2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Affordance Graph: A Framework to Encode Perspective Taking and Effort based Affordances for day-to-day Human-Robot Interaction

Amit Kumar Pandey and Rachid Alami

***Abstract*— Analyzing affordances has its root in socio-cognitive development of primates. Knowing what the environment, including other agents, can offer in terms of action capabilities is important for our day-to-day interaction and cooperation. In this paper, we will merge two complementary aspects of affordances: from agent-object perspective, what an agent afford to do with an object, and from agent-agent perspective, what an agent can afford to do for other agent, and present a unified notion of *Affordance Graph*. The graph will encode affordances for a variety of tasks: take, give, pick, put on, put into, show, hide, make accessible, etc. Another novelty will be to incorporate the aspects of effort and perspective-taking in constructing such graph. Hence, the *Affordance Graph* will tell about the action-capabilities of manipulating the objects among the agents and across the places, along with the information about the required level of efforts and the potential places. We will also demonstrate some interesting applications.**

I. Introduction

Ability to analyze affordance is one of the essential aspects of developing complex and flexible socio-cognitive behaviors among primates, including human. In fact, affordance - what something or someone can offer or afford to do - is an important aspect in our day-to-day interaction with the environment and with others.

The pioneer work of Gibson [1], in cognitive psychology, refers affordance as what an object offers, as all action possibilities, independent of the agent's ability to recognize them. Whereas, in Human Computer Interaction (HCI) domain, Norman [2] tightly couples affordances with past knowledge and experience and sees affordance as perceived and actual properties of the things. Affordance could be learnt [3], even through vision based data [4], and by manipulation trials [5], as well as could be used to learn action selection [6]. Moreover, research on *action-specific perception* proposes that people perceive the environment in terms of their abilities to act on it, which in fact help the perceivers to plan future actions, [7]. This equally holds from the perspective of Human-Robot Interaction (HRI), mere existence of an affordance (in its typical notion of action-possibilities) is not sufficient, more important is that the agent should be able to perceive and ground it with their abilities. In [8], the idea to integrate robots and smart environments along with the Gibson's notion of affordances has been presented. The aim is to provide mechanisms by which such system, which the authors term as Peis-Ecology (Ecology of Physically Embedded Intelligent Systems, such as one shown in fig. 1) may gain awareness of the opportunities, which are available in it, and use this awareness to self-configure and to interact with a human user. In this paper, we will extend this idea from the perspective of day-to-day human-robot interactive tasks, by incorporating the notions of *perspective taking* and *effort*, to make the robot *agent and effort based affordance-aware*. We are essentially interested in the aspect of *computing* different types of affordances and integrating them in a unified framework.

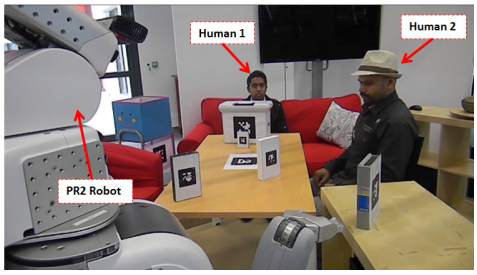


Fig. 1: In a typical human-robot interaction scenario, various types of affordances exist, e.g. some objects offer to be picked, some to put-on something, whereas some agents can afford to give or to show something to someone. External sensors such as Kinect mounted on the ceiling are not shown.

In robotics, affordance has been viewed from different perspectives: agent, observer and environment, [9], studied with respect to object (e.g. for tool use [10]) and location (e.g. for traversability [11]). In [12], linguistic instruction from the human has been grounded based on the affordances of the objects, i.e. the services they provide. In [13], the robot learns self affordance, i.e. action-effect relation of its movement on its own body. In [14], the proposed inter-personal maps relate the information about the robot's own body with that of other robot, and serve for imitating other's sensory-motor affordances. In, [15], we enriched the notion of affordances by *Agent-Agent Affordance* for HRI tasks, based on the reasoning about *what* an agent affords to do for another agent (give, show, hide, make accessible, ...), with *which effort level* and *where*.

On the other hand, perspective taking- what others see, reach, do - is an important aspect of social interaction, [16], [17], [18], [19], [20]. Hence, perceiving the self-centered affordances are not sufficient, the robot should also be able to *perceive the affordances from the other agents' perspectives*. Further, in [21], we have argued that perspective taking

This work has been conducted within the EU SAPHARI project (http://www.saphari.eu/) funded by the E.C. Division FP7-IST under Contract ICT-287513.

Authors are with CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France; Univ de Toulouse, LAAS, F-31400 Toulouse, France
{*akpandey@laas.fr*; *rachid.alami@laas.fr*}

TABLE I: Effort Classes for Visuo-Spatial Abilities

Effort to Reach	Effort to See
No_Effort	*No_Effort*
Arm_Effort	*Head_Effort*
Arm_Torso_Effort	*Head_Torso_Effort*
Whole_Body_Effort	*Whole_Body_Effort*
Displacement_Effort	*Displacement_Effort*
No_Possible_Known_Effort	*No_Possible_Known_Effort*

Effort Level
Minimum: 0
Maximum: 5

from the current state of the agent is not sufficient. In fact analyzing *effort* plays an important role in cooperation. Hence, in [21], we have developed the notion of perspective taking from a set of states attainable by putting a set of efforts [15], e.g. turn, lean, etc. In this paper, it will serve to incorporate the notion of *effort* in *affordance analysis*.

Novelty: The novelty of the paper is to geometrically ground and merge the two complementary aspects of affordance: *Agent-Object* and *Agent-Agent*, further by incorporating the notions of perspective taking and effort analysis. We will present a unified framework as the concept of *Affordance Graph*. This will elevate the robot's awareness about not only the capabilities of itself but also the action potentialities of other agents for a set of basic human robot interactive tasks: *give*, *take*, *pick*, *hide*, *show*, *make accessible*, *put on*, *put into*, etc. The graph encodes 'what' and 'where' aspects for all the agents, which are identified as important components for joint-task, [22]. Hence, the *Affordance Graph* will facilitate an *affordance-aware human robot interaction*, with various potential applications in day-to-day life: achieving joint-task, cooperation, grounding interaction and changes, playing human-interactive games, etc.

In the next section, for continuity, we will briefly present our earlier works on *human-aware effort hierarchy* and *agent-agent affordances* (*Taskability Graph*). In sec. III-A, we introduce the novel concept of *Manipulability Graph*, which encodes agent-object affordances. *Taskability Graph* together with *Manipulability Graph* will be used to derive the novel notion of *Affordance Graph* in sec. III-B. Sec. IV presents the experimental result and analysis on constructing and updating such graphs. In sec. V, we will discuss a couple of implemented potential applications of *Affordance Graph*.

II. Background Work

A. Human-Aware Effort Analysis: Qualifying Effort

For a robot to interact and cooperate with us in complete harmony, it should be able to reason on efforts at the human-understandable level of abstraction. In [21] we have argued that perspective taking analysis only from the current state of the agents is not sufficient. Following this, in [15], we have conceptualized qualitative effort classes, as shown in table I. We have also proposed comparative effort levels, which of course can differ depending upon the requirement and context. In fact these are based on the body parts involved and motivated from human movement and behavioral psychology studies on reach taxonomy, [23], [24], see fig. 2.

Fig. 2: Taxonomy of reach actions: (a) arm-shoulder reach, (b) arm-torso reach, (c) standing reach.

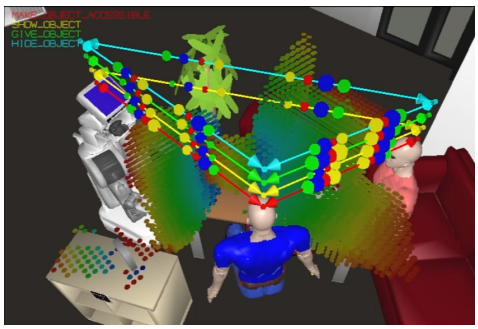


Fig. 3: Constructed *Taskability Graphs* (see [15]), for different tasks, for the actual scenario of fig. 1. Given the criterion of mutual-effort balancing and restricting desired maximum individual effort level to *Arm_Torso_Effort*. Point clouds show potential places and sphere size shows required effort level.

B. Taskability Graph: Encoding Agent-Agent Affordances

As mentioned earlier, in [15] we have introduced the notion of *agent-agent affordance* and present the concept of *Taskability Graph*. The construction of *Taskability Graph* is based on the *Mightability Maps*, [21]. *Mightability* stands for "*Might be Able to...*". In short, *Mightability Analysis* is visuo-spatial perspective taking, not only from the current state but also from a set of different states achievable by the agent. This enables the robot to find effort-based potential places to perform a task between two agents. By using this, *Taskability Graph* is constructed by finding agent-agent affordances among all the agents, [15]. *Taskability Graph* encodes *what* all agents in the environment might be able to do for all other agents, with *which levels of mutual efforts* and at *which places*. *Taskability Graph* corresponding to the actual scenario of fig. 1 is shown in fig. 3. Currently 4 basic tasks are encoded: *Make Accessible*, *Show*, *Give* and *Hide*, as shown by the 4 layers of edges, in bottom up order. Please see [15] for the details.

We represent a *Taskability Graph* for a *task* as a directed graph TG_{task}:

$$TG_{task} = (V(TG), E(TG)) \tag{1}$$

$V(TG)$ is set of agent vertices in the environment:

$$V(TG) = \{v(TG) \,|\, v(TG) \in AG\} \tag{2}$$

where AG is the set of agents in the environment. $E(TG)$ is set of edges between an ordered pair of agents:

$$E(TG) = \{e(TG) \,|\, e(TG) = \langle v_i(TG), v_j(TG), e_{prop} \rangle \wedge v_i(TG) \neq v_j(TG)\} \tag{3}$$

e_{prop} is property of an edge:

$$e^{TG}_{prop} = \big(CS, EC^{TG} = \langle EC^{ag}_{ab} | \forall ag \in \{source\,(e)\,, \\ target\,(e)\} \wedge \forall ab \in RelAb_{ag}\rangle\big) \quad (4)$$

where CS is candidate space where potentially the task could be performed. EC^{TG} is a list of enabling condition. In the current implementation each enabling condition EC^{ag}_{ab} corresponds to the required effort for an ability $ab \in RelAb$, for an agent ag. Currently, we focus on two basic abilities, *to see* and *to reach*, hence, $RelAb = \{see, reach\}$.

Hence, currently an edge of a *Taskability Graph* encodes:

$$e^{TG}_{prop} = \big(places, \big\langle effort^{target\_agent}_{see}, effort^{target\_agent}_{reach} \\ effort^{performing\_agent}_{see}, effort^{performing\_agent}_{reach}\big\rangle\big) \quad (5)$$

III. Methodology

A. Manipulability Graph: Encoding Effort based Agent-Object Affordances

In this section we will introduce the first contribution of this paper: *Manipulability Graph*. This will encode *what* an agent might be able to *do with an object*, and with *which effort level*.

Complementary to the *Taskability Graph*, which encodes *agent-agent affordances*, *Manipulability Graph* represents *agent-object affordances*. Currently, we have implemented four such affordances: *Touch*, *Pick*, *PutOnto* and *PutInto*. We have chosen them because they are the most used basic action primitives within our domain of day-to-day *pick-point-and-place* type tasks. We regard *Pick* as the ability to $See \wedge Reach \wedge Grasp$, whereas *Touch* as the ability to just $See \wedge Reach$, *PutOnto* as the ability to $See \wedge Reach$ the *places on horizontal surface*, *PutInto* as the ability to put something into some container object, i.e. to $See \wedge Reach$ the *places belonging to horizontal open side* of an object.

We are using a dedicated *grasp* planner, built in-house [25], to compute a set of grasps for 3D objects, using the robot's gripper and an anthropomorphic hand. The robot computes and stores the set of such grasps for each new object it encounters. Then based on the environment, the grasps in collision are filtered out. An agent can show reaching behavior to touch, grasp, push, hit, point, take out or put into, etc., hence, the precise definition of reachability of an object depends upon the purpose. So, at first level of finding reachability, we chose to have a rough estimate based on the assumption that if at least one cell belonging to the object is reachable, then that object is said to be *Reachable*. See [21] for our approach to find the reachability of a cell in the workspace. Similar is done for finding visibility of an object. However, using [26] such reachability is further tested for execution if required, based on IK and feasibility of a collision free trajectory. Similarly, a pixel based visibility score is computed, if required, by using [20].

Similar to *Taskability Graph* we represent a *Manipulability Graph* for a *task* as a directed graph MG_{task}:

$$MG_{task} = (V\,(MG)\,, E\,(MG)) \quad (6)$$

$V\,(MG)$ is set of vertices representing entities $ET = AG \cup OBJ$ (OBJ is set of objects in the environment):

$$V\,(MG) = \{v\,(MG)\,| v\,(MG) \in AG \vee v\,(MG) \in OBJ\} \quad (7)$$

$E\,(MG)$ is set of edges between an ordered pair of agent and object:

$$E\,(MG) = \{e\,(MG)\,| e\,(MG) = \langle v_i\,(MG)\,, v_j\,(MG)\,, \\ e^{MG}_{prop}\big\rangle \wedge v_i\,(MG) \in AG \wedge\, v_j\,(MG) \in OBJ\} \quad (8)$$

e^{MG}_{prop} is property of an edge of the *Manipulability Graph*:

$$e^{MG}_{prop} = \big(CS, EC^{MG} = \langle EC^{ag}_{ab} | ag = source\,(e) \\ \wedge\ \forall ab \in RelAb_{ag}\rangle\big) \quad (9)$$

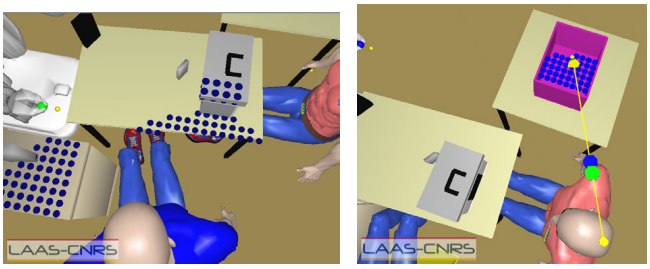
Note that each edge in *Manipulability Graph* contains the list of enabling condition for a single agent (see eq. 10), belonging to the source vertex, because the target vertex belongs to an object. Whereas, each edge of *Taskability Graph* has enabling conditions for two agents (see eq. 5). CS is the candidate search space in which the feasible solution will fall. Depending upon the type of the task, it could be the set of collision free grasp configurations for $Pick$, or it could be the visible and reachable places belonging to a plane for $PutOnto$, to an open side of the container for $PutInto$ or to an object's surface for $Touch$. Hence,

$$e^{MG}_{prop} = (CS \in \{grasp\_configurations, places\} \\ \big\langle effort^{performing\_agent}_{see}, effort^{performing\_agent}_{reach}\big\rangle\big) \quad (10)$$

In the current implementation, if there exist at least one collision free grasp, and the object is reachable and visible with a particular effort level of the agent, then it is said that the agent can afford to *pick* the object with that effort level.

We have equipped the robot with the capability to autonomously find *horizontal supporting facet* and *horizontal open side*, if exist, of any object. For this, it extracts horizontal planes by finding the facet having vertical normal vector, based on the convex hull of the corresponding 3D model of the object. Then such planes are uniformly sampled into cells and a virtual small cube (currently of dimension (*5cm x 5cm x 5cm*)) is placed at each cell. As the cell already belongs to a horizontal surface and is inside the convex hull of the object, therefore, if the placed cube collides with the object, it is assumed to be a cell of support plane. Otherwise, the cell belongs to an open side of the object, from where something could be put inside. With this method the robot could find, which object offers to put something onto it and which offers to put something inside it and which are the places to do these. This avoids explicitly indicating the planner about the supporting objects, such as table, box-top, and the container objects, such as trashbin.

Fig. 4a shows the automatically extracted places where the human in the middle can put something onto with Arm_Effort. Note that the robot not only found the table as the support plane, but also the top of the box. Similarly in fig. 4b, the robot autonomously identified the pink trashbin as a container objects having horizontal open facet. And it



(a) Affordance for *PutOnto* (b) Affordance for *PutInto*

Fig. 4: Agent-Object Affordances

also found the places from where the human on the right can put something inside it.

Manipulability Graph for an agent-object affordance type is constructed by finding least feasible effort for each agent, satisfying the task criteria. Again, the *Mightability Analysis* serves to find the least feasible effort, as it performs visuo-spatial perspective taking from multiple states of the agents.

Fig. 5a shows *Manipulability Graphs*, for *Pick* and *PutInto* affordances. For clarity, we do not superimpose *Touch* and *PutOnto* affordances. Each edge of *Manipulability Graph* shows the agent's least feasible effort to see and reach the objects, fig. 5b. Different maximum desired effort levels can be assigned for different affordances and agents for constructing the graph. To show this, we provided the maximum allowed effort for *Human1* as $Displacement\_Effort$, whereas for PR2 robot and *Human2*, it was $Torso\_Effort$. Hence, the resulted graph shows that *Human1* and *Human2* both can pick *Obj2*, with different effort level. For *Human1*, the planner is able to compute collision free placement positions and configurations around the object, fig. 5c, from where he can reach, see and grasp it. However, the graph shows that *Human1* can put something into the trashbin on his right, whereas *Human2* cannot, because of his more restricted maximum effort level. An interesting side note, relatively smaller green sphere on *Human1-Obj2* edge encodes that *Human1* can see *Obj2* directly, hence more easily than *Human2*, who needs to turn head i.e. to put *Head_Effort*.

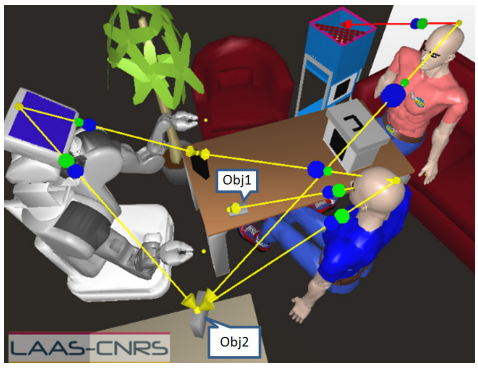
Note that there is no edge from PR2 to *Obj1*. Actually PR2 can reach and see, i.e. touch *Obj1*, however, there exist no collision free grasp configuration for its gripper to pick it, because of the object placement and its gripper size. However, we will see in section IV, that changing *Obj1* position facilitates the *Pick* affordance of PR2 for this object.

B. Affordance Graph

By combining a set of *Taskability Graphs* TGr and a set of *Manipulability Graphs* MGr for a set of affordances, we have developed the concept of *Affordance Graph (AfG)*. Hence, the *Affordance Graph* will tell the action-possibilities of manipulating the objects among the agents and across the places, along with the information about the required level of efforts and the potential spaces.

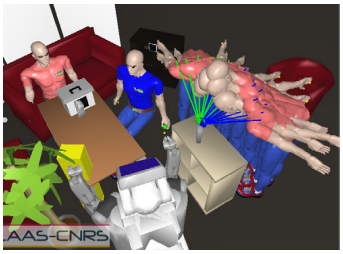
Affordance Graph (*AfG*) is obtained as:

$$AfG = \biguplus_{\forall tg \in TGr} TG_{tg} \uplus \biguplus_{\forall mg \in MGr} MG_{mg} \qquad (11)$$



(a) Manipulability Graph for *Pick* and *PutInto* affordances from the perspective of different agents

(b) Single edge description



(c) *Human1* affordance to pick *Obj2*, requires $Displacement\_Effort$.

Fig. 5: Manipulability Graph, taking into account different maximum allowed effort levels for different agents

$\uplus$ is the operator, which depending upon the affordance type, appropriately merges a *Taskability* or *Manipulability Graph* in *Affordance Graph*. As will be explained below it will create some virtual edges to ensure one of the desired properties to be maintained, that is between any pair of vertices of the *Affordance Graph*, there should be at most one edge. Further, it also assigns proper labels, weights and directions to the edges, which will become evident from the discussion below.

Figure 6 shows the *Affordance graph* of the current scenario. The $\uplus$ operator uses following set of rules for constructing *Affordance Graph*:

(i) Create unique vertices for each agent and each object in the environment.

(ii) For each edge *Et* of *Taskability Graph* from the performing agent *PA* to the target agent *TA*, introduce an intermediate virtual vertex *Vt* and split *Et* into two edges, *E1*, connecting *PA* and *Vt*; and *E2*, connecting *Vt* and *TA*.

(iii) The direction of *E1* and *E2* depends upon the task:
   (a) If the task is to *Give* or *Make-Accessible*, *E1* will be directed inward to *Vt* and *E2* will be directed outward from *Vt* towards *TA*.
   (b) If the task is to *Hide* or *Show*, *E2* will also be directed towards *Vt* from *TA*. This is to incorporate the intention behind such tasks, i.e. the object is not expected to be transported to *TA*, and *E2* is for the purpose of grounding *Vt* to corresponding *TA*.

(iv) Assign meaningful symbolic labels to each of the new edges *E1* and *E2*. For example, if *Et* belongs to *Give*

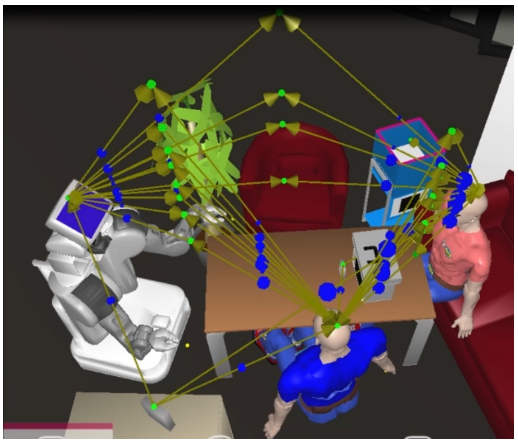


Fig. 6: Constructed *Affordance Graph* of the real world scenario of fig. 1. It encodes effort-based, agent-object and agent-agent affordances for a set of basic HRI tasks.

task, then label *E1* as *To_Give* and label *E2* as *To_Take*; if *Et* belongs to *Make-Accessible* task, then label *E1* as *To_Place* and label *E2* as *To_Pick* and so on.

(v) For each edge *Emt* of *Manipulability Graph* to *Pick* an object, an edge is introduced in the *Affordance Graph* directing from the object to the *PA*.

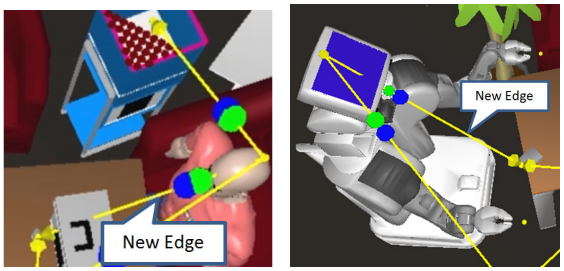
(vi) For each edge *Emp* of *Manipulability Graph* for *PutInto* and *PutOnto* affordances, an edge is introduced in the *Affordance Graph* from *PA* to the objects.

Rule (iii) encodes potential flow of an object between two agents whereas rules (v) and (vi) encode the possible flow of an object to an agent, hence sometime we refer *Affordance Graph* also as *Object Flow Graph*.

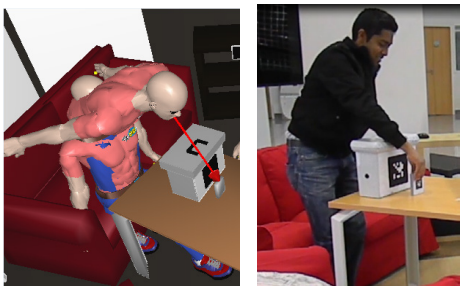
Each edge has a weight depending upon the efforts encoded in the corresponding edge of the parent *Taskability* or *Manipulability Graph*. There could be various criteria to assign such weights. We will discuss below one such choice.

The weights shown by Blue spheres in the *Affordance Graph* of figure 6 have been selected based on the maximum effort of the relevant abilities to see and/or reach. For example, if the edge corresponds to the *Give* task, the highest effort between reach and see, encoded in the *Taskability Graph* will be assigned for both the edges: performing agent, *PA-Vt* (Vt is the virtual vertex as discussed above) and target agent, *Vt-TA*. But, if the task is to *Show*, then for the edge *PA-Vt* still the highest between the reach and see effort will be assigned, however, for the edge *Vt-TA*, the weight is assigned as effort to see. This is because *TA* is not required to reach the object. In fact, the relevant abilities for a task are provided to the system a priori, which could also be learnt for basic HRI tasks as we begin to show in [27].

The novel aspects of *Affordance Graph* are: *(i)* Incorporates *Perspective Taking* and *Effort* with *Affordances* in an unified concept. *(ii)* Provides a graph based framework to query about affordances by simply using existing graph search algorithms. *(iii)* Allows playing with edges, vertices, weights, to guide the graph search. Hence, facilitates incorporating a range of social constraints, desires, preferences, effort criteria, in finding desirable/suitable affordance potentialities. *(iv)* Supports in a range of HRI problems, such as grounding interaction and changes, generating shared cooperative plans, etc., as will be demonstrated in section V.



(a) Human's new affordance edge (b) PR2's new affordance edge



(c) Need whole body effort (d) Actual verification

Fig. 7: Changes in the environment and its effect successfully encoded in the updated *Manipulability Graph*.

IV. Experimental Results and Analysis

We have tested our system on real robot *PR2*.The robot uses *Move3D* [28], an integrated planning and visualization platform. The robot, through various sensors, maintains and updates the 3D world state in real time. For object identification and localization, tags based stereovision system is used. For localizing humans and tracking the whole body, data from *Kinect (Microsoft)* sensor is used.

Fig. 7 shows effect of the environmental changes on the *Manipulability Graph*. We have displaced *Obj1* of fig. 5a behind the box, as shown in fig. 7c. There exists a new edge in the updated corresponding *Manipulability Graph*, as shown in fig. 7a. Earlier there was no edge in the *Manipulability Graph* of fig. 5a because of non-existence of collision free placement of *Human1* around *Obj1*, even with $Displacement\_Effort$. In the changed situation, the robot finds a feasible *Human1 - Obj1 Pick* affordance. Thanks to the *Mightability Analysis*, it finds the least feasible effort to pick $Obj1$ as $Whole\_Body\_Effort$, fig. 7c, and that the agent will be required to stand up and lean forward, which has been also verified by the actual agent, fig. 7d.

Next, we placed *Obj1* at the edge of the table, which facilitated collision free grasp by the PR2 gripper. The updated corresponding *Manipulability Graph* automatically contains a new edge for PR2 - *Obj1 Pick* affordance, fig. 7b.

Table II shows the computation time for different components to obtain the *Affordance Graph* of fig. 1. Note that it is for the first time creation of the graphs, which is acceptable

TABLE II: Computation Time in s

3D grid size: *60 x 60 x 60* cells, each of dimension *5 cm x 5 cm x 5 cm*	
3D grid creation and Initialization (one time process)	*1.6*
Computation of Mightability Analysis (provided 3D grid initialization is done)	*0.446*
Computation of Taskability Graph (provided Mightability Analysis is done)	*1.06*
Computation of Manipulability Graph (provided Mightability Analysis is done	*0.14*
Computation of Affordance Graph (provided Taskability and Manipulabity graphs are calculated)	*0.002*
To obtain the shared plan to clean the table (see fig. 8a (provided Affordance Graph has been created)	*0.01*

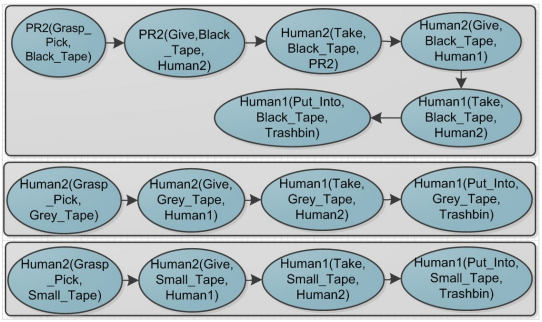
for a typical human-robot interaction scenario. As during the course of interaction or some action, generally a part of the environment changes, hence selectively updating these graphs will be even faster. However, we are aware about the exponential complexity of the system, with significantly increased number of affordances, agents and objects. Therefore, the future work is to interface it with our supervisor system [29], which will decide which part, and how much of these graphs will be updated depending upon the changes, situation and requirements.

V. Potential Applications

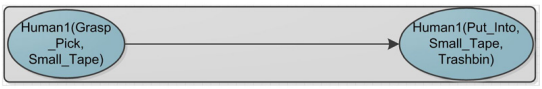
A. Generation of cooperative shared plans

As long as the robot reasons only on the current states of the agents, the complexity as well as the flexibility of cooperative task planning is bounded. If the agent cannot reach an object from the current state, it means to the planner that the agent cannot manipulate that object. But thanks to the *Affordance Graph*, our robot is equipped with agents' rich effort-based multi-state reasoning of action potentialities. It facilitates incorporating effort in cooperative task planning and allows the planner to reason beyond the current state of the agent. Moreover, while querying the graph, by altering the nodes, weights and edges, different types of constraints can be incorporated, such as, *agent busy*, *tired*, *bored*, *back pain*, *neck pain*, *cannot move*, *equally supporting*, etc.

For example, consider the task of cleaning the table by putting all the manipulable objects of fig. 1 into the trashbin. To solve this, the *Affordance Graph* has been used to find the shortest path between each object's node and the node which corresponds to the trashbin. We assume that the agents are tired and don't want to stand up or move. This can be directly incorporated by restricting the agents' maximum desired effort as Arm_Torso_Effort. To reflect this, the edges having higher weights than Arm_Torso_Effort have been assigned an infinite weight, hence avoiding any path through them. The resulted object-wise clean the table shared cooperative plan has been shown in fig. 8a, restricting the individual efforts to Arm_Torso_Effort. Last row of table II shows that once we have the *Affordance Graph*, finding the solution for such tasks is very quick, $0.01s$ to obtain the plan of fig. 8a.



(a)



(b)

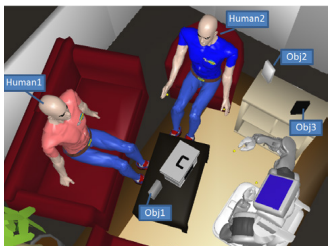
Fig. 8: Generated shared plan for clean the table. (a) With maximum effort level of all the agents as Arm_Torso_Effort. (b) Modified plan for small tape in case *Human1* is allowed to put upto $Whole\_Body\_Effort$.

Next, we have tried the framework by increasing *Human1* willingness to put $Whole\_Body\_Effort$. Then the sub-plan to trash the small tape was changed as shown in fig. 8b.

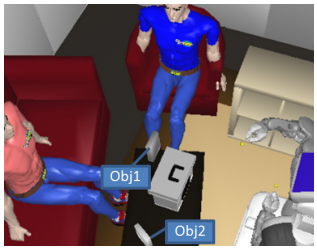
Note that these examples demonstrate that various requirements could be easily incorporate in the presented graph based framework. In principle a higher level robot supervisor system, such as SHARY [29], or the high-level task planner, such as HATP [30], will take such decisions on preferences based on the requirements and adjust the parameters.

B. Grounding Changes, Analyzing Effects and Guessing Potential Actions, Cooperation and Efforts

Based on Affordance graph, a set of hypotheses could be generated about potential agents and actions, which might be responsible for some changes in the environment in the absence of the robot. Consider at a particular instance of time the state of the environment observed by the robot was s_0, as shown in fig. 9a. The robot went away for a moment, meanwhile the humans made some changes in the environment. Now the robot is back and observes the new



(a)



(b)

Fig. 9: (a) Initial environment state, s_0. (b) The changed state, s_1. During the course of changes the PR2 robot was absent. Now, from PR2's perspective, *Obj3* is lost, whereas the positions of *Obj2* and *Obj1* have been changed. The robot will try to guess the lost object's position as well as to ground the changes in terms of agents and actions.

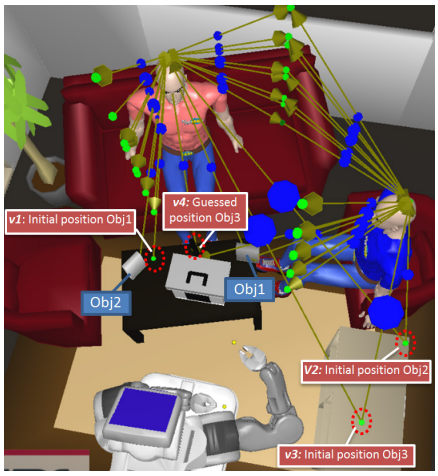


Fig. 10: Modified *Affordance Graph* to ground the changes of fig. 9. For the lost objects, the reasoner also guesses one feasible placement and orientation (in this scenario, vertex *v4*), which might be making the object invisible to the robot.

state of the environment as s_1, fig. 9b. The problem of grounding the changes is: ***given*** $\langle s_0, s_1 \rangle$***, find*** $\langle C, E, A \rangle$, where C is the physical changes, E is the effect of C on abilities and affordances and A is the potential sequence of actions behind such C.

To analyze E on the agents abilities, the reasoner compares the computed *Manipulability Graphs* corresponding to the states s_0 and s_1, and generates a set of comparative and qualifying facts, such as *effort increased to see an object by an agent* and so on. However, we will discuss the more interesting aspects: guessing *lost* objects' positions, and grounding changes to potential agents and actions. *Lost* objects are those, which are no more visible in state s_1 form the robot's current perspective, such as *Obj3* of figure 9a.

First of all, the reasoner adds state s_0's positions of those objects, which are now displaced in state s_1, in a list of dummy object vertices, DV. Then, to guess a position of *lost* object, it uses the *Taskability Graph* for *hide* affordance and finds the agents who might potentially hide the object from the robots as well as the corresponding potential places. Currently, it is explicitly provided to the reasoner that the object could be lying on wooden furniture (table, shelf, in our current scenario). Hence, for the current example, it guessed a possible placement and orientation of the *lost* object $Obj3$ behind the white box, as indicated in figure 10. This guessed position is also inserted in the list of dummy object vertices DV. Then, DV is inserted into the set of vertices belonging to agents (AG) and objects (OBJ), to get $GV = \{AG \cup OBJ \cup DV\}$. Finally, using GV, *Manipulability Graph Taskability Graph* and *Affordance Graph* (AfG_1) are obtained. Hence, this graph merges both the states, s_0 and s_1 as well as the guessed positions (if any) in a single hypothetical state.

As the robot was not responsible for the changes, the reasoner removes the outgoing edges from the robot vertex in (AfG_1). Then for each displaced object Obj_d, a pair of vertices $\langle vs, vg \rangle$ is found from (AfG_1), where $vs \in DV$ and vg correspond to the positions of Obj_d in s_0 and s_1. Now, simply finding a shortest path (AfG_1) for $\langle vs, vg \rangle$, the reasoner can guess about the agent, the action and the effort behind the change for Obj_d position.

In the current example, the set of dummy vertices found by the reasoner is $DV = \{v1, v2, v3, v4\}$, as encircled in red in figure 10. To get possible explanations behind the changes, the set of vertices pairs obtained are $\{\langle v1, Obj1 \rangle, \langle v2, Obj2 \rangle, \langle v3, v4 \rangle\}$. Note that, $\langle v3, v4 \rangle$ corresponds to the *lost* object, *Obj3*. Below we show the partial output of the explanations produced by the reasoner. (Names mapping: $Obj1$: $GREY\_TAPE$, $Obj2$: $WALLE\_TAPE$, $Obj3$: $LOTR\_TAPE$).

== POSSIBLE EXPLANATIONS == ⌊⌊ interpretation ⌋⌋

- LOTR_TAPE *Moved*:
  LOTR_TAPE GRASP_PICK by HUMAN2, GIVE at_a_place, TAKE by HUMAN1, PUT_ONTO at_a_place
  ⌊⌊ *Human2 picked the object and gave it to Human1, then Human1 placed it at its new position, which in fact was* ***guessed*** *by the robot, as it is a lost object.* ⌋⌋
- WALLE_TAPE *Moved*:
  WALLE_TAPE GRASP_PICK by HUMAN2, GIVE at_a_place, TAKE by HUMAN1 PUT_ONTO at_a_place
  ⌊⌊ *Human2 picked the object and gave it to Human1 and then Human1 placed it at its new perceived position* ⌋⌋
- GREY_TAPE *Moved*:
  GREY_TAPE GRASP_PICK by HUMAN1, PUT_ONTO at_a_place
  ⌊⌊ *Human1 picked and placed it at its new position* ⌋⌋

Above result shows the capability of the system to infer the potential cause of changes with a possible explanation. This is based on different assumptions about the agents and their willingness cooperate and put efforts, hence not necessarily be guessing the actual course of actions. Depending upon various factors, such as the criteria on effort used for computing *Taskability* and *Manipulability Graphs*, the criteria for assigning weight to the edges of *Affordance Graph*, and the criteria to find the path in the *Affordance Graph*, the resultant path in the graph could be different and could imply different assumption for guessing the actions. In the current example, as the criteria was effort balancing and the resultant shortest path was minimizing overall effort, hence the explanation in some sense assumes that whenever possible and feasible, agents will cooperate to achieve the changes. However, that is how we also guess, based on some assumptions about the agents, their states and behaviors.

A remark on planning complexity: It is important to discuss about the complexity of such task planning. Such graphs are meant for a given environment state s_i, aiming to provide rich information about various types of affordances in the given situation. Therefore, while finding cooperative shared plan, transition from one vertex to another will make some changes in the environment state and result into a new state s_n. Hence, the graphs computed in s_i might no longer be representing the actual affordances in s_n, and a partial or full re-computation might be needed, before searching for

the next segment of the plan. This might lead to exponential complexity. In our example scenarios, we assume that as the agents' relative positions are not changing significantly, and the objects' are not large enough to significantly affect the *Taskability Graphs* from one state to another. Hence, we relaxed the need of updating these graphs and relying on the plan entirely produced by searching in the single graphs. As such graphs are fast to compute, and updating them will be even faster. It is interesting research challenge to design and develop algorithm, which intelligently updates such graphs (completely or partially), when they are used for planning.

C. Supporting Higher level symbolic task planners

Even with the limiting assumption discussed above, such graphs could provide the high-level task planners, such as [30], with the flexibility to choose from different feasible sub-actions and the associated cost, at different stages of planning. Our research is moving in this direction [31].

D. Grounding interaction, enhancing human robot interaction, proactivity and action recognition

The graph can be used for querying during various human robot verbal communication, grounding referred objects, synthesizing proactive behaviors, etc. Moreover, knowing the affordances possibilities can help in predicting *where* and *what* part of other's action to efficiently or proactively involve in joint-task or cooperation, [22]. We are working in these research directions as well.

VI. Conclusion

We have geometrically grounded *agent-object affordances* from the HRI perspective, and introduced the notion of *Manipulability Graph*. By merging this with the *Taskability Graph*, introduced in our earlier work to encode *agent-agent affordances*, we presented a new kind of graph, *Affordance Graph*, which introduces the aspects of *Perspective Taking* and *Effort* in *Affordance Analysis*. This encodes the action potentialities of the agents for a set of basic human-robot interactive object manipulation tasks. In fact, it elevates the robot's knowledge about effort-based affordances for agents, objects and tasks and will serve as the basis for developing more complex socio-cognitive behaviors. Fast updating and quick querying of the graph, make it feasible for almost real-time human-robot interaction. We have shown its use for generating shared cooperative plans and grounding changes and discussed other applications and pointers of future work.

References

[1] J. J. Gibson, *The Theory of Affordances*. Psychology Press, Sep. 1986, pp. 127–143.

[2] D. Norman, *The psychology of everyday things*. Basic Books, 1988.

[3] E. J. Gibson, "Perceptual learning in development: Some basic concepts," *Ecological Psychology*, vol. 12, no. 4, pp. 295–302, 2000.

[4] H. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *International Journal of Robotics Research*, vol. 32, no. 8, pp. 951–970, 2013.

[5] B. Moldovan, P. Moreno, M. van Otterlo, J. Santos-Victor, and L. De Raedt, "Learning relational affordance models for robots in multi-object manipulation tasks," in *IEEE ICRA*, 2012, pp. 4373–4378.

[6] M. Lopes, F. S. Melo, and L. Montesano, "Affordance-based imitation learning in robots," in *IEEE/RSJ IROS*, 2007, pp. 1015–1021.

[7] J. K. Witt, "Action's effect on perception," *Current Directions in Psychological Science*, vol. 20, no. 3, pp. 201–206, 2011.

[8] A. Saffiotti and M. Broxvall, "Affordances in an ecology of physically embedded intelligent systems," in *Towards affordance-based robot control (2006)*. Springer-Verlag, 2008, pp. 106–121.

[9] E. Şahin, M. Çakmak, M. R. Doğar, E. Uğur, and G. Üçoluk, "To afford or not to afford: A new formalization of affordances toward affordance-based robot control," *Adaptive Behavior*, vol. 15, no. 4, pp. 447–472, Dec. 2007.

[10] A. Stoytchev, "Behavior-grounded representation of tool affordances," in *IEEE ICRA*, April 2005, pp. 3060 – 3065.

[11] E. Ugur, M. R. Dogar, M. Cakmak, and E. Sahin, "Curiosity-driven learning of traversability affordance on a mobile robot," in *IEEE ICDL*, 2007, pp. 13–18.

[12] R. Moratz and T. Tenbrink, "Affordance-based human-robot interaction," in *2006 international conference on Towards affordance-based robot control*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 63–76.

[13] E. Erdemir, M. Wilkes, K. Kawamura, and A. Erdemir, "Learning structural affordances through self-exploration," in *IEEE Ro-Man*, 2012, pp. 865–870.

[14] V. Hafner and F. Kaplan, "Interpersonal maps: How to map affordances for interaction behaviour," in *Towards Affordance-Based Robot Control*, ser. LNCS, E. Rome, J. Hertzberg, and G. Dorffner, Eds. Springer Berlin / Heidelberg, 2008, vol. 4760, pp. 1–15.

[15] A. K. Pandey and R. Alami, "Taskability graph: Towards analyzing effort based agent-agent affordances," in *IEEE Ro-Man*, 2012, pp. 791–796.

[16] J. H. Flavell, B. A. Everett, K. Croft, and E. R. Flavell, "Young children's knowledge about visual perception: Further evidence for the level 1 - level 2 distinction," *Developmental Psychology*, vol. 17, no. 1, pp. 99–103, 1981.

[17] P. Rochat, "Perceived reachability for self and for others by 3 to 5-year old children and adults," *Journal of Experimental Child Psychology*, vol. 59, pp. 317–333, 1995.

[18] C. Breazeal, M. Berlin, A. G. Brooks, J. Gray, and A. L. Thomaz, "Using perspective taking to learn from ambiguous demonstrations," *Robotics and Autonomous Systems*, vol. 54, pp. 385–393, 2006.

[19] J. Trafton, N. Cassimatis, M. Bugajska, D. Brock, F. Mintz, and A. Schultz, "Enabling effective human-robot interaction using perspective-taking in robots," *IEEE Trans. SMC, Part A: Systems and Humans*, vol. 35, no. 4, pp. 460 – 470, 2005.

[20] L. Marin-Urias, E. Sisbot, A. Pandey, R. Tadakuma, and R. Alami, "Towards shared attention through geometric reasoning for human robot interaction," in *Humanoids 2009*, dec. 2009, pp. 331 –336.

[21] A. K. Pandey and R. Alami, "Mightability maps: A perceptual level decisional framework for co-operative and competitive human-robot interaction," in *IROS*, 2010, pp. 5842 – 5848.

[22] N. Sebanz and G. Knoblich, "Prediction in joint action: What, when, and where," *Topics in Cognitive Science*, vol. 1, no. 2, pp. 353–367, 2009.

[23] H. J. Choi and L. S. Mark, "Scaling affordances for human reach actions," *Human Movement Science*, vol. 23, no. 6, pp. 785–806, 2004.

[24] D. L. Gardner, L. S. Mark, J. A. Ward, and H. Edkins, "How do task characteristics affect the transitions between seated and standing reaches?" *Ecological Psychology*, vol. 13, no. 4, pp. 245–274, 2001.

[25] J.-P. Saut and D. Sidobre, "Efficient models for grasp planning with a multi-fingered hand," *Robot. Auton. Syst.*, vol. 60, pp. 347–357, 2012.

[26] M. Gharbi, J. Cortes, and T. Simeon, "A sampling-based path planner for dual-arm manipulation," in *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 2008, pp. 383–388.

[27] A. K. Pandey and R. Alami, "Towards task understanding through multi-state visuo-spatial perspective taking for human-robot interaction," in *IJCAI-ALIHT*, July 2011.

[28] T. Simeon, J.-P. Laumond, and F. Lamiraux, "Move3d: a generic platform for path planning," in *4th Int. Symp. on Assembly and Task Planning*, 2001, pp. 25–30.

[29] A. Clodic, H. Cao, S. Alili, V. Montreuil, R. Alami, and R. Chatila, "Shary: A supervision system adapted to human-robot interaction," vol. 54, pp. 229–238, 2009.

[30] S. Alili, R. Alami, and V. Montreuil, "A task planner for an autonomous social robot," in *Distributed Autonomous Robotic Systems (DARS)*, 2008, pp. 335–344.

[31] L. de Silva, A. K. Pandey, and R. Alami, "An interface for interleaved symbolic-geometric planning and backtracking," in *IEE/RSJ IROS*, 2013.