MASH

Massive Sets of Heuristics for Machine Learning

http://mash-project.eu

–

Seventh Framework Programme – ICT

Small or medium-scale focused research

Grant agreement No. 247022

Deliverable D4.1-m9: Operational robot

Due date:	30/09/10	**Submission date:**	30/09/10
WP leader:	CVUT	**Revision:**	1
Project start:	1/1/2010	**Duration:**	36 months

Authors: Vladimír Smutný, Martin Dubec

Dissemination Level		
PU	Public	PU ×
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

D4.1-m9: Operational robot

Vladimír Smutný, Martin Dubec

November 3, 2010

Abstract

The deliverable D4.1-m9 per se is the operational robot. This report just documents some of its technical properties.

This document describes the setup, components, functionality and main features of the robotic platform. The platform will demonstrate results of the project on the practical example.

This report overviews the physical setup of the robotic cell. The definition of the software interface which defines the semantics of protocol between the Experiment server and the Application server.

1 Introduction

The deliverable D4.1-m9 per se is the operational robot. This report just documents some of its technical properties.

The robotic platform is a part of the coordinated effort in the MASH project. The robotic platform role is described in Annex 1, Part B, Description of Work of the project documentation.

In a short the robot platform acts as physical embodiments of the general machine, which demonstrates the capabilities of computer vision, pattern recognition, machine learning, and goal planning algorithms implemented during the project.

The robotic platform operates simple world of blocks lying on the table. The robot is operated and controlled remotely from Experiment server by its users distributed all over the world.

The software controlling the robot is called Robot Application server within MASH project.

The initial standard experiment by the robotic cell has the following scenario:

1. **Task definition** The Experiment server asks Application server for initialization of the robot and selects the task from the list and parameterizes it.

2. **Initialization** The robot distributes blocks stored in Store into the playground according the rules described in task. The robot puts itself into the starting position.

3. **Loop** The Experiment server controls the robot by commands e.g. Go North, Go South,... and the robot follows commands. After each command the images captured by the cameras are sent to the Experiment server. The Application server also sends information about the current situation in the field as a reward.

4. **Cleaning** When the Experiment server finishes experiment, the robot cleans the field and returns all blocks into their predefined position in the store area.

5. **Ready** The robotic cell is ready for new experiment after cleaning.

2 Setup description

2.1 Robot

The robot is 6 degree of freedom angular manipulator. It is equipped with a magnet to manipulate magnetic blocks and a gripper. The maximum size of the grasped objects is about 60 mm between fingers. The maximum weight of the blocks is about 1 kg. The robot manipulates the blocks lying on the horizontal table around the robot.

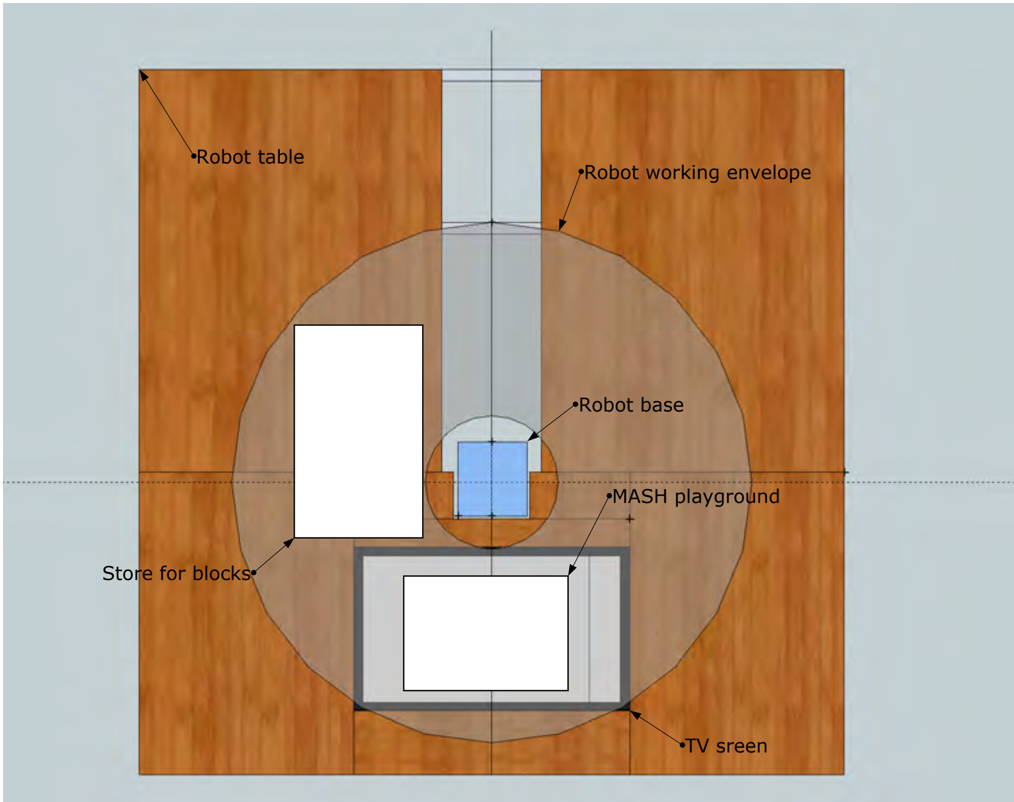

Figure 1: Top view on the robotic cell layout. Robot is in the center operating mainly in the MASH playground and store area. The TV screen is located bellow playground to allow to change background image.

2.2 MASH playground

The region of the table in front of the robot is a playground, where MASH algorithms operates. The playground is a physical space satisfying following criteria:

- **Background** The TV screen located under the playground displays different images which effectively make a controlled background of image processing algorithms. The displayed images are controlled by the Experiment server. The controlled background also simplifies cleaning procedure when experiment is reset.
- **Robot manipulated** The robot has to be able to reach and manipulate the objects located within the playground.
- **Camera observed** The images of playground scene are captured by cameras located in the robotic cell.
- **Easy to describe** To simplify the description of playground area, the playground is a rectangle whose axis are aligned with the robot base axes.

The back side of the robot table serves as a store of the blocks.

The rectangular area of the table in front of robot (see Fig. 1) is dedicated to be a MASH playground. This area is divided logically into the lattice of points, where the blocks could be placed. This logical division is used to allow simplified description of the task and the planing algorithm. The blocks are supposed to be in a single layer. Both restrictions are on the SW layer and could be released in future set of tasks if that proves necessary or useful.

2.3 Store

The area behind the robot is intended as a store of blocks. The printed plan allows to manually place the blocks into the store and the robot could blindly grasp for them. The robot will take blocks from the store during initialization and return them after the experiment during routine use.

2.4 TV screen

TV screen is placed underneath the playground to allow to change background appearance during advanced experiments. TV screen has a HD resolution (1920x1080 pixels).

2.5 Cameras

One camera is placed just above playground to capture approximately top view of the playground. See Fig. 2.

Second camera is capturing oblique view of the playground.

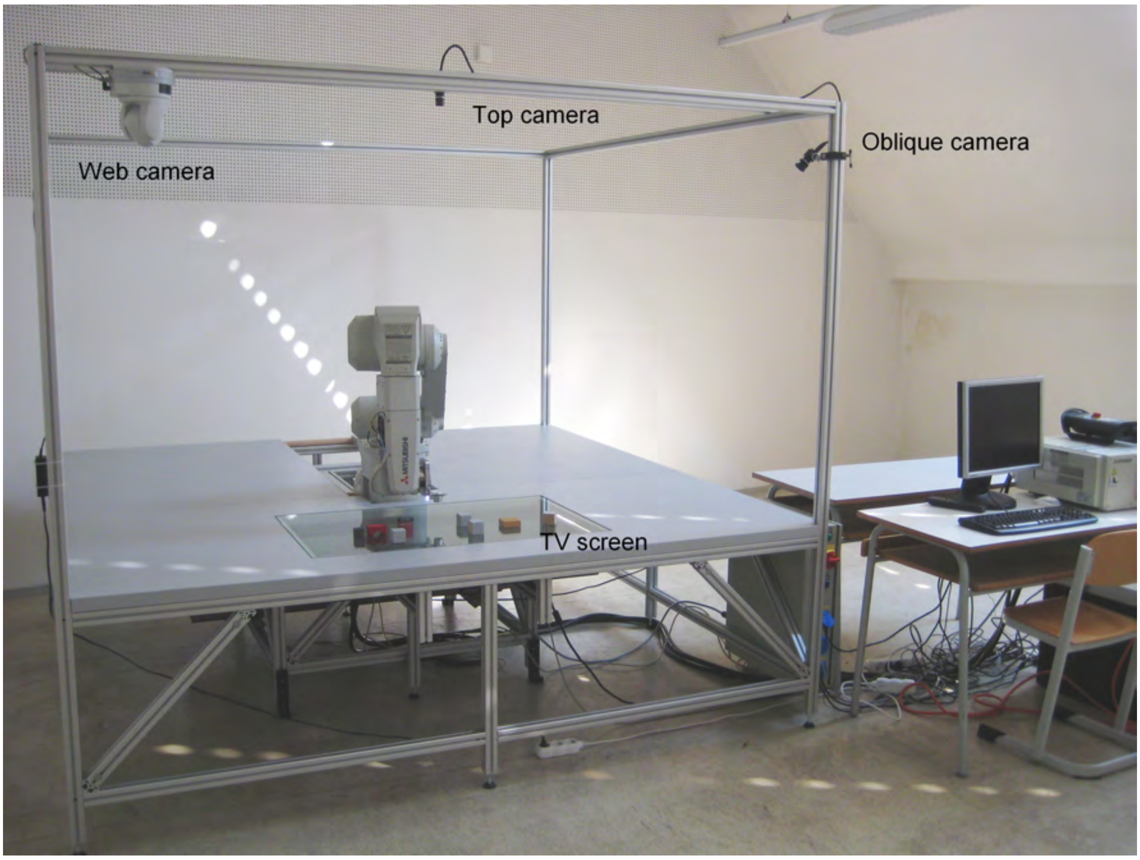

Figure 2: Photo of the MASH robotic cell. Note the camera above the playground. The second camera is on the right pillar capturing the oblique view. The pan and tilt web camera is located on the upper beam of the cell construction. The TV screen is located bellow the playground behind the a protecting glass.

2.6 Web camera

The web pan and tilt camera allows to check the situation of the robotic cell independently to Experiment and Application server infrastructure.

The access to the camera will be allowed to people experimenting with it.

3 Robot platform Application server interface

Experiment and Application server communicate in MAS protocol described elsewhere. MAS protocol uses following terms:

- Task
- Environment
- Goal
- Action
- View

The semantics of those terms in the context of the robotic platform is defined in following section.

3.1 Task

Task is specified by *goal* and the appropriate *environment*. The environment comprises a geometrical configuration, placements of blocks, etc. all defined with some random measurements to ensure an infinite variety of the setup. From that goal derives a reward, negative when the robot gripper collides with the environment and positive otherwise. When the goal is reached, the task is over, and another round can be started.

3.2 World

World is a two dimensional space where robot can move. It is defined as a rectangle. Dimensions and position of the *world* depends on the environment. *World* uses its own coordinates called lattice coordinates.

3.3 Environment

Environment contains informations about available views, actions, goals. *Environment* specifies properties which are described in the MAS protocol:

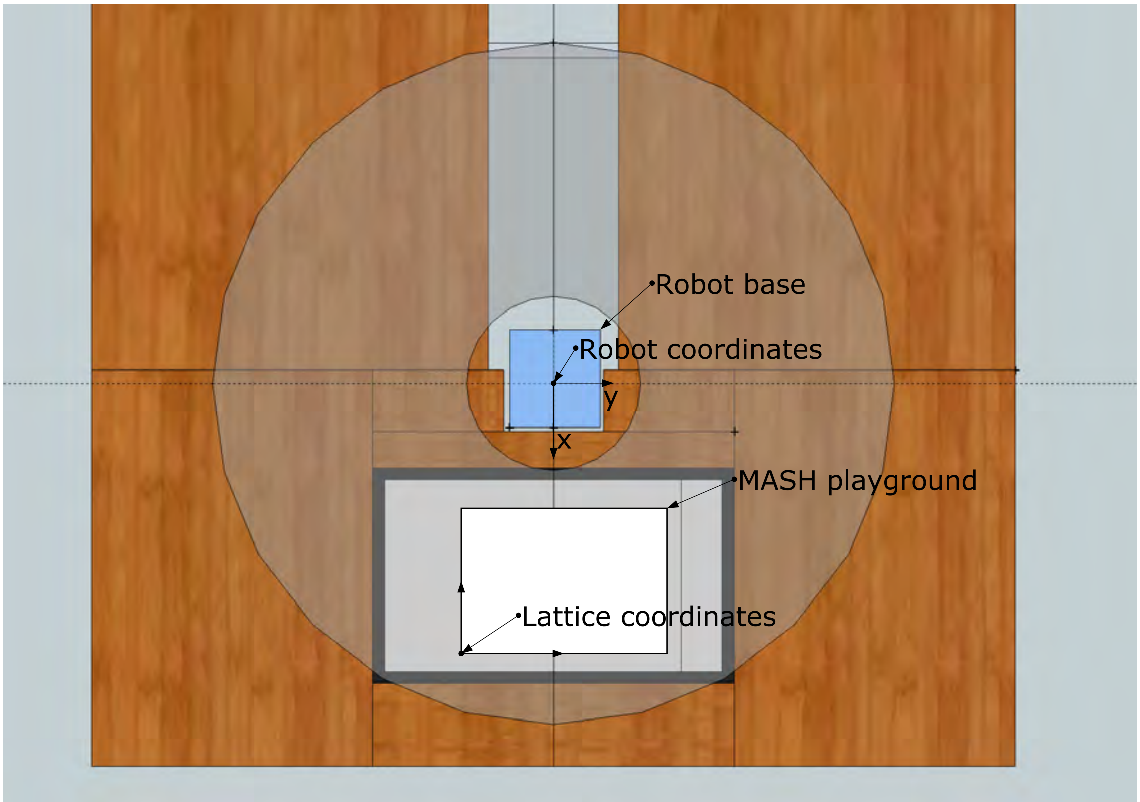

Figure 3: Coordinates of the robot (measured in millimeters) and lattice coordinates (measured in lattice units).

- **unique name** - unique identifier of environment. The environment is chosen by its name during initialization.
- **supported goals** - list of goals, which are supported by appropriate environment.
- **supported actions** - list of actions, which are supported by the environment.
- **supported teaching** - true if teaching is supported in the environment.
- **supported views** - list of cameras to obtain images of the playground

  Properties of the environment specific to robot applications server:

- **tv image** - image which will be displayed on the TV screen.
- **border** - specifies dimensions and position of the *world* in the lattice coordinates. Border is an array of points in lattice coordinates in order lower-left corner, lower-right, upper-right, upper-left,
- **blocks** - initial positions and types of blocks specified by block type and its position $[x, y, \phi]$.

The currently implemented environments in robot application server:

- `STATIC_12x12` - world size is $[12 \times 12]$, environment contains 3 blocks,
- `STATIC_8x6` - dimensions of the world are $[8 \times 6]$, environment contains 3 blocks,
- `STATIC_4x4` - dimensions of the world are $[4 \times 4]$, environment contains 2 blocks,
- `RANDOM_12x12` - world size is $[12 \times 12]$, block positions and types are random.

3.4 Goal

Goal defines properties:

- start position - start position of the robot gripper in lattice coordinates after initialization of the *task*,
- end position - position of the target block in lattice coordinates,
- target block - type of the target block.

Currently implemented goals in robot application server :

- `ReachRedCube` - goal is to find a red block in the world. Position of the target block is always in upper-right corner of the world. The reward is +20.
- `ReachRedCube_Random` - same as goal `ReachRedCube` but only position of the block is random. The reward is +20.

3.5 Action

Action move robot tool to the neighboring cell in the lattice coordinates. Robot is able to move in four directions :

- UP
- DOWN
- LEFT
- RIGHT

Action moves the gripper 27.744mm up (respectively down, left, right). *Reward* is computed after action:

- -10 robot hit the wall,
- 0 robot performed allowed move,
- $+20$ robot reached the target.

3.6 View

Currently implemented views:

- CAMERA_TOP - top camera, resolution: 1280x960, output format: MIF
- CAMERA_OBLIQUE - oblique camera, resolution: 1280x960, output format: MIF