From Perception to Action and vice versa: a new architecture showing how perception and action can modulate each other simultaneously*

Antonio Jesús Palomino[1], Ángel García-Olaya[2], Fernando Fernández[2] and Juan Pedro Bandera[1]

Abstract— **Artificial vision systems can not process all the information that they receive from the world in real time because it is highly expensive and inefficient in terms of computational cost. However, inspired by biological perception systems, it is possible to develop an artificial attention model able to select only the relevant part of the scene, as human vision does. From the Automated Planning point of view, a relevant area can be seen as an area where the objects involved in the execution of a plan are located. Thus, the planning system should guide the attention model to track relevant objects. But, at the same time, the perceived objects may constrain or provide new information that could suggest the modification of a current plan. Therefore, a plan that is being executed should be adapted or recomputed taking into account actual information perceived from the world. In this work, we introduce an architecture that creates a symbiosis between the planning and the attention modules of a robotic system, linking visual features with high level behaviours. The architecture is based on the interaction of an oversubscription planner, that produces plans constrained by the information perceived from the vision system, and an object-based attention system, able to focus on the relevant objects of the plan being executed.**

I. INTRODUCTION

In biological vision systems, the attention mechanism is responsible for selecting the relevant information from the sensed field of view. In robotics, this ability is specially useful because of the restrictions in computational resources which are necessary to simultaneously perform different vision related tasks [1].

From a deliberative point of view, there are several behaviours to be accomplished that depend on the perception of a specific set of objects. From that definition, we can deduce the effects on deliberative planning: we have partial observability, since the attention model constrains the information that the robot perceives; we have uncertainty, because we can not expect that elements perceived in the past remain as they were in the past (sometimes not even for a small period of time).

In other words, there exists a very close relationship between an attention-driven perception system and a deliberative planner typically included in the reasoning phase of the classical perception-reasoning-action loop. This loop is usually addressed in an unidirectional way: the deliberative layer proposes a set of visual features to be found and the perception system only needs to look for them. But in this work, we focus on the bidirectional perception-planning connection. Specifically, we analyse the application of automated planning in the attention model of the vision module of a robot and we introduce an architecture that permits their mutual interaction.

The connection between perception and action, specially when an artificial attention system is employed, is still an open question. Besides, the problem has been addressed from different points of view because it is a meeting point between Computer Vision and Planning lines of research.

On the one hand, people working on developing attention models have faced the problem by including a task-dependant component in saliency computation. Thus, [2], [3] and [4] add a top-down saliency map able to pop out objects or regions that fit with the current task. However, none of them explicitly defines how to obtain those task-dependant maps (i.e., there is no link with a deliberative model that points out what elements in scene are relevant to the task).

On the other hand, models such as [5] and [6] deal with the classical unidirectional assumption of perception-action loop. They use hierarchical planning and Bayesian approaches respectively to specify the features to be searched in the scene in order to accomplish a specific task (*where and what to look?*). But these models do not take into account the appearance of new objects which could modify the task to be executed because they are bounded to very specific behaviours. Moreover, they do not apply the visual attention concept so they can not use the advantages of focussing only in relevant parts of the whole image.

The proposed approach addresses the perception-planning-action loop in a bidirectional way, coping with the drawbacks of the aforementioned models. On the one hand, we can define different top-down templates depending on the expectations or requirements of a multi-purpose planner. Thus, we can change the relevance of an object depending on its utility for the ongoing task. On the other hand, the introduction of new objects in scene triggering new tasks (e.g. critical or higher priority tasks) is taken into account. Thereby, a continuous adaptation of the plan depending on the perceived objects is allowed.

In summary, the attention model returns only information from relevant areas and only that information can be used to generate an action plan. But planning should also affect the attention module, since the planning system defines what to

*This work has been partially granted by the Spanish Ministerio de Economía y Competitividad (MINECO) projects no. TIN2008-06196, TIN2012-38079-C03-03 and TIN2012-38079-C03-02. It has been also granted by Universidad de Málaga, International Campus of Excellence Andalucía Tech.

[1]A.J. Palomino and J.P. Bandera are with Departamento de Tecnología Electrónica, Universidad de Málaga, Campus de Teatinos, 29071 Málaga, Spain {ajpalomino, jpbandera} at uma.es

[2]A. García-Olaya and F. Fernández are with the Planning and Learning Group, Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911 Leganés, Madrid, Spain {agolaya, ffernand} at inf.uc3m.es

do. Therefore, it must suggest what type of information is relevant or not to the attention module. Such information is derived from the task that the robot must solve and hence, from the objects involved in the execution of such plan.

The underlying idea of this work comes from neurophysiological observations that suggest that particular perceptual characteristics, such as location or shape, engage actions related to those characteristics, such as reaching and prehension networks, and those activating action systems may prime the processing of stimuli defined by the perceptual characteristics related to these actions [7], [8].

The remainder of this paper describes the proposed visual attention and planning systems used and introduces the architecture which connects them in order to close and solve the perception-reasoning-action loop proposed. Finally, the approach is evaluated using a coloured card ordering problem, where the different colours of the cards are the basis of the attention model.

II. The Attention Model

In this section, we introduce an object-based model of visual attention for a social robot which works in a dynamic scenario [1]. Over the past few years, computer vision researchers have been trying to take inspiration from biological visual systems, which are able to filter out the irrelevant information in the scene to focus all its resources in processing only relevant parts. A complete survey about existing artificial attention models can be found in [9] and [10].

The psychological basis to develop artificial visual attention systems are mainly two complementary theories: Treisman's *Feature Integration Theory* [11] and Wolfe's *Guided Search* [12]. The first one suggests that the human vision system detects separable features in parallel in an early step of the attention process (the *pre-attentive* stage, which is totally task-independent) to finally integrate them through a bottom-up process into a single saliency map. Several years later, Wolfe proposed that a top-down component in attention can increase the speed of the process giving more relevance to those parts of the image corresponding to the current task. Furthermore, attention theories introduce another important concept: the *Inhibition of Return*. This mechanism implies that an already attended object should not be selected again until some time later. Otherwise, the most relevant object would be always selected.

The used attention system integrates task-independent bottom-up processing and task-dependent top-down selection. In this model, the units of attention are the so-called *proto-objects* [13], that are defined as units of visual information that can be bounded into a coherent and stable object. On the one hand, the bottom-up component determines the set of proto-objects present in the image, describing them by a set of low-level features that are considered relevant to determine their corresponding saliency values. On the other hand, the top-down component weights the low-level features that characterize each proto-object to obtain a single saliency value depending on the task to perform.

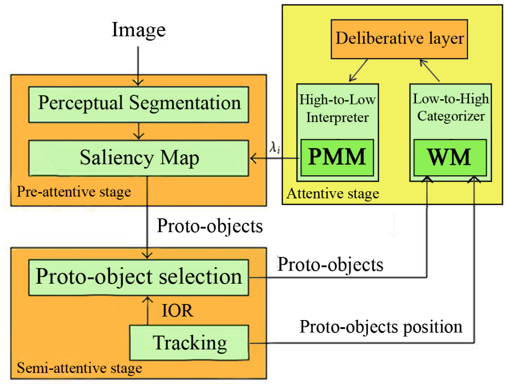


Fig. 1. Overview of the Object-Based Attention Model

An overview of the system is shown in fig. 1. In the *pre-attentive* stage, the different proto-objects present in the image are extracted, using a perceptual segmentation algorithm based on a hierarchical framework [14]. Then, the relevance of each proto-object is computed taking into account different low-level features (concretely, colour contrast, intensity contrast and dominant colour -red, blue, green or yellow-) weighted by a set of parameters (λ_i) stored in a *Perception-Modulation Memory* (PMM). Depending on the value of these parameters, the system is able to modify the influence of each low-level feature in the global saliency computation. The idea of perception-modulation parameters is supported by the biological concept of "attentional sets" proposed by Corbetta et al. [15]. As a result of this stage, a set of proto-objects ordered by their saliency is obtained.

The next stage, the *semi-attentive* stage, deals with the management of the *Working Memory* (WM) and the *Inhibition of Return* (IOR). The WM establishes the maximum number of attended elements that can be maintained at once. It is a short-term memory where the system stores the recently attended objects and it has a reduced capacity, up to 5 elements [16]. Each proto-object in WM is characterized by a set of descriptors: its saliency value, its position in the image, the different low-level features values and a *time-to-live* value which establishes the maximum time that the proto-object can stay in WM. A proto-object's saliency also depends on this last parameter, so the longer an element is kept in WM, the lower its saliency is. A new proto-object gets into the WM if and only if it has bigger saliency than the currently stored elements. If the memory is full, the least salient element is dropped out. Regarding the IOR, a tracker module keeps permanently updated the position of each element in WM, allowing to manage not only moving objects but also camera and robot movements. Thereby, it is avoided to attend an already selected proto-object. If a proto-object is lost, it is also removed from WM.

Both WM and PMM are the interface between early attention stages and the rest of the system, including the deliberative level. This interface incorporates a categorizer which is able to classify the perceived proto-objects into categories corresponding to high-level predicates. Besides, the PMM has been modified to translate high-level instructions into a new set of perception parameters λ_i, so it is allowed to change the way the vision system perceives the world in terms of a high-level decision.

III. The Planning Framework

Since the attention model limits perception, we need to plan with only partial information about the initial state, being able to use *sensing actions* to increase our knowledge about it. This has been usually addressed by using contingent planning [17]. A contingent planning problem is a tuple $P = \{F, A, I, G\}$, where F is the set of literals and fluents, A is the set of actions, $I \subseteq F$ is the initial state and $G \subseteq F$ is the set of goals. Actions in contingent planning include conditional effects, allowing the effects of actions to depend on the real state to which the action is applied. Sensing actions discover the value of a certain previously unknown literal. We assume deterministic actions, though our approach can be extended to non-deterministic ones [18], [19].

Contingent planning aims to find complete plans achieving all the goals by intercalating sensing actions whenever it is needed. This works well when the uncertainty about the initial state is small, but does not scale well in general. Recent approaches do not aim to produce complete plans but only to return at least a valid action to be executed [18], [19]. Generally speaking, they create a belief state by selecting a small subset of the possible initial states and create a plan according to this belief state. Plan is executed until an unexpected observation occurs or the preconditions of the next action to be applied do not hold. In this point, the belief state is updated and a new plan is generated.

We use a similar approach, but starting only from the currently perceived state. Instead of a belief state we have a single initial state containing the perception plus some static known facts. It is very likely that no plan achieving all goals will exist given the limited available information. We model our problem as an oversubscription planning [20] (OSP) one, where the planner is able to return a plan reaching just a subset of all the goals. OSP is a special case of planning with soft goals, where it is assumed that no plan achieving all soft goals exists. Usual causes making impossible to reach all the soft goals are limitation of resources or mutex goals. In our case, the lack of information is what makes some goals unachievable. There are some advantages in solving our problem in this way. First, it results in a simpler model of actions, as no conditional effects are needed. Second, it allows to overcome the problem of non-deterministic sensing actions by not reasoning about them; we just apply a sensing action after each planning cycle. Third, oversubscription planners tend to scale better than contingent ones. This is specially true in real environments with limited perception, where the number of objects for which the state is unknown is quite large and the results of the sensing actions are non predictable.

To solve the oversubscription problem we use the procedure introduced in [21]. First, goals are selected and a new problem is constructed removing all the non-selected goals. Summarizing their approach, relaxed plans are constructed from the initial state to each goal and from every goal to each other. If a relaxed plan achieving a goal is found, the goal is added to the set of possibly achievable goals. Once goals are selected the new problem is solved using any classical planner. If after a certain time no plan is found for the new problem, one of the goals is removed and a new problem is created. If a plan is found, it is executed and the environment is perceived again. If no goal remains, a new perception cycle is initiated.

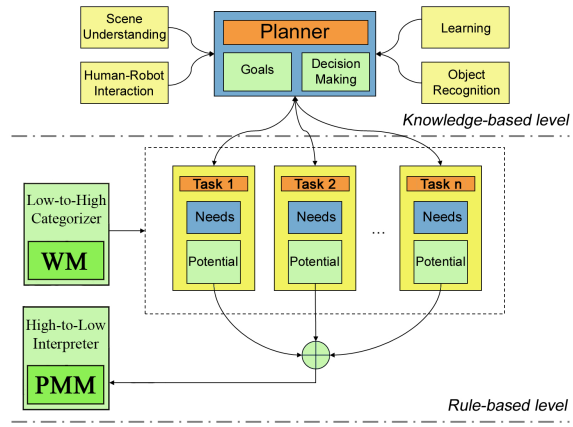


Fig. 2. Two-level architecture connecting planning and attention.

In addition to actions reaching the goals, the domain includes actions to guide the attention model for the next step. Given the current perceived state, the perception module is biased to obtain the information needed for the next planning cycle. This is specially important when no goal can be reached with the current information. The procedure will be detailed in the evaluation section.

IV. The Perception-Planning-Action Loop

In order to define the relationship between the previously described attention model and the oversubscription planner, we introduce a new two-level architecture based on Rasmussen's psychological proposal [22]. In terms of attention systems, the planner implements the top-down part of attention.

As it is shown in fig. 2 and it was aforementioned, the connection with the attention system is made through both Working Memory (WM) and Perception Modulation Memory (PMM). On the one hand, the different tasks that the system has to perform are located in the *Rule-Based level*. In this level, each task has a set of *needs* in form of categories that must be covered (e.g. the task "look for a red card" needs the categories "red things" and "square things"). Depending on the number of satisfied needs due to the elements present in the WM, the influence of the task in the modulation of the perception parameters stored in the PMM will be greater (fully covered) or smaller (weakly satisfied).

On the other hand, the oversubscription planner is placed in the *Knowledge-based level*. In this case, the planner has to

TABLE I

THE PERCEPTION-PLANNING-ACTION LOOP

- Given
  1) The domain description, D
  2) The static part of the problem description, P_s
  3) An oversubscription planner, $OPlanner$
  4) The vector $\vec{\lambda}$ of perception parameters
- Repeat
  – $PerceiveState(\vec{\lambda}) \rightarrow s$
  – $ComposeProblem(P_s, s, \vec{\lambda}) \rightarrow P$
  – $Plan(OPlanner, D, P) \rightarrow Plan$
  – $Execute(Plan)$
  – $UpdateAttentionParameters(\vec{\lambda}, Plan)$
- Until a goal state is achieved

manage the tasks in the system by activating or deleting them and setting their priority in terms of the different achievable goals. Additional high-level information such as scene understanding, human interaction or object recognition, can also be used by the planner in order to accomplish its behaviour.

The double imbrication between planning and perception is easily observable: depending on the categories perceived (needs covered), different tasks are triggered; and vice versa, depending on the dominant tasks, the perception system modifies its parameters so the most relevant objects in the scene can change.

Finally, the concept of different features or categories in early-vision stages triggering different tasks is closely related to the *affordances* proposed by Gibson [23] or the *Reference Features* postulated by Pryor [24]. In both cases, the presence of specific features or categories of objects involves the execution (or the possibility of executing) of certain tasks.

The process described above can be explained through an algorithm (see Table I). This algorithm receives the domain description, D, and the static part of the problem, P_s. With the static part of the problem we mean all the elements of the problem that do not change during the resolution of the problem: the header, the types and objects definition, the goals, and all the static predicates that do not need to be perceived. We also assume an oversubscription planner which will be able to generate partial plans depending on the perceived information and the vector of perception parameters, which constrain perception, and that can be modified by the planning actions.

The algorithm is a repeat loop until all the goals are achieved. In the first step of the loop, the current state, s, is perceived. Such step depends on the perception parameters, $\vec{\lambda}$. With the static part of the problem, P_s, the perceived state, s, and the attention parameters, $\vec{\lambda}$, the new planning problem, P, is composed. Next, the oversubscription planner is called, generating a plan able to reach those goals, if any, that can be solved with the perceived information. The generated plan is then executed and the attention parameter vector $\vec{\lambda}$ is updated according to such plan.

Although the architecture presented here shows a general solution for the problem of linking attention and planning in robotics systems, the way the perception parameters are computed from the solution plan is strongly dependent on each particular robot application. Therefore, there exists a particular *high-to-low* interpreter for each concrete problem.

V. EVALUATION

The proposed architecture is evaluated through a domain compound by a set of coloured cards labelled with letters. The experiment presented here is mainly a proof-of-concept study about the proposed solution to integrate attention and planning.

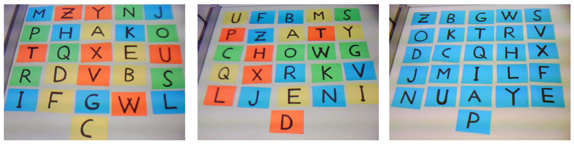
The static predicates of the planning domain, expressed in PDDL (Planning Domain Definition Language) [25], [26], contain the information about the correct ordering of cards `(order A B)` and also about the colour of each card `(colour A yellow)`. The initial state is completed by the perception module by marking as visible the cards that are stored in the WM: `(visible ?x)`. A fourth predicate, `(stack ?x ?y)` stores information about the top and the bottom of the already created stacks. In the initial state, all the cards form a one-card stack. We assume deterministic non-sensing actions, so the information about already stacked cards can be rolled over next iterations. There are only two actions in the planning domain. The first one stacks the visible cards in order if they can be stacked, namely, if two consecutive cards or stacks are visible. As a side effect, every time a card is stacked the salience of its colour is increased, so in the next perception step it is very likely to be perceived again. The second action is applied when no card can be stacked in an iteration. It randomly selects one of the visible cards and increases both the salience of its colour and the next card's colour. To avoid this action to be executed unless nothing else can be done, its cost is 300 times higher than the stack action's one. Since the aim of this work is to show the relationship between perception and planning, the action proposed by the planner is executed by a human.

Fig. 3 shows the different configurations of the domain analysed in the evaluation. The actions in the domain are able to take all the possible advantages from the scenario described in fig. 3.a. In this case, there are 4 sets of letters with the same colour: A-F (yellow), G-M (blue), N-S (green) and T-Z (red). Consecutive letters have the same colour. On the contrary, we define another scenario (fig. 3.b) where there are no consecutive letter with the same colour. Concretely, the four sets are:

- *Yellow* letters: A,E,I,M,Q,U,Y
- *Blue* letters: B,F,J,N,R,V,Z
- *Green* letters: C,G,K,O,S,W
- *Red* letters: D,H,L,P,T,X

Finally, a last configuration where all cards have the same colour is introduced (fig. 3.c) in order to cancel the influence of the planner over the perception system.

With respect of the attention system, a set of 4 feature maps (RED, GRN, BLU and YLW, one per colour) is obtained for each proto-object in the image. If the mean colour of the proto-object is similar to red, green, blue or yellow (measured in terms of HSV-colour space distance), the corresponding map receive a value of 255; otherwise,



(a) (b) (c)

Fig. 3. The three different scenarios for evaluation. In (a), consecutive letters have the same colour. In (b), two consecutive letters never have the same colour. In (c), all letters are blue.

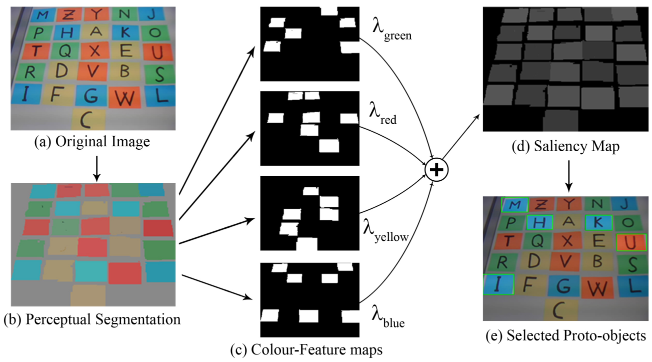


Fig. 4. The image processing.

the value is 0. Then, the global saliency of each proto-object (sal_i) is computed as a linear combination of each feature map, being the weights the perception parameters, λ_i, provided by the PMM

$$\begin{aligned} sal_i &= \lambda_{red} RED + \lambda_{green} GRN + \\ &+ \lambda_{blue} BLU + \lambda_{yellow} YLW \end{aligned}$$

and verifying $\sum_i \lambda_i = 1$.

In this case, we are not using other features also available in the attention model in order to clarify the interpretation of the results. Therefore, all the cards have the same *a priori* saliency, i.e., the influence of bottom-up attention is highly reduced.

Fig. 4 shows the image processing involved in obtaining the saliency map. Once the WM is filled up with the most relevant proto-objects, an OCR (Optical Character Recognition) algorithm is employed to assign a category to each one. Therefore, the planner will receive the corresponding letter as a predicate. It can be seen in fig. 4 that the biggest relevance is given to blue color (λ_{blue} is bigger than the others) so in the saliency map, blue proto-objects are brighter (more salient).

The whole system is evaluated using 4 different approaches:

case 1 We use the scenario from fig. 3.a and the planner provides the solution as aforementioned.

case 2 The configuration from fig. 3.b is employed and the planner tries to follow the same strategy as in case 1.

case 3 Same scenario as in case 2 but the colour proposed by the planner is marked as **less relevant** than the others, instead of the strategy followed before.

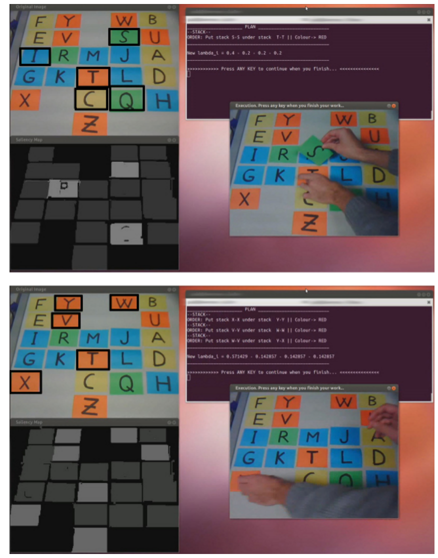


Fig. 5. Two consecutive iterations in the evaluation task

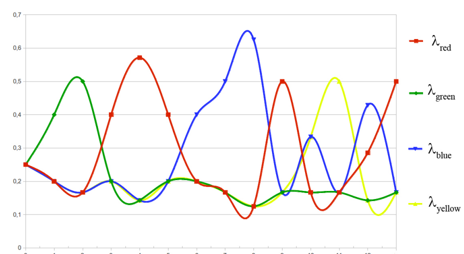


Fig. 6. Evolution of perception parameters, λ_i, over time

case 4 All the cards are blue (fig. 3.c) so the planner is not able to highlight a specific colour as the most relevant.

In fig. 5, two significant consecutive iterations of evaluation (case 1) are shown (the complete video sequence is available in `http://youtu.be/1fZWBJMnzXc`). In the first one (upper in the figure), only letters T and S can be sorted from the objects in the WM (marked with black bounding-boxes). Consequently, the solution plan consists on putting in order these letters and giving more relevance to the card-in-the-top's colour (in this case, red) varying the related λ_i. As a result, in the next iteration, the most salient objects are the red ones, allowing the planner to obtain more solutions at once. The variation over time of the different perception parameters depending on the plan to execute is shown on fig. 6. As depicted in the figure, the system began putting in order the green cards, followed by the red ones, the blue ones and, finally, the remaining loose stacks. When all the letters of the same colour are stacked, that colour loses relevance because the planner guides the attention system to look for the rest of colours.

Table II shows the results of the experiments and fig. 7 represents the number of iterations needed in each case to solve the problem. As it was expected, the best results are obtained for *case 1* because the planner is able to guide the

TABLE II
EXPERIMENTAL RESULTS

Case	Iterations		Total Time		Efficiency	
	μ_i	σ_i	μ_t	σ_t	μ_e	σ_e
1	11,9	2,025	2,864	0,558	2,154	0,346
2	55	12,944	13,929	3,161	0,476	0,127
3	27,3	7,675	7,125	2,122	0,981	0,3
4	43,7	13,158	11,175	3,532	0,629	0,217

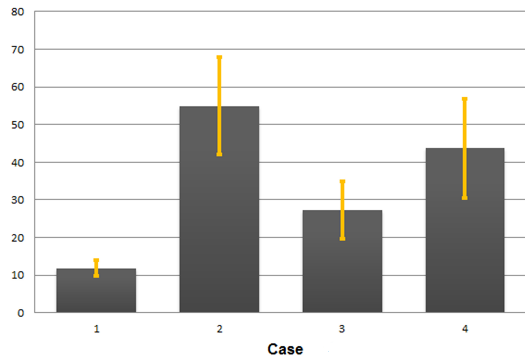


Fig. 7. Number of iterations on average needed in each case to solve the whole problem. The standard deviation is also included in the graph.

perception system in an efficient way. On the contrary, *case 2* produces the worst results due to the fact that the planner guides the vision system in a wrong manner. This solution is even worse than the one in *case 4* where the planner can not highlight any colour to speed up the process. Thus, a bad guiding of the perception system is even worse than the no guiding option. The column corresponding to *case 3* shows that a little modification looking for a more intelligent strategy of the planner is enough to increase significantly the response of the system, reducing the iterations needed to solve the problem almost to the half.

VI. CONCLUSIONS

In this paper, we describe a new architecture for integrating Automated Planning, specifically, oversubscription planning (OSP), with an attention model in the classical perception-planning-action loop of an intelligent system. Attention models allow to process images efficiently, introducing the cost of partial observability, since only relevant areas are processed. Only partial information is processed, thus, only partial plans can be built. In addition, the planner can guide the attention model by modifying the attention parameters and, hence, guiding the perception to focus on the objects required to solve the whole task. The architecture have been successfully evaluated in a real scenario, where we demonstrate that both perception and planning are perfectly integrated.

REFERENCES

[1] A. J. Palomino, R. Marfil, J. P. Bandera, and A. Bandera, "A novel biologically inspired attention mechanism for a social robot," *EURASIP Journal on Advances in Signal Processing*, 2011.

[2] J. Tnnermann, C. Born, and B. Mertsching, "Top-down visual attention with complex templates," in *International Conference on Computer Vision Theory and Applications*, February 2013, pp. 370–377.

[3] V. Navalpakkam and L. Itti, "An integrated model of top-down and bottom-up attention for optimizing detection speed," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 2049–2056.

[4] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search." *Psychological review*, vol. 113, no. 4, p. 766, 2006.

[5] M. Sridharan, J. Wyatt, and R. Dearden, "Planning to see: A hierarchical approach to planning visual actions on a robot using pomdps," *Artificial Intelligence*, vol. 174, no. 11, pp. 704–725, 2010.

[6] E. Gilet, J. Diard, and P. Bessière, "Bayesian action–perception computational model: Interaction of production and recognition of cursive letters," *PloS one*, vol. 6, no. 6, p. e20387, 2011.

[7] T. Collins, T. Heed, and B. Rder, "Visual target selection and motor planning define attentional enhancement at perceptual processing stages," *frontiers on Human Neuroscience*, vol. 4, 2010.

[8] S. Fagioli, B. Hommel, and R. I. Schubotz, "Intentional control of attention: action planning primes action-related stimulus dimensions," *Psychological Research*, vol. 71, pp. 22–29, 2007.

[9] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 1, pp. 185–207, 2013.

[10] S. Frintrop, E. Rome, and H. I. Christensen, "Computational visual attention systems and their cognitive foundations: A survey," *ACM Transactions on Applied Perception (TAP)*, vol. 7, no. 1, p. 6, 2010.

[11] A. Treisman and G. Gelade, "A feature integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[12] J. Wolfe, K. Cave, and S. Franzel, "Guided search: An alternative to the feature integration model for visual search," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 15, no. 3, pp. 419–433, 1989.

[13] R. Rensink, "Seeing, sensing, and scrutinizing," *Vision research*, vol. 40, no. 10-12, pp. 1469–1487, 2000.

[14] R. Marfil, L. Molina-Tanco, J. Rodríguez, and F. Sandoval, "Real-time object tracking using bounded irregular pyramids," *Pattern Recognition Letters*, no. 28, pp. 985–1001, 2007.

[15] M. Corbetta and G. L. Shulman, "Control of goal-directed and stimulus-driven attention in the brain," *Nature reviews neuroscience*, vol. 3, no. 3, pp. 201–215, 2002.

[16] C. Bundesen, T. Habekost, and S. Kyllingsbaek, "A neural theory of visual attention and short-term memory (ntva)," *Neuropsychologia*, vol. 49, no. 6, pp. 1446 – 1457, 2011.

[17] M. Peot and D. Smith, "Conditional nonlinear planning," in *Proceedings of the first international conference on Artificial intelligence planning systems*, San Mateo, California, 1992, pp. 189–197.

[18] G. Shani and R. I. Brafman, "Replanning in domains with partial information and sensing actions," in *IJCAI*, 2011, pp. 2021–2026.

[19] A. Albore, H. Palacios, and H. Geffner, "A translation-based approach to contingent planning," in *IJCAI*, 2009, pp. 1623–1628.

[20] D. Smith, "Choosing objectives in over-subscription planning," in *Proceedings of the 14th Intl. Conf. on Automated Planning & Scheduling*, 2004, pp. 393–401.

[21] A. García-Olaya, T. de la Rosa, and D. Borrajo, "Using the relaxed plan heuristic to select goals in oversubscription planning problems," in *Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, J. Lozano, J. Gómez, and J. Moreno, Eds. Springer Berlin-Heidelberg, 2011, vol. 7023, pp. 183–192.

[22] J. Rasmussen, "Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. SMC-13, no. 3, pp. 257–266, 1983.

[23] J. Gibson, *The ecological Approach to Visual Perception*. Houghton Mifflin (Boston), 1979.

[24] L. Pryor, "Perceptual and functional representations: Bridging the gap," in *Notes of the AISB Workshop on Computational Models of Cognition and Cognitive Functions*. Citeseer, 1994.

[25] D. McDermott, "PDDL the planning domain definition language," Yale Center for Computational Vision and Control, Tech. Rep. CVC TR-98-003/DCS TR-1165, 1998.

[26] A. Gerevini, P. Haslum, D. Long, A. Saetti, and Y. Dimopoulos, "Deterministic planning in the fifth international planning competition: Pddl3 and experimental evaluation of the planners," *Artificial Intelligence*, vol. 173, no. 5, pp. 619–668, 2009.