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 D4.2-m12: First series of robotic tasks

Due date:	31/12/2010	**Submission date:**	XXXXXXX
WP leader:	CVUT	**Revision:**	1
Project start:	1/1/2010	**Duration:**	36 months

Authors: Vladimír Smutný, Martin Dubec

Dissemination Level		
PU	Public	×
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)	

D4.2-m12: First series of robotic tasks

Vladimír Smutný, Martin Dubec

December 20, 2010

Abstract

This document describes the first series of tasks defined in robotic environment. These tasks will be used to assess the performance of the goal planning methods developed over the course of the project.

To the description provided here correspond the reference implementations maintained in the MASH platform repository.

Contents

1 Introduction

The main goal of the MASH project is to develop tools for the design of very large and complex learning systems. The performance of this strategy will be assessed on multiple goal-planning tasks, either in a simulation environment, or with a real robotic arm.

We present here the description of the first series of tasks for the robotic environment. The robotic cell is a physical device already described in [SD10], which interacts with the rest of the MASH platform with an extension of the application server protocol, described in [AF10]. Each task of this first series is defined by an environment and a goal. The environment comprises a geometrical configuration, boxes used, background image etc.

This document gives a description of the tasks sufficient for understanding their purpose and the challenges they induce. However, they are fully specified through the open-source implementations stored in the MASH repository, which correspond to this document at the time of submission. Francois, is this correct for robotic platform??: This repository will be made open to the public later in the project.

2 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.

2.1 Task

Task is specified by *goal* and the appropriate *environment*. The environment comprises a geometrical configuration, placements of blocks, etc. Although the set of all possible setups is finite, it is sufficiently large to make the task nontrivial. The infinite variety of tasks is effectively given by the changing illumination of the scene and noise in the cameras and robot motions. A reward derives from that goal, negative when the robot gripper collides with the environment and non-negative otherwise. When the goal is reached, the task is over, and another round can be started.

2.2 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 and measured in lattice units.

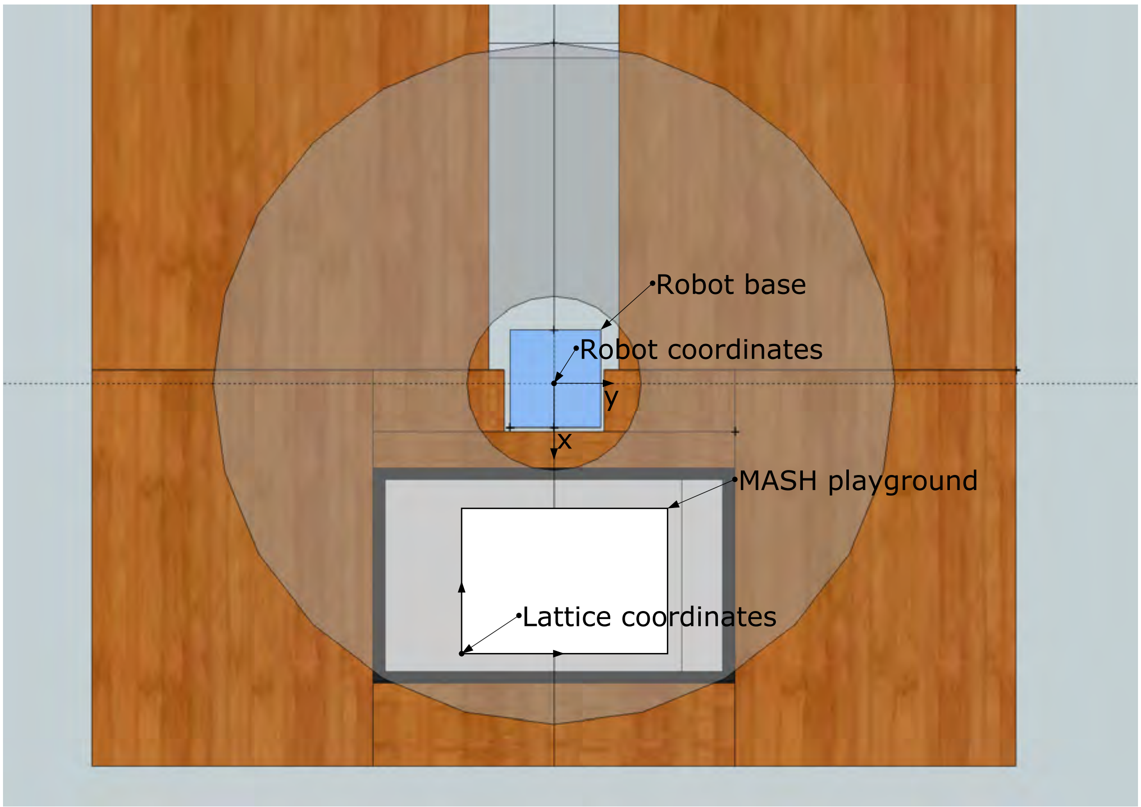

Figure 1: Coordinates of the robot (measured in millimeters) and lattice coordinates (measured in lattice units).

2.3 Environment

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

- **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. The TV image has the width 1920 and height 1080 pixels.
- **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 list of available blocks:

- Cylinders: red, green, yellow, blue.
- Boxes (right rectangular): red, green.
- Cubes: red, green, yellow, blue (two pieces each), unpainted wood (four pieces).

The currently implemented environments in robot application server:

- `STATIC_12x12` - world size is $[12 \times 12]$, environment contains 5 blocks,
- `STATIC_8x6` - dimensions of the world are $[8 \times 6]$, environment contains 4 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.

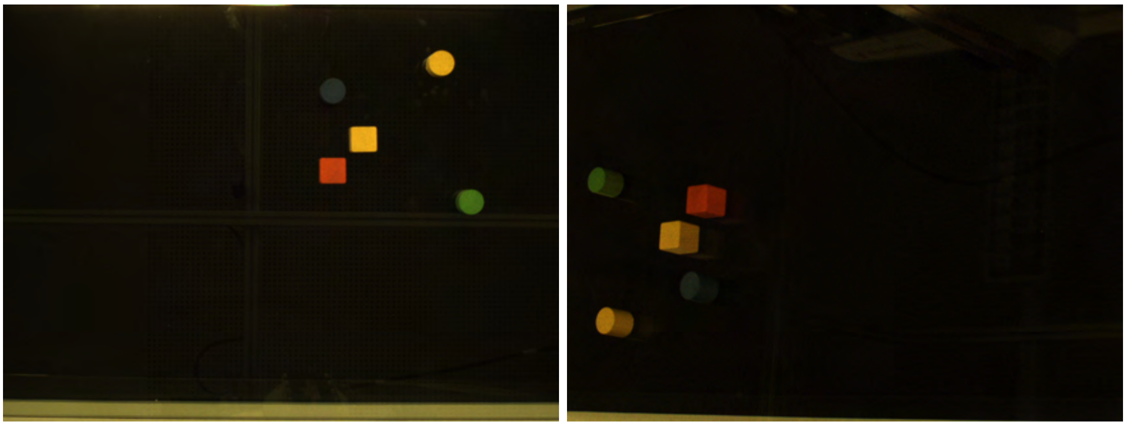

Figure 2: The images of the environment from the top camera (left) and from the oblique camera (right). The images are captured while the robot arm is in store area.

2.4 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.

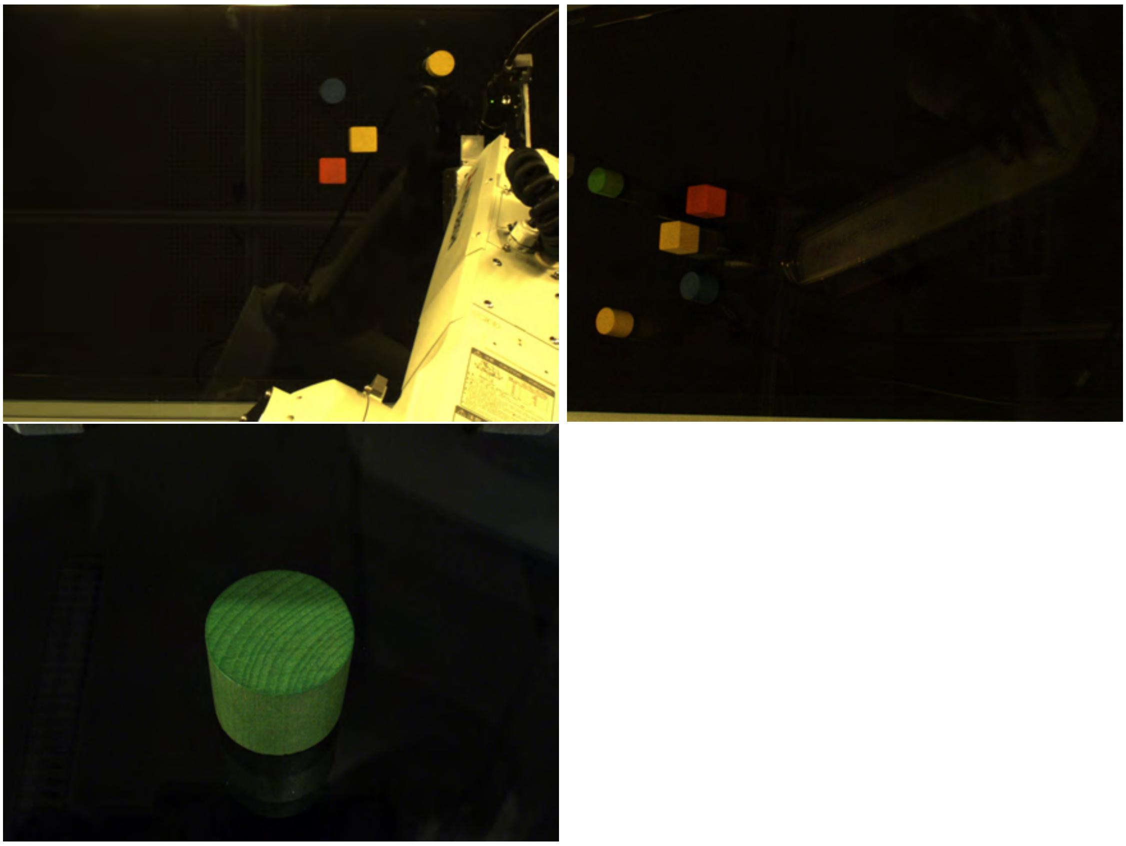

Figure 3: The images of the environment from the top camera (left), from the oblique camera (right), and from the gripper camera (bottom). The images are examples of the images captured during learning/planning phase. The robot arm is at position (0,11), that is just above green cylinder.

2.5 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 20mm 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.

2.6 View

Currently implemented views:

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

References

[AF10] Philip Abbet and Francois Fleuret. First series of simulator tasks. European Project MASH - Massive sets of Heuristics for Machine Learning deliverable, June 2010.

[SD10] Vladimír Smutný and Martin Dubec. Operational robot. European Project MASH - Massive sets of Heuristics for Machine Learning deliverable, September 2010.