

# Indoor scene perception for object detection and manipulation

L.J. Manso, P. Bustos, P. Bachiller and J. Franco  
C aceres Polytechnic School, University of Extremadura,  
C aceres, Extremadura.

lmanso@unex.es

July 4, 2012

## Abstract

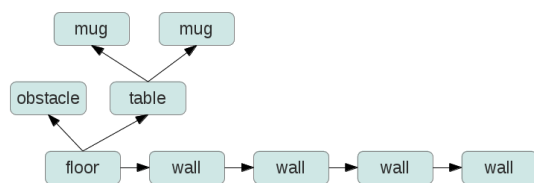
Social robots are designed to interact and share their environments with humans while performing daily activities. They need to build and maintain rich representations of the space and objects around them in order to achieve their goals. In this paper we propose a framework for building model-based representations of the space surrounding the robots and the objects nearby. The approach considers active perception as the phenomena resulting from controlled interactions between different model-fitting algorithms and a grammar-based generative mechanism called “Grammars for Active Perception” (GAP). The production rules of these grammars describe how world models can be built and modified, and are associated with the behaviors needed by the model-fitting algorithms in order to succeed. Such descriptions can be used to compute the required actions to build consistent models of the environment. The resulting behavior seizes the a priori knowledge available to the robot, not only to improve the modeling process, but also to guide exploration and visual attention. The models generated using these grammars are attributed graphs that can contain geometric and other semantic properties.

## 1 Introduction

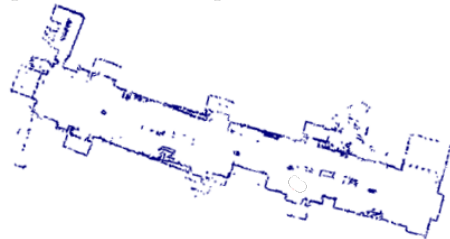
Autonomous robot must be endowed with modeling capabilities in order to operate in human environments. This fact has found deep support during the last years of research in mobile robotics, where localization and mapping have captured much of the interest of the field. The mainstream approach has successfully centered on the mathematical derivation of a solution to the simultaneous localization and mapping problem (SLAM). This solution uses 3D points as the map construction material. Current research has now diversified targeting problems such as the topological organization of large point maps, see Figure 1.b.

In this paper we follow a model-based approach to the spatial modeling problem that assumes there is certain amount of structured knowledge available that can be used to improve the perception process. This knowledge takes the form of simple parameterized geometric primitives (e.g., cuboids or cylinders)

Figure 1: Topological and metric representations of space



(a) Structured high-level model (node attributes are not shown).



(b) Two-dimensional metric map in the right hand side.

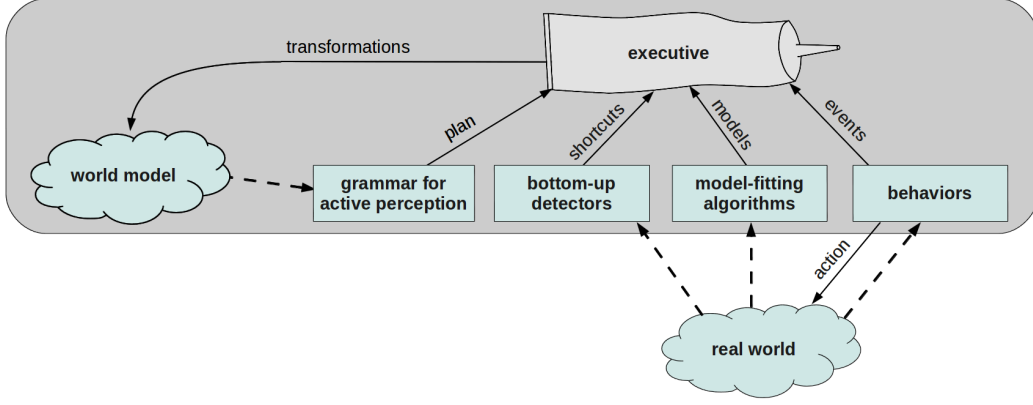


Figure 2: Organization of the perceptive system and data flows. Dashed lines indicate access to the information of the origin of the arrow.

and compositions of them (e.g., tables or mugs) that are fitted to the sensed data through an active perception process. However, the limited field of view of cameras, occlusions, noise or limited sensor resolution introduce uncertainty in the perception process that must be reduced by moving the robot and changing the point of view. This perceptive actions have to be interleaved with task-oriented behaviors to achieve a final goal.

We propose a framework in which perception uses the available knowledge of the structure of the environment and enables robots to appropriately model their surroundings by taking the appropriate actions. When the robot is given a task such as “find the blue mug and bring it here” it unfolds a series of perceptive and motor processes that progressively model the key elements of its environment, constructing a safe way towards the goal. The elements needed to satisfy the goal are perceived in order, following their natural kinematic relationship (e.g., being supported by). Modeling new elements is supported by the context provided by the elements already modeled. When searching for a new object, this context is subtracted from the incoming data, resulting in a crude segmentation of new potential candidates. One candidate is selected, modeled and integrated in the current context until the plan is finished. In the previous example, the room is modeled first, followed by the table and the mug, fitting 3D shapes on them until there is enough information for the arm controller to plan a grabbing behavior.

The main advantages of this type of representation are: a) they can be used before they are complete, since partial models are there from the beginning; b) the model can easily adapt to environment changes once it is built; and c) the abstract nature of the models facilitates human-robot communication. The proposed framework also provides a formal means to achieve active perception through the use of grammars. As depicted in Figure 2 it is composed of five subsystems:

- A set of bottom-up detectors that provide shortcuts among data and models and mitigate the model selection problem.
- A set of model-fitting algorithms.
- A formal grammar that can generate the set of possible perceptive plans. Effective plans are opportunistically selected as time unfolds.
- A set of behaviors for low-level robot control.
- An executive process that controls the interactions among the subsystems.

The rest of the paper describes each of these subsystems in the context of the “mug task” mentioned before. The solution of this task requires the robot to build a representation of its environment, Figure 1.a. The experiment is run in a robotics simulator to provide an initial validation of the idea.





























ID	Rule	behavior
1	 $\Rightarrow$ 	watchFloor
2	 $\Rightarrow$ 	watchFloor
3	 $\Rightarrow$     	alignNextWall
4	 $\Rightarrow$  	alignNextWall
5	 $\Rightarrow$  	alignNextWall
6	 $\Rightarrow$      	lookForObstacles
7	 $\Rightarrow$ 	approachObstacle
8	 $\Rightarrow$  	observeTable

Table 1: Graphical description of the grammar designed for the experiment.

## 2 Grammars for Active Perception

Grammars are sets of rules describing how specific families of structured patterns can be incrementally built. They are generally applied to strings but they can also be applied to graph-like structures such as the one in Figure 1.a. In our approach we use an extension of the concept of grammar, the so called ‘Grammar for Active Perception’ (GAP) [1]. It extends previous graph-grammar formalisms to allow unambiguous descriptions of graph transformations and, by associating behaviors to model transformations, it also provides a means for perceptive planning. GAPs can be used to reason about the valid world model transformations and the actions required to perceive or modify specific world elements. To design a GAP we have to define a working set of symbols. For this example task the symbols used are:

**S** The start symbol.

**N** Represents the normal vector of the ground plane from the point of view of the camera.

**P** Contains the information of the previous symbols and the distance from the camera to the floor, that is, the full plane equation.

**F** Contains the data of the *P* symbol and the yaw angle of the robot respect to one of the four walls. It can be thought as a plane with orientation.

**W, w** For walls at unknown or known distance, respectively.

**O** For obstacles. These symbols contain their position with respect to the walls and its size.

**T** For tables. *T* symbols contain the position, size and height of the table.

**M** For mugs. The robot stores information about their color and size.

The set of rules describing how the world representation can be built is shown in Table 2. Rule 1 is triggered when the robot perceives the normal vector of the floor plane. It substitutes *S* with *N* (which has the normal vector as an attribute). Rule 2 has similar consequences but introduces the height of the camera. Rule 3 is triggered when the room orientation has been successfully modeled and substitutes *P* with *F*, adding new information to it. The rule also includes new *W* symbols for the walls. Rules 4 and 5 substitute *W* by *w* when the wall distances are estimated. Rule 6 is triggered when the robot detects and models an obstacle. Rule 7 is used when an object previously modeled as an obstacle is found to be a table. Rule 8 is used to include new mugs in the model. As can be seen in Table 2, each rule is associated with behaviors that will help the detectors to provide the necessary information to trigger the rules.

As introduced in [1], GAPs have interesting applications. The robot achieves *bottom-up parsing* by monitoring the rules that can be potentially triggered and activating their corresponding behaviors and

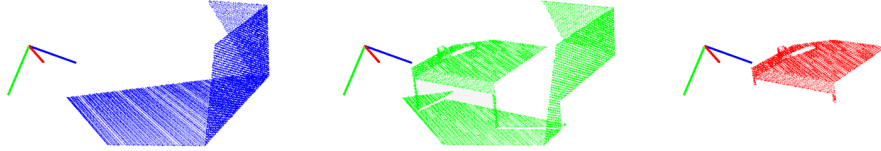


Figure 3: Illustration of the process of subtracting points that are already explained by the world model and that allows the robot to detect obstacles and tables. The leftmost point cloud is generated using the known world model. The one in the center is the cloud directly perceived by the RGBD sensor. Finally, in the right side, the cloud of points that can not be explained by the current model of the environment.

modeling algorithms. This also leads to *context-aware restrictions*, as only valid modeling algorithms are run, only valid transformations are introduced in the model. *Covert perception* is achieved by activating detectors of objects that will not be introduced in the model, but can help detecting other objects. For example, the previous grammar can only insert mugs in tables. Thus, the table modeling process can be triggered by a table detector or be forced by the detection of a mug. *Action selection* is performed by the executive using the Grammar for Active Perception as a planner.

### 3 Model-fitting algorithms and bottom-up detectors

Model-fitting methods are being increasingly used for scene understanding. Model-fitting algorithms have many advantages over discriminative methods but they are also much slower, specially with poor initial information (e.g., when the object to model is small and its location is unknown). To overcome this drawback we use a combination of bottom-up discriminative detectors and top-down generative algorithms. Bottom-up detectors are used as attention-attractors that trigger and initialize model-fitting algorithms. This strategy, known as ‘Data-driven Markov Chain Monte Carlo’ [2], speeds up the fitting process.

As indicated in section 2, the room is not modeled atomically but in successive steps of two (pitch, roll), one (height) and one (wall distance) dimensions respectively –the last one is repeated four times for each of the walls–, which converges considerably faster than a seven-dimensional search. Moreover, it allows the robot to execute behaviors granting appropriate points of view that depend on the element to perceive (i.e., the floor in the two first cases, the walls in the last case). Each search begins with an a priori model of the room and is optimized using particle filters to fit the input data. The a priori values of the height and orientation of the camera with respect to the floor are obtained using the known kinematic structure of the robot. There is no a priori for the wall distances.

Obstacles and tables are different because their number and positions are not known. Thus, we use a discriminative detector in order to hypothesize possible candidates and trigger the corresponding behaviors and model-fitting algorithms needed to perceive tables. The detector uses the already available room model to detect clusters of points from the RGBD sensor that can not be explained by the room model (see Figure 3). When a cluster of points is found, the executive includes an obstacle in the model and executes an approaching behavior. This behavior tries to attain an appropriate point of view to run a model-fitting algorithm which will fix an obstacle or transform it into a table. A similar procedure is used when detecting mugs: the mug detector subtracts the points that can be explained using the current model and leaving only noisy points and those that correspond to mugs. The remaining point cloud is fitted to a mug model.

### 4 Behaviors

Behaviors play an essential role in robot perception. In order to perceive the environment correctly, robots have to attain favourable points of view of the parts they try to model. This must be achieved by adopting the appropriate behaviors. In the experiment that we describe the robot can engage in four behaviors. The first two are used to model the room:

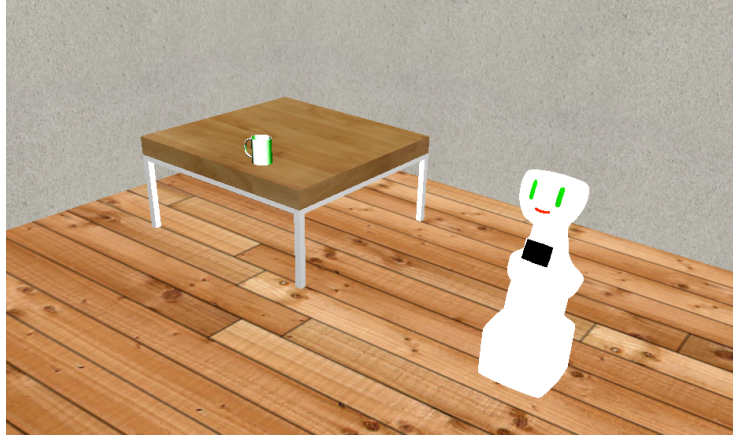


Figure 4: Environment used for the experiment.

- **watchFloor** The robot tries to localise an obstacle-free area of the floor. It is used to model the normal vector to the floor or its distance from the camera.
- **alignNextWall** Used when modeling wall-to-robot distances and when the robot selects the main orientation of the room. The robot navigates and directs its gaze towards the next wall to the right and stands still waiting for the model-fitting algorithm to finish.

The remaining two behaviors are used to model the objects in the room.

- **lookForObstacles** This behavior makes the robot wander in order to enable the obstacle detector to do its job.
- **approachObstacle** Used to attain a favourable view of a specific obstacle and manage to fix the obstacle as it is or transform the obstacle symbol into a table, and to fit mugs to the points not explained by the model.

This small set of behaviors gives the robot the capability to model the room in which it is located, and the obstacles, tables and mugs in it. To model new objects this set should be increased correspondingly as well as the grammar and model-fitting algorithms.

## 5 Experiment

This section describes the steps followed by the robot to achieve the goal “find the blue mug and bring it here”. The experiment is run in a simulated environment in which noise can be added to recreate real world conditions (see Figure 4).

The shortest plan is computed by the grammar engine as explained in [1] and is composed of the following sequence of rules: (1,2,3,4,5,5,5,6,7,8). In order to execute a rule the automata always performs the same tasks: a) activate the behavior and detectors corresponding to the rule; b) activate the necessary model-fitting algorithms when the behavior succeeds; c) trigger the rule when the model is fitted. The sequence of behaviors, detectors, model-fitting algorithms and rules triggered are shown in Table 5. Figure 2 shows the evolution of the model of the room during the experiment. In grey, the cloud of sensor points show the partial view seen by the robot and its relation to the fitting model.

## 6 Conclusions

We have presented a framework for active perception in which the context is modelled hierarchically and used in turn to detect new objects using a subtraction operation. Also, we have presented its application

behavior	Detectors	Model-fitting	Rule number
watchFloor		plane normal	1
watchFloor		plane distance	2
alignNextWall		plane distance	3
watchNextWall		plane distance	4
watchNextWall		plane distance	5
watchNextWall		plane distance	5
watchNextWall		plane distance	5
lookForObstacles	outlierDetector	bounding sphere	6
approachObstacle	outlierDetector	table model	7
observeTable	outlierDetector	mug model	8

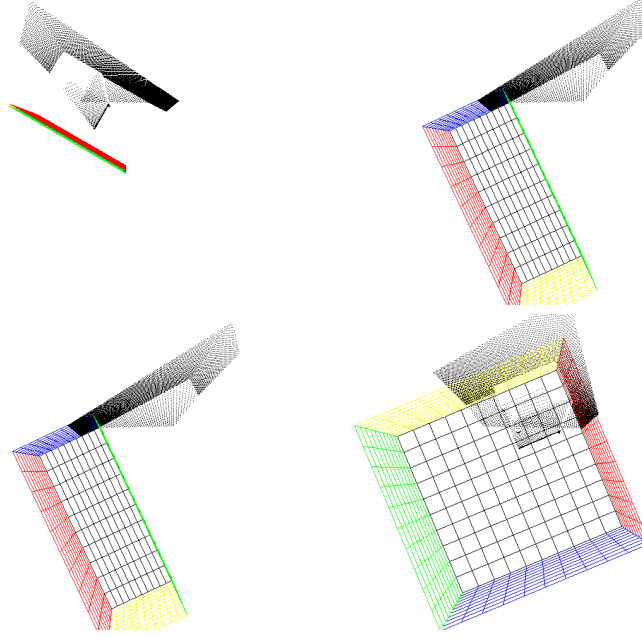


Table 2: Graphical representation of the evolution of the room model during the experiment

in the “mug” robot task where rooms, obstacles, tables and the mugs that tables contain are modelled. The approach has been tested in a simulated robot environment showing encouraging results. We expect to continue this research in cognitive subtraction operations and turn it into a formal theory of active perception.

## References

- [1] Luis J. Manso, Pablo Bustos, Pilar Bachiller and Marco A. Gutierrez. “Graph Grammars for Active Perception”. *Proc. of 12th International Conference on Autonomous Robot Systems and Competitions*. Pp 63-68, 2012.
- [2] Z. Tu and S-C Zhu. Image Segmentation by Data-Driven Markov Chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 24, n. 5, pp. 657-673. 2002.