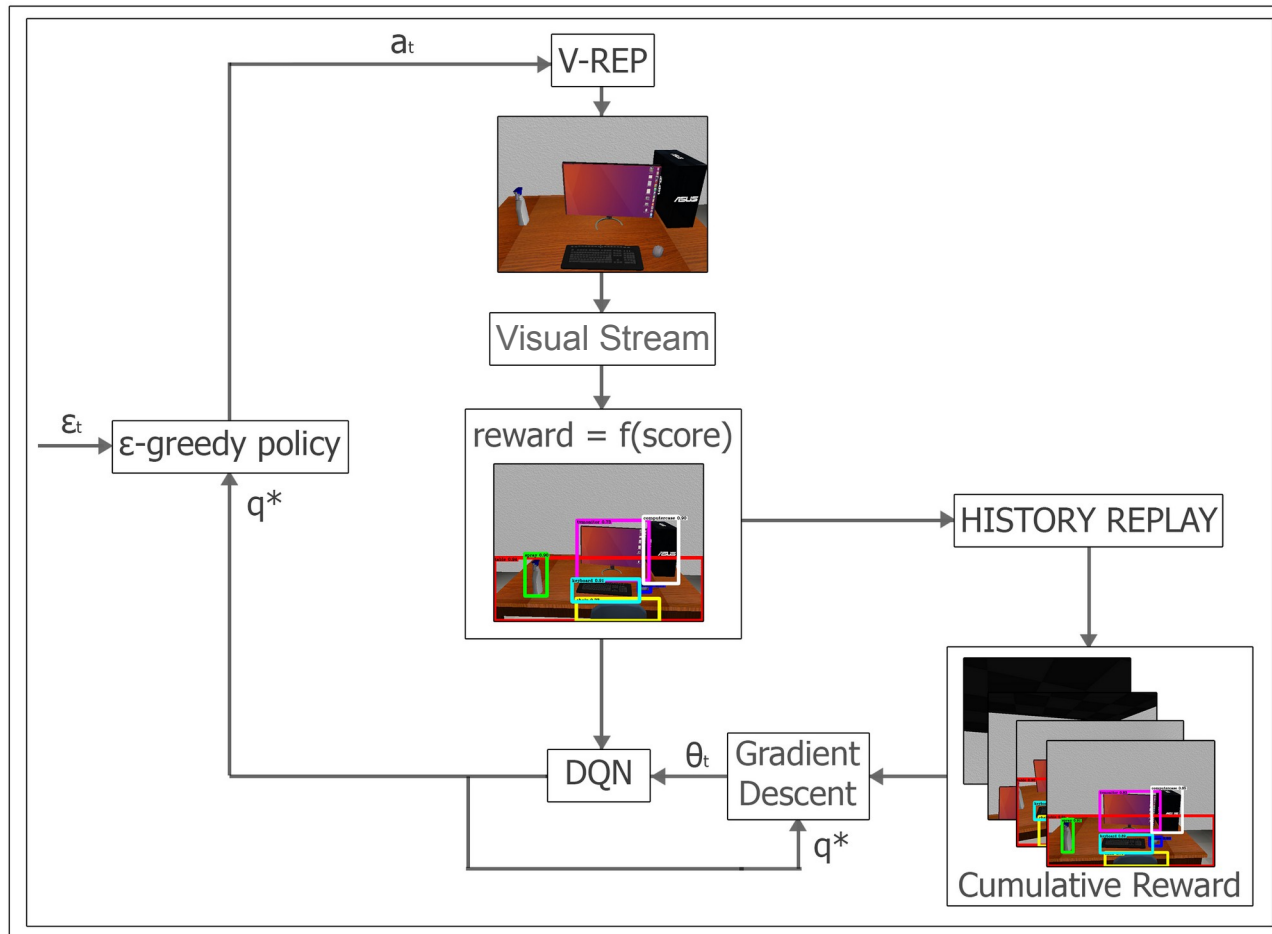
- Uses a **Deep Q-Network** [9] adapted to receive RGB input
- Learns to compute an estimation of the future rewards associated with each possible action using a **deep reinforcement learning paradigm**
- The sequence of actions with the highest prediction is the ***optimal policy***

The huge amount of experiments needed during the learning phase are performed using a virtual environment created with the **V-Rep** Simulation Software

The **Visual System** component is used to create:

- the **Mental Maps**, used as an **input** to the DQN
- the **reward** calculated in relation to the **object detection score**

The Visual search policy training



Experiments

Experiments have been done with a custom made robot built with:

- A **Pioneer 3 DX** differential-drive robot as a mobile base
- A **Kinova Jac 2** with 6 degrees of freedom and a reach of 900mm as robotic arm to interact with the environment
- An **Asus Xtion PRO** live RGB-D as camera, mounted on a Direct Perception PTU46-17.5 **pan-tilt unit**

Experiments

The task considered have been:

- **Bringing** an object on the table or inside the shelf
- **Put away** an object in the toolbox

Each experiment in a class has been run:

- 35 times **manually** driving the robot to collect images of the scene
- 20 more times with the **hybrid planning** environment
-
- 120000 images have been collected from **robot experiments** and
- 25000 from **ImageNet**

TABLE I

SUBSET OF OBJECTS, RELATIONS, SUPPLY-ACTIONS AND EGOCENTRIC ACTIONS OF THE ROBOT LANGUAGE \mathcal{L}

Relations	Objects	Supply actions	Egocentric actions
CloseTo	Bottle	Close	Look-down
Found	Chair	Grasp	Look-left
Free	Cup	Open	Look-right
Holding	Floor	Hand-over	Look-up
Inside	Hammer	Place	Move-forward
On	Person	Lift	Move-backward
InFront	Spray Bottle	Push	Turn-left
Left	Screwdriver	Spin	Turn-right
Right	Shelf	Dispose	Localize
Under	Toolbox		Rise-arm
Behind	TV-Monitor		Lower-arm
Clear	Table		Close-Hand
Empty	Door		Open-Hand

Experiments

The DCNNs models is trained using images taken from the **ImageNet** dataset and collected by the ASUS Xtion PRO RGB-D **camera**

We split the set of images in training and validation sets with a proportion of 80%-20%

We performed 70000 training iterations for each model on a PC equipped with 8 GPUs

The DQN training is done with the use of the **V-Rep simulation** software with over 5 millions of iterations

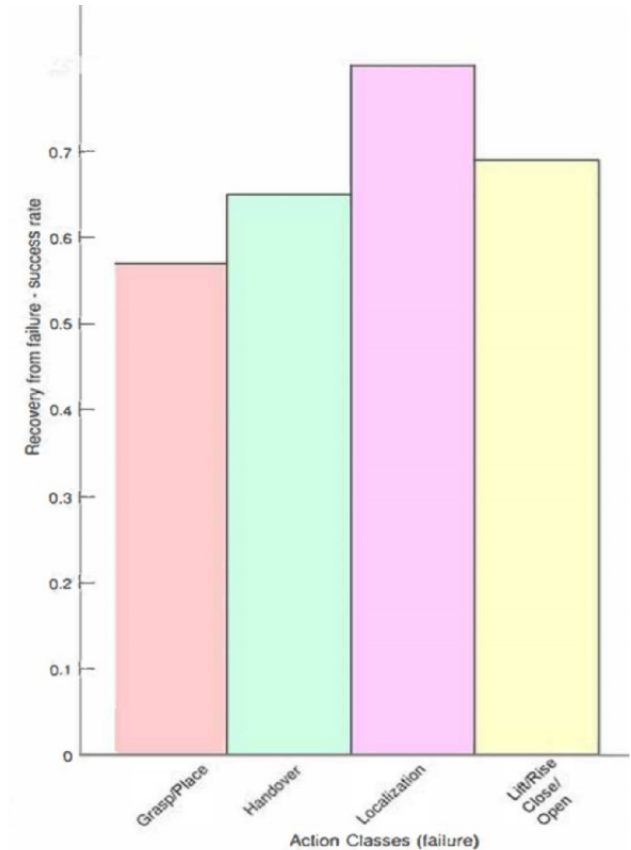
Experiments

TABLE III
RELATIONS DETECTION ACCURACY

Relations	Accuracy
CloseTo	64%
Found	73%
Free	62%
Holding	79%
Inside	75%
On	83%
] InFront	78%
Left	68%
Right	72%
Under	71%
Behind	89%
Clear	67%
Empty	76%
Average	73.6%

TABLE II
OBJECT DETECTION ACCURACY

Objects	Free	Holding
bottle	72%	61%
chair	69%	-
cup	57%	53%
floor	84%	-
hammer	78%	66%
person	71%	-
spraybottle	84%	72%
screwdriver	77%	66%
shelf	69%	-
toolbox	64%	-
TV-Monitor	91%	-
table	76%	-
door	81%	-
Average	75%	64%



Experiments

The presented approach addresses Execution Monitoring as an hybrid system for both executing and monitoring specifically the **targeting visual perceptions**

The system reaches

- 75% in object detection and
- 73.6% in relation detection accuracy

In case of **failures** the VExM resorts into a state where execution can be retrieved according to the learned policies from DQN with a success rate of 70%

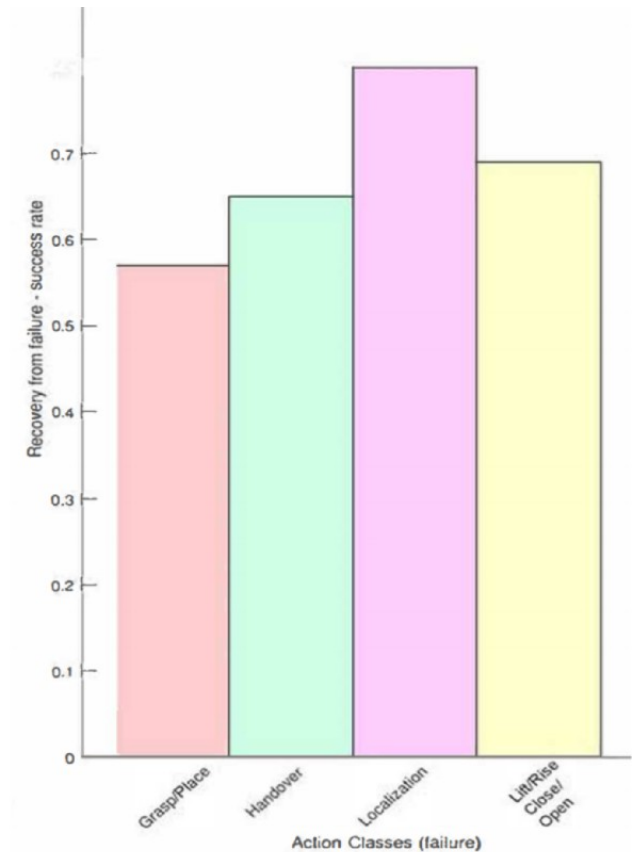
Experiments

TABLE III
RELATIONS DETECTION ACCURACY

Relations	Accuracy
CloseTo	64%
Found	73%
Free	62%
Holding	79%
Inside	75%
On	83%
] InFront	78%
Left	68%
Right	72%
Under	71%
Behind	89%
Clear	67%
Empty	76%
Average	73.6%

TABLE II
OBJECT DETECTION ACCURACY

Objects	Free	Holding
bottle	72%	61%
chair	69%	-
cup	57%	53%
floor	84%	-
hammer	78%	66%
person	71%	-
spraybottle	84%	72%
screwdriver	77%	66%
shelf	69%	-
toolbox	64%	-
TV-Monitor	91%	-
table	76%	-
door	81%	-
Average	75%	64%



Conclusions

- Hybrid system for both monitoring the execution and specifically targeting visual perception
- VExM refocuses and recover from a failure according to **DQN**
- Failure case: VExM resort to a recovery policy that ensures an optimized visual search and to resort into a state where execution can be retrieved

Bibliography

- [1] O. Pettersson, “Execution monitoring in robotics: A survey,” *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73–88, 2005. 1
- [2] F. Ingrand and M. Ghallab, “Deliberation for autonomous robots: A survey,” *Artificial Intelligence*, vol. 247, pp. 10–44, 2017. 1
- [3] R. J. Doyle, D. J. Atkinson, and R. S. Doshi, “Generating perception requests and expectations to verify the execution of plans.” 1
- [4] D. E. Wilkins, “Recovering from execution errors in sipe,” *Computational. Intelligence*, vol. 1, no. 1, pp. 33–45, 1985. 1
- [5] S. Guadarrama, L. Riano, D. Golland, D. Go, Y. Jia, D. Klein, P. Abbeel, T. Darrell, et al., “Grounding spatial relations for humanrobot interaction,” in (IROS 2013), 2013, pp. 1640–1647. 2
- [6] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, “Visual relationship detection with language priors,” in *ECCV*, 2016, pp. 852–869. 1, 2
- [7] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra, “Human attention in visual question answering: Do humans and deep networks look at the same regions?” *arXiv preprint arXiv:1606.03556*, 2016. 2
- [8] T. S. Ferguson, “A bayesian analysis of some nonparametric problems,” *Ann. Stat.*, pp. 209–230, 1973. 1, 2, 3
- [9] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. 1, 2
- [10] M. Helmert, “The fast downward planning system.” (*JAIR*), vol. 26, pp. 191–246, 2006. 5

Thank you!



The Visual System

The visual system is built with a hierarchical model combining:

- two deep convolution neural networks (**DCNNs**) [11] for object recognition
- a non parametric Bayes model (**DPM**) [8] that segment collected depth images using DCNN features.

The **combination** of segmented depth and object labels allows to **infer** visual **relations** from the robot point of view.

All the information can be then represented in a singular picture called “**Mental Maps**”.

