Real-time Multisensory Affordance-based Control for Adaptive Object Manipulation

Vivian Chu[1], Reymundo A. Gutierrez[2], Sonia Chernova[3], and Andrea L. Thomaz[4]

***Abstract*— We address the challenge of how a robot can adapt its actions to successfully manipulate objects it has not previously encountered. We introduce Real-time Multisensory Affordance-based Control (RMAC), which enables a robot to adapt existing affordance models using multisensory inputs. We show that using the combination of haptic, audio, and visual information with RMAC allows the robot to learn afforance models and adaptively manipulate two very different objects (drawer, lamp), in multiple novel configurations. Offline evaluations and real-time online evaluations show that RMAC allows the robot to accurately open different drawer configurations and turn-on novel lamps with an average accuracy of 75%.**

I. Introduction

A robot operating in unstructured, uncertain human environments cannot rely only on pre-programmed actions, but will need to learn and adapt. Affordances are one representation designed to enable robots to reason about how its actions impact its environment [1], modeling skills as the relationship between actions and effects [2]–[4]. Here, we introduce Real-time Multisensory Affordance-based Control (RMAC), which allows robots to adapt affordance models using multisensory inputs. RMAC makes two main contributions: we learn multimodal sensory models of affordances, and we take a Learning from Demonstration (LfD) approach to connecting a robot's actions with its sensory experiences.

A multisensory approach is crucial for a robot to adapt an action to achieve a specific effect because interactions with the environment are multisensory. While a robot could rely on vision to turn on the lamp (Fig. 1), it should also utilize other modalities (*e.g.* touch or audio) to model the effects that may be critical to achieving an affordance. However, there exist few prior works using multisensory information due to the challenges associated with data of varying time scales and signal types (*i.e.* continuous vs. discrete signals). We show that a robot using RMAC can utilize multisensory input to focus on the key sensor modalities for different affordances and improve the robot's ability to interact with objects.

Using LfD enables the robot to take a previously learned affordance model and transfer it to another object (*e.g.* adapt its existing model for scoop-able to scoop with a bowl rather than the spoon used to teach the affordance) without requiring an expert to provide an objective function by hand or specify the feature space of the feedback controller.

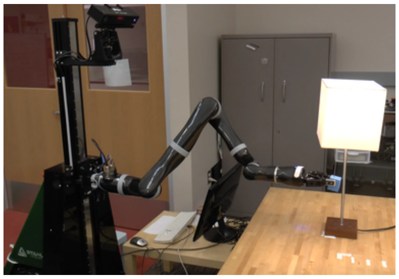


Fig. 1: Robot platform turning on a lamp.

We show that RMAC allows robots to adaptively manipulate two very different objects (drawer, lamp) without requiring an expert to explicitly specify what sensory modalities to focus on. Through both online and offline evaluations, we show that the combination of haptic, audio, and visual input with RMAC allows the robot to open a drawer at 5 different configurations and turn on two never-before-seen lamps with an average accuracy of 75%.

II. Related Work

Multisensory control has been studied for decades. Seminal work [5]–[7] in hybrid control showed manipulation skills split into segments utilize different sensory spaces as feedback (*e.g.* position vs. force). Early methods were limited to structured environments due to requiring an expert to hand-tune trajectories. To execute trajectories in *unstructured* environments, one can leverage LfD [8]. Here we focus on methods most directly related to RMAC.

Multisensory control: Dynamic Movement Primitives (DMPs) are a common method for control [9]. DMPs have been extended to include Force/Torque (F/T) data [10], compliance control [11], online error recovery with F/T sensing [12], and Associated Skill Memories that store sensory inputs [13]. Other approaches include using Hidden Markov Models (HMMs) [14], and learning a kinematic model of an articulated object (doors) using only F/T sensing [15]. Several works demonstrate the value of multimodal data over a single modality [16]–[19]. While prior works merged haptic and visual data in skill learning, they adapt a specific learned trajectory to a specific object. Our work has the ability to generalize to different objects with similar affordances.

Segmentation-based control: Typical approaches focus on how to segment trajectories [20]–[22], build representations of skills using segments [23, 24], and understand skill transitions based on the effects of a segment [19, 25,

*This work is supported in part by NSF IIS 1564080 and ONR N000141612835.

[1] Diligent Robotics, Inc., Austin, Texas vchu@diligentrobots.com

[2] University of Texas at Austin, Austin, Texas Email: ragtz@cs.utexas.edu, athomaz@ece.utexas.edu

[3] Georgia Institute of Technology, Atlanta, Georgia Email: chernova@gatech.edu

26]. Most use modified DMPs to model a segment and classifiers to predict transitions [19, 25, 27, 28]. [25] and [19] use a multisensory approach to detect when and what skills to switch to and are most similar to our work. [25] segments demonstrations with an HMM based algorithm (STARHMM), which detects transitions based on *effects* of actions using visual and haptic data. [19] expand ASMs to include multisensory information about effects of actions including visual, haptic, and auditory data. While these works generalize to different object configurations, they have not been shown to adapt to new objects. Our work focuses on sensory modality saliency in adapting affordance skills.

Control for affordance transfer typically relies on simple PID controllers (*e.g.* pushing across a table [29]–[31], stacking [32]) or rely on hand-provided actions [33, 34]. Most related from Wang et al. [35], which reduces exploration to adapt affordance models. In contrast, we transfer affordances without exploration and use a variety of sensory modalities. Like hybrid control, affordances is a long studied field and these surveys contain a broader view of the field [2]–[4].

III. Approach: Real-time Multisensory Affordance-based Control (RMAC)

We define affordance as an *agent* performing an *action* on the *environment* to produce an *effect*. A robot (agent) performs a set of actions $A = \{a_1, .., a_N\}$ on a set of objects $O = \{o_1, ..., o_M\}$, in order to model the effects that a_i can have on o_j, where $i = \{1, ..., N\}$, $j = \{1, ..., M\}$, and N and M are the number of actions and objects respectively. The robot collects the effect of each object-action (o_j, a_i) pair, making this a supervised affordance learning problem. We introduce RMAC, which builds an affordance-based hybrid controller from human demonstration. This section covers each aspect in detail (see overview in Fig. 2).

Data Collection: We obtain demonstrations with keyframe-based kinesthetic teaching [36], where a person physically guides the robot in performing the skill and specifies specific keyframes (*i.e.* points) along the trajectory the robot should record. The trajectory is executed afterwards by performing a fifth-order spline between the provided keyframes (KFs). We then use human-guided exploration [37], where the robot executes the demonstrated trajectory exactly, while a person modifies the environment to show varied interactions. This requires a person to be present, but could be extended to use self-exploration techniques [38]. Note: RMAC does not depend on keyframe-based demonstrations; only the pose (position $\vec{r}$ and orientation $\vec{q}$) of the end-effector (EEF) and multisensory traces of the skill execution.

Segmentation: While recent work in LfD for trajectory learning has had success with automatic segmentation [21, 25, 39], they require careful hand-tuning. We take a different approach and use the KFs from demonstration to segment the trajectories. People are goal-oriented [40, 41] and KFs likely provide meaningful subgoals. For example, to turn on the lamp, the KFs provided are the start, approach, grasp point, pulling point, release point, retract point, and end. Fig. 6 shows the sensory space correlates well to the KF changes.

We refer to each segment as a subskill segment, and generate two sets: D_A represents the ***A***ctual location the trajectory should be split based on the given KF and D_E represents ***E***xtended segments that are slightly longer (0.5 seconds[1]) than D_A (shown in Fig. 3). D_E captures the sensory input of what the robot should expect when it has successfully completed the current subskill segment and moves to the next. In this work, we use KFs to segment, however RMAC can work with any segmentation algorithm.

Affordance Switching Matrix: After identifying each subskill segment, we generate a "switching matrix" for each skill that represents the high-level action (*i.e.* control mode) to move the robot, similar to matrices in traditional hybrid control methods [6]. The matrix indicates the constrained modalities and open degrees of freedom for each segment. We have two control modes (pose and sensory) where the robot's action depends only on (1) the EEF pose ($\vec{r}, \vec{q}$) and (2) the direct feedback of the real-time sensory inputs.

For each affordance skill, we represent the control modes as a single MxN switching matrix (S) where M is the number of modes in the controller and N the number of segments. Traditionally, M represents the different constraints in Cartesian and sensory space. In this work, we simplify these constraints and assume in pose mode all Cartesian directions (*i.e.* the exact vector (x,y,z)) matter, and in sensory mode all sensory input is important to each subskill segment. While RMAC could still be used without this simplification, we use the sensory model to automatically capture the importance of each direction/modality.

$$S = \begin{bmatrix} \textit{pose}(1) & \textit{pose}(2) & \cdots & \textit{pose}(n) \\ \textit{sensory}(1) & \textit{sensory}(2) & \cdots & \textit{sensory}(n) \end{bmatrix} \quad (1)$$

An example of S can be found in Equation 1. Each column of S represents a subskill segment and each row represents the control mode (*i.e.* S_{ij} where $i = \{1, ..., M\}$ and $j = \{1, ..., N\}$). For each S_{ij}, we assign a binary value (0/1) to represent if that channel is constrained during that segment. For example, if $S_{1,j}$ of the matrix in Equation 1 were $[1, 1, 0]$, the controller would use pose control for the first two segments and sensory for the third. In this work, an expert provides the control mode to use for each subskill segment. In the future, a switching matrix could be learned by computing the variance through each subskill segment.

Similar to prior work [19], we assume the execution sequence of subskill segments is pre-defined and the system will either naturally progress through each subskill segment, or stop if something has occurred that cannot be adapted.

Subskill Segment Modeling: Once we have each subskill segment, we create an action model and a sensory model of each subskill. Prior work often represents the action model of a subskill segment using DMPs [19, 20, 25], and then learn a high-level policy that dictates what DMPs to execute based on sensory models. These sensory models typically either

[1] the average amount of time sensory effects occur based on prior literature

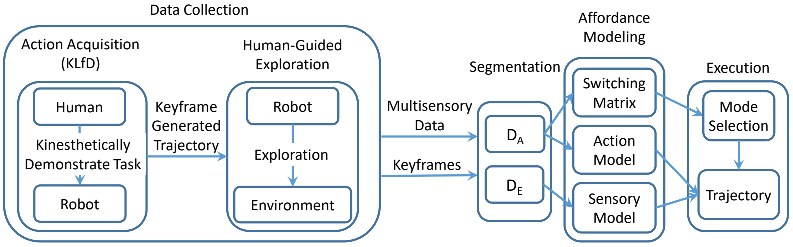


Fig. 2: Real-time Multisensory Affordance-based Control (RMAC): collect multisensory data using human-guided exploration. Affordance model: a switching matrix, an action model, and a sensory model. Execution performed in real-time.

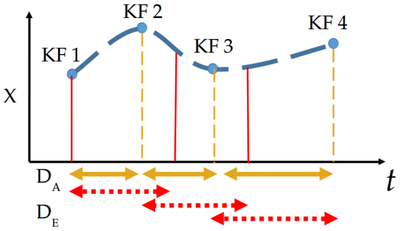


Fig. 3: Example D_A and D_E given a trajectory and keyframes (KFs). X is a single data stream over time.

utilized time and use HMMs [25] or discretized the effect space and use Support Vector Machines (SVMs) [19, 26]. While DMPs can adapt to slight perturbations, the end goal (*i.e.* EEF $\vec{r}$ and $\vec{q}$) must be clearly defined. To address this, [19] built a library of DMPs and selected the correct DMP to execute at each instance by tracking the DMP's trajectory and comparing it to the expected sensory traces.

We take a different approach and represent the action model of each subskill segment as a velocity vector v_n. v_n is generated from D_A and $v_n = \frac{1}{T}\sum_{t=1}^{T} q_t^n - q_{t-1}^n$, where T is the total number of time steps in the trajectory q^n and q_t^n is the pose of the trajectory for subskill segment n at time t. This breaks down each subskill segment into incremental time steps of a subskill (*e.g.* pulling on a handle can be viewed as a sequence of small motions away from the handle until a large force is felt). Representing the trajectory as a velocity, increases the adaptability of the motion. We rely on the effect space to determine if the robot has succeeded. However, this representation adds a challenge that DMPs avoid: while DMPs give a clear ending position to the robot, RMAC needs to determine when to stop the motion.

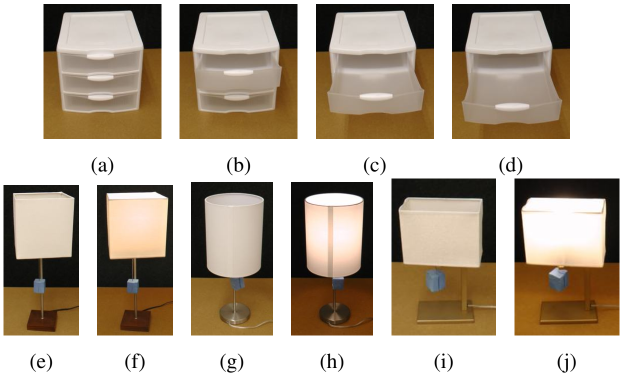
To model the sensory space (*e.g.* forces, sounds, visual change in the scene), similar to [18], we use left-to-right HMMs. We train the HMMs using the D_E segments. The parameters of an n-state HMM, (A,B,π), are estimated using Expectation Maximization (EM) where A is the transition probability distribution (nxn), B the emission probability distributions (nx1), and π the initial state probability vector (nx1). The observation space, O, is modeled with a continuous multivariate Gaussian distribution. The exact state space can be found later in Section V. We use HMMs' hidden-states to track where within the subskill segment the robot is currently executing as well as model the likelihood of experiencing the different sensory inputs in each state. This allows us to integrate time into the model whereas SVM-based approaches do not [19, 26]. To determine when a subskill segment is finished, we track the current hidden-state of the left-to-right HMM. If the robot reaches the final state of the HMM, we conclude it has completed this subskill segment. While not in the scope of this work, these HMMs also allow us to determine when the robot has failed by tracking the likelihoods of an anomaly similar to [18]. Once the robot detects that it has completed this subskill segment, it moves directly to the next segment. Although we specify the exact sequence of segments, this could easily be replaced with a high-level policy similar to [25] and [42].

Execution for Adaptively Interacting with Objects: Once we have built a switching matrix, action models, and sensory models, the robot executes the following steps: (1) If in pose mode, the user gives the segment a specific pose that the EEF must reach. (2) If in pose mode, the robot generates a trajectory computed using the relative pose of the object and the EEF. After pose execution, the segment's HMM determines if we are ready to go to the next segment. (3) If in sensory mode, the robot executes the velocity, v_n, and collects sensory feedback at each time step. After each step, the robot stays in the current sensory mode segment until we reach the final state of the HMM. When in pose mode, we plan a trajectory through the demonstrated keyframes using Rapidly-exploring Random Trees (RRTs) [43] after the EEF pose is converted into joint space using TRAC-IK [44]. The robot then executes the trajectory on the object (Fig. 5).

IV. Robot and Experimental Setup

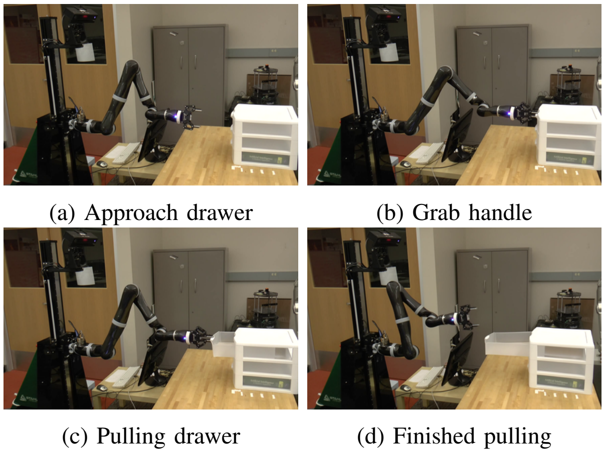
We evaluate RMAC in three experiments, using the robot platform (Fig. 1), with one Kinova Jaco2 7 DOF arm and a Robotiq pinch gripper. The arm can be physically moved in a gravity-compensated mode. The robot has a Microsoft Kinect v2 RGB-D sensor mounted to a pan/tilt unit. We record: gravity compensated wrench at the wrist from the Jaco2 internal forward computed kinematics, visual and audio data from the Kinect2, and the gripper width from the gripper.

To evaluate adaptation of previously learned affordance models, we choose two skills that vary in difficulty. The first examines adaptation to changes to a previously learned object (Case 1). This situation tests RMAC's ability to adapt the robot's trajectory without explicitly tracking the state (*i.e.* how far open the drawer is). For Case 1 (Fig. 4), the robot opens the drawer in five configurations, varying in 2 inch intervals (*i.e.* 1in, 3in, 5in, 7in), systematically showing how RMAC performs under environment changes. The second situation (Case 2) evaluates RMAC's ability to learn the effects of an action and transfer this to a different object with a similar effect. We also look at the impact of sensory modalities for affordance transfer and show that RMAC performs better with multisensory input. For Case 2 (Fig. 4), we use 3 different lamps with varying pull chains.


(a) (b) (c) (d)

(e) (f) (g) (h) (i) (j)

Fig. 4: Selected experiment configurations. Drawer states: (a-d) Closed, 3in, 7in, fully open. Lamp states: (e-j) Original Lamp off/on, New Lamp 1 off/on, New Lamp 2 off/on


(a) Approach drawer

(b) Grab handle

(c) Pulling drawer

(d) Finished pulling

Fig. 5: Robot executing trajectory on the drawer

We chose these objects to show transfer of an existing affordance model to a novel object without requiring the robot to re-explore the object [35]. Specifically, we want to show that the robot can transfer the knowledge of the effects that it is seeking (*e.g.* light change, forces felt, etc.) to a different object that also has these effects.

V. Offline Validation: Adapting Learned Affordance Models to Changed and New Objects

We conduct a series of offline experiments to evaluate RMAC's ability to (1) adapt an affordance model to changing environments without explicitly requiring hand-tuning a closed-looped feedback controller on the state of the object, (2) select what modalities to focus on during adaptation without requiring an expert to provide this information beforehand, and (3) examine the most informative modalities for each skill and relating this to user provided information.

A. Data Collection and Multisensory Features

TABLE I: Sensor Data

Sensor	Modality	Resolution	Features
JACO2	Haptic	100 *Hz*	Raw Forces (F_x,F_y,F_z)
JACO2	Haptic	100 *Hz*	Raw Torques (T_x,T_y,T_z)
Robotiq	Haptic	100 *Hz*	Raw Gripper Width (G_w)
Kinect2	Audio	44.1 *kHz*	Audio Power/Energy (A_e)
Kinect2	Visual	7 *Hz*	PC Color (V_{RGBA})
Kinect2	Visual	7 *Hz*	PC Volume (V_{vol})

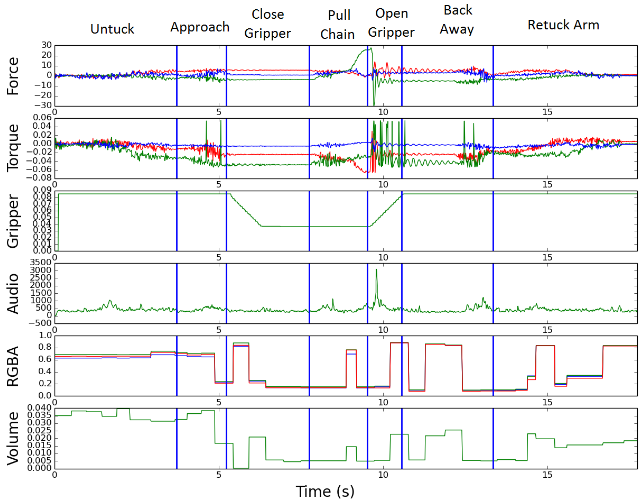


Fig. 6: Features from one interaction with the lamp. The different sensory channels are displayed with vertical lines that indicate the location of the segments of the set D_A

We collected 50 interactions of both opening a fully-closed drawer (Fig. 4a) and of turning on a single lamp (Fig. 4e) using the method described in Section III. We collect data from the sources shown in Table I. From each data source we compute several features that are used to train the sensory model. For haptic data, the robot collects the gravity compensated F/Ts at the EEF and the gripper width and the resulting features are the raw forces (F_x,F_y,F_z), raw torques (T_x,T_y,T_z) and raw gripper width (G_w). The raw audio data is recorded at 44.1 *kHz*. We compute root-mean-square (RMS) of the energy of the Short-time Fourier Transform (STFT) of the audio signal. The specific parameters used to generated the feature (A_e) are frame length: 2048 and hop window: 512. We use the python audio library librosa [45] to compute the audio feature. We compute two visual features from the point clouds: the average color (RGBA) (V_{RGBA}) and the volume of the object (V_{vol}). We use [46] to segment the object from the table. To align the different data sources, we up- or down-sample the data to 100 *Hz*. Fig. 6 shows the computed features from each of the sensory channels and the vertical lines for the location of the KF-segmented version of the trajectory of the lamp. The frames can be viewed semantically as: (1) untuck the arm (2) approach the chain (3) close the gripper (4) pull down on the chain (5) open the gripper (6) back away from the lamp (7) retuck arm. The segmentation for the drawer is omitted due to space. These features are typically used with multisensory data, but future work will look into automatically generating them [18].

B. Training Sensory Models

To test the importance of each sensory modality, we build 7 different sensory models for every combination of the three sensory inputs (*i.e.* visual, haptic, audio). For each sensory model, we change the observation space. The different combinations and feature spaces for O are split by modality (*i.e.* haptic, visual, audio, haptic+visual, haptic+audio, audio+visual, haptic+visual+audio). To train each HMM, we used the successful interactions from the collected runs (lamp: 29, drawer: 32). We select the best number of states (2-15 states inclusive) for the HMMs by performing 5-fold cross validation (CV). When scoring the HMMs during CV, we do not use the log-likelihood (unlike during EM when training a single HMM). Instead, we use the distance away from the true segment switching point. The smaller the value (*i.e.* closer to stopping at the correct location), the better the score. We normalize each observation space by subtracting the mean and scaling the features to have unit variance.

C. Test Set

We do not collect test data using the demonstration as we do in Section III. We use the real-time execution controller described in Section III. This results in data that (a) simulates what the robot will experience when performing execution online and (b) allows us to collect the data past the actual stopping point. We modify the real-time execution during a sensory feedback subskill segment to ignore any sensory feedback and keep executing its velocity vector, v_n, to a specific stopping point significantly past when the robot should have detected the subskill segment has succeeded.

For each object and configuration, we collect 5 test interactions. This results in 35 drawer tests (5 * 7 configurations) and 15 lamp test (5 * 3 lamps). With these, we can compare the importance of each modality for adaptation. The goal is to determine if RMAC can automatically select what modalities are most salient when given all sensory inputs without requiring an expert to provide this information beforehand. By comparing all combinations, we can examine what modalities tend to contribute the best feedback to the task.

D. Results

Case 1 - Drawer: Fig. 7 shows RMAC's predicted stopping point with different sensory inputs. The figure shows one test run for the closed drawer configuration. Each subplot of the figure contains the data source (*e.g.* haptic, haptic and visual, etc.) as well as two vertical bars of differing colors (red and blue). The blue vertical bar indicates where the robot should have stopped (*i.e.* the drawer fully opened: hand-labeled by one of the authors). The red vertical bar indicates where the robot would have stopped using RMAC. The closer the red bar is to the blue bar, the more accurate the controller is at stopping when the drawer is fully open.

For this particular test run, audio information does not help the robot decide if it has opened the drawer. While visual information is helpful, it is not as informative as haptic information. Furthermore, the combination of visual and haptic feedback provides the greatest contribution to the stopping accuracy. This intuitively makes sense for opening drawers: the forces and visual feedback change when pulling open a drawer.Finally, we can see that the combination of haptic, visual, and audio data performs on par with haptic and visual. This shows that the algorithm can automatically determine what modalities matter, without any external sources indicating importance.

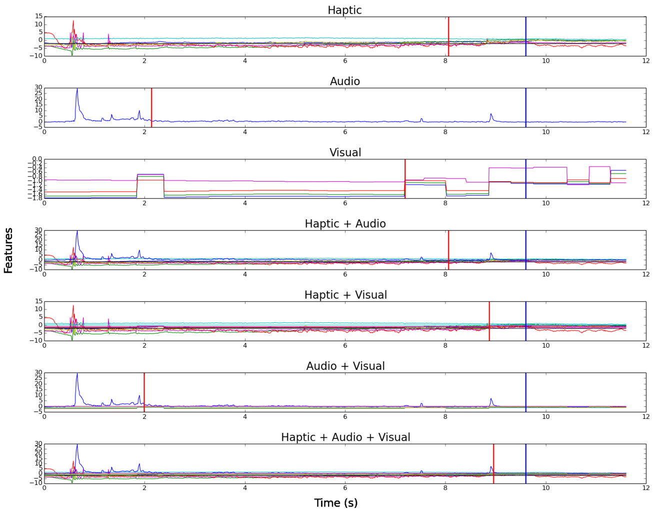


Fig. 7: Drawer fully closed test case. Red bars: predicted stopping point. Blue bars: true stopping point. Haptic+Audio+Visual model is on par with Haptic+Visual.

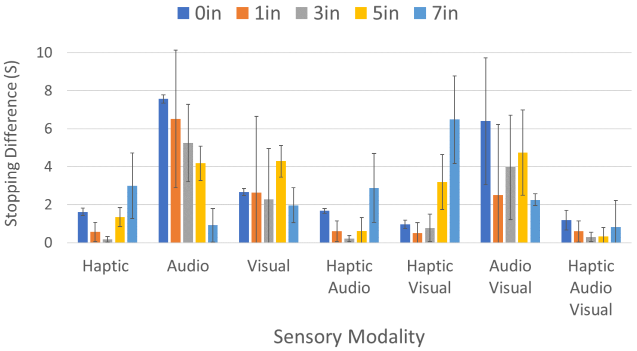


Fig. 8: Avg and STD of absolute difference in stopping times for different drawer configurations across the 7 combinations of the three modalities.

More holistically, Fig. 8 shows that the average differences in time between the ground truth stopping point and estimated stopping point for all five configurations, with smaller distance values being more favorable. It is interesting to note that while the 7 inch drawer configuration is the most difficult to detect because there is only a short amount of time the robot is pulling before it stops, the model built from all modalities does equally as well as under the fully closed and 7 inches open conditions. In addition, the haptic-audio-visual model provides a significant improvement over the haptic-visual model for the 5 inches and 7 inches open conditions. Note, this adaptation occurs without the robot ever being trained on any configuration other than fully closed. Overall, results show that (1) the robot can adapt to different configurations without an expert modeling the exact state of the object and (2) RMAC can select the salient modalities for adaptation without being specified in advance.

TABLE II: Accuracy (%) of Turning on Lamps

Lamp	H	A	V	H,A	H,V	A,V	H,A,V
Original	80	20	20	80	60	20	60
New 1	0	20	20	0	60	0	20
New 2	80	20	20	0	60	0	100

Case 2 - Lamps: Now we evaluate a situation where the object changed. While the drawer interactions all used the same drawer, the lamp interactions require transferring the affordance model to several new lamps. These lamps look similar, but they differ in shape and size of shade and length of pull chain. Further, the sensory readings for this affordance are more difficult to detect than drawer opening because the sensory data can either be discrete or continuous. With the drawers, all of the modalities are continuous, which are easier to model; discrete signals are short so the models have only a short time window to capture any change and look similar to noise. The lamp has two discrete changes to model (*i.e.* audio and haptic) when the lamp switch clicks. The only continuous signal is the visual change in light. Furthermore, the lamp will not turn on if the algorithm stops before the desired location and stopping late risks tipping the lamp.

Table II shows the robot's overall accuracy in turning on the lamps across all modalities. A trial is classified as a success if RMAC places the stopping point at or after the ground truth (provided by the authors). If RMAC places the stopping point before ground truth or does not predict a stopping point, the trial is considered a failure. Note, this might be an overestimate of success because all late predictions are considered successful despite the possibility of tipping the lamp. This was required because the exact moment the lamp could tip was not recorded in the data. This metric is used in place of stopping distance due to the discrete nature of success. Overall, the models perform well but not perfectly at predicting when to stop pulling. Similar to the drawer, the full sensory model (visual, haptic, audio) performs on par with the models that use a smaller subset of modalities (*e.g.* only haptic and visual). Overall, both experiments show that the importance of using multisensory information and the ability of RMAC to select the salient sensory modalities for a skill without any expert guidance.

VI. Online Validation: Adapting Learned Affordance Models in Real-time

In a final experiment, we validate offline results with an online robot implementation of RMAC. We use the controller (Section III) used to collect the offline test (Section V-C). For this evaluation, we connect the data streams to a real-time feature extractor and connect it to the sensory models trained as described in Section V-B. Specifically, we load the trained HMMs using all modalities (haptic+visual+audio), and during the sensory subskill segment playback, determine whether the robot has completed the particular subskill segment or if it should continue executing at velocity v_n.

For the drawer, we execute 5 trials (i.e. closed, 1in, 3in, 5in, 7in) where the robot executes the real-time controller to determine when to stop pulling on the drawer. We measure two things: (1) how far the drawer opened (fully open: 9 inches) and (2) how far the robot dragged the drawer. (1) tells us if the robot stopped too early (*i.e.* if the drawer opened less than 9 inches) and (2) tells us if the robot stopped too late (*i.e.* robot dragged the drawer set across the table). The mean values across the 5 trials can be seen in Table III. As the configuration gets more difficult (*i.e.* the controller has less time to stop), the distance the robot pulls the drawer increases. However, compared to no adaptation, the distance pulled would be much greater (*e.g.* 0.25in vs. 9in).

TABLE III: Online Drawer Using Full Multisensory Model

Config.	Avg. Movement	Open Size	Non-Adapt Movement
0 inch	0.05 in	8.3 in	2 in
1 inch	0.08 in	8.36 in	3 in
3 inch	0.04 in	8.48 in	5 in
5 inch	0.16 in	9.0 in	7 in
7 inch	0.25 in	8.8 in	9 in

The robot executed the controller 10 times for each online lamp evaluation. Accuracy on the Original Lamp, New Lamp 1, and New Lamp 2 were 70%, 60%, and 90% respectively. For the Original Lamp and New Lamp 2, online results were similar to offline. Interestingly, New Lamp 1 performs better than the offline results suggested, likely due to the slight time delay between the real-time signals and the controller. In the offline evaluation, we measured the exact moment the algorithm chooses to stop. However, in the real-time controller, there can exist a slight delay.

The online evaluations show that the controllers using the full multisensory model can be executed in real-time with results similar to those found offline. Furthermore, this evaluation provides a sense of scale to the offline evaluations. While some test configurations did not perform perfectly (*e.g.* the absolute difference in expected time and ground truth were greater than 0), this does not translate into large errors in real-time. In particular, we can see that the robot opens the drawer fully in all cases (*i.e.* between 8.3 and 9 inches) with only a few instances where the robot pulled slightly too long. In these situations, the drawer moves no greater than 0.25 inches. For the lamp evaluation, the difference in expected and ground truth stopping points does not significantly impact the overall success rate.

VII. Conclusion and Discussion

We introduced RMAC, a novel approach to learning and executing affordances. We show that affordances can be adapted in situations where the object or its state has changed and it occurs without an expert specifying an objective function or identifying sensory feature space to focus on. We show in both offline and online experiments that using multisensory input improves the quality of skill adaptation. The evaluations show that the combination of using haptic, audio, and visual information with RMAC allows the robot to open a drawer at 5 different configurations and turn on two never-before-seen lamps. Real-time online evaluations verify offline results showing RMAC allows a robot to accurately open different drawer configurations and turn on novel lamps.

References

[1] J. J. Gibson, "The concept of affordances," *Perceiving, acting, and knowing*, pp. 67–82, 1977.

[2] L. Jamone, E. Ugur, A. Cangelosi, L. Fadiga, A. Bernardino, J. Piater, and J. Santos-Victor, "Affordances in psychology, neuroscience and robotics: a survey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[3] H. Min, C. Yi, R. Luo, J. Zhu, and S. Bi, "Affordance research in developmental robotics: A survey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 4, pp. 237–255, Dec 2016.

[4] P. Zech, S. Haller, S. R. Lakani, B. Ridge, E. Ugur, and J. Piater, "Computational models of affordance in robotics: a taxonomy and systematic classification," *Adaptive Behavior*, vol. 25, no. 5, pp. 235–271, 2017.

[5] M. T. Mason, "Compliance and force control for computer controlled manipulators," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 11, no. 6, pp. 418–432, June 1981.

[6] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 1981.

[7] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 1, pp. 43–53, February 1987.

[8] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469 – 483, 2009.

[9] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 2, 2002, pp. 1398–1403.

[10] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, "Online movement adaptation based on previous sensor experiences," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, Sept 2011, pp. 365–371.

[11] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.

[12] F. J. Abu-Dakka, B. Nemec, J. A. Jrgensen, T. R. Savarimuthu, N. Krger, and A. Ude, "Adaptation of manipulation skills in physical contact with the environment to reference force profiles," *Autonomous Robots*, vol. 39, no. 2, pp. 199–217, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10514-015-9435-2

[13] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal, "Towards associative skill memories," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, Nov 2012, pp. 309–315.

[14] L. Rozo, P. Jimenez, and C. Torras, "A robot learning from demonstration framework to perform force-based manipulation tasks," *Intelligent Service Robotics*, vol. 6, no. 1, pp. 33–51, 2013.

[15] A. Jain and C. C. Kemp, "Improving robot manipulation with data-driven object-centric models of everyday forces," *Autonomous Robots*, vol. 35, no. 2-3, pp. 143–159, Oct. 2013.

[16] S. Wieland, D. Gonzalez-Aguirre, N. Vahrenkamp, T. Asfour, and R. Dillmann, "Combining force and visual feedback for physical interaction tasks in humanoid robots," in *2009 9th IEEE-RAS International Conference on Humanoid Robots*, Dec 2009, pp. 439–446.

[17] J. Sinapov, C. Schenck, K. Staley, V. Sukhoy, and A. Stoytchev, "Grounding semantic categories in behavioral interactions: Experiments with 100 objects," *Robotics and Autonomous Systems*, vol. 62, no. 5, pp. 632 – 645, 2014, special Issue Semantic Perception, Mapping and Exploration. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092188901200190X

[18] D. Park, Z. Erickson, T. Bhattacharjee, and C. C. Kemp, "Multimodal execution monitoring for anomaly detection during robot manipulation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 407–414.

[19] D. Kappler, P. Pastor, M. Kalakrishnan, M. Wuthrich, and S. Schaal, "Data-driven online decision making for autonomous manipulation," in *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[20] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A. G. Barto, "Learning grounded finite-state representations from unstructured demonstrations," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 131–157, 2015.

[21] S. Niekum, S. Osentoski, C. G. Atkeson, and A. G. Barto, "Online bayesian changepoint detection for articulated motion models," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 1468–1475.

[22] N. Figueroa and A. Billard, "Learning complex manipulation tasks from heterogeneous and unstructured demonstrations," In Proceedings of Workshop on Synergies between Learning and Interaction. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017.

[23] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot learning from demonstration by constructing skill trees," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 360–375, 2012.

[24] S. Manschitz, J. Kober, M. Gienger, and J. Peters, "Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations," *Robotics and Autonomous Systems*, vol. 74, Part A, pp. 97 – 107, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889015001438

[25] O. Kroemer, C. Daniel, G. Neumann, H. van Hoof, and J. Peters, "Towards learning hierarchical skills for multi-phase manipulation tasks," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 1503–1510.

[26] Z. Su, O. Kroemer, G. E. Loeb, G. S. Sukhatme, and S. Schaal, *Learning to Switch Between Sensorimotor Primitives Using Multimodal Haptic Signals*. Cham: Springer International Publishing, 2016, pp. 170–182. [Online]. Available: https://doi.org/10.1007/978-3-319-43488-9_16

[27] L. Righetti, M. Kalakrishnan, P. Pastor, J. Binney, J. Kelly, R. Voorhies, G. Sukhatme, and S. Schaal, "An autonomous manipulation system based on force control and optimization," *Autonomous Robots*, vol. 36, no. 1-2, pp. 11–30, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10514-013-9365-9

[28] Y. Chebotar, O. Kroemer, and J. Peters, "Learning robot tactile sensing for object manipulation," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept 2014, pp. 3368–3375.

[29] A. Stoytchev, "Behavior-grounded representation of tool affordances," in *International Conference on Robotics and Automation (ICRA)*, April 2005, pp. 3060–3065.

[30] T. Hermans, F. Li, J. M. Rehg, and A. F. Bobick, "Learning stable pushing locations," in *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, Aug 2013, pp. 1–7.

[31] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning about objects through action - initial steps towards artificial cognition," in *International Conference on Robotics and Automation (ICRA)*, Sept 2003, pp. 3140–3145.

[32] E. Ugur and J. Piater, "Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2627–2633.

[33] D. Katz, A. Venkatraman, M. Kazemi, J. A. D. Bagnell, and A. T. Stentz, "Perceiving, learning, and exploiting object affordances for autonomous pile manipulation," in *RSS Berlin*, June 2013.

[34] N. Krger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wrgtter, A. Ude, T. Asfour, D. Kraft, D. Omren, A. Agostini, and R. Dillmann, "Objectaction complexes: Grounded abstractions of sensorymotor processes," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 740 – 757, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889011000935

[35] C. Wang, K. V. Hindriks, and R. Babuska, "Effective transfer learning of affordances for household robots," in *4th International Conference on Development and Learning and on Epigenetic Robotics*, Oct 2014, pp. 469–475.

[36] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz, "Keyframe-based learning from demonstration," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 343–355, Nov 2012.

[37] V. Chu, B. Akgun, and A. L. Thomaz, "Learning haptic affordances from demonstration and human-guided exploration," in *2016 IEEE Haptics Symposium (HAPTICS)*, April 2016, pp. 119–125.

[38] V. Chu, T. Fitzgerald, and A. L. Thomaz, "Learning object affordances by leveraging the combination of human-guidance and self-exploration," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2016, pp. 221–228.

[39] V. Chu, R. A. Gutierrez, S. Chernova, and A. L. Thomaz, "The role of multisensory data for automatic segmentation of manipulation skills,"

in *RSS Workshop on (Empirically) Data-Driven Robotic Manipulation*, 2017.

[40] G. Csibra, "Teleological and referential understanding of action in infancy," *Philosophical Transactions: Biological Sciences*, vol. 358, no. 1431, pp. 447–458, 2003.

[41] A. N. Meltzoff and J. Decety, "What imitation tells us about social cognition: a rapprochement between developmental psychology and cognitive neuroscience," *Philosophical Transactions of The Royal Society B: Biological Sciences*, vol. 358, pp. 491–500, 2003.

[42] B. Akgun and A. Thomaz, "Simultaneously learning actions and goals from demonstration," *Autonomous Robots*, vol. 40, no. 2, pp. 211–227, Feb 2016. [Online]. Available: https://doi.org/10.1007/s10514-015-9448-x

[43] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, 2000, pp. 995–1001 vol.2.

[44] P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," in *Proceedings of the IEEE RAS Humanoids Conference*, Seoul, Korea, November 2015.

[45] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python." *14th python in science conference*, pp. 18–25, 2015.

[46] A. J. B. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient organized point cloud segmentation with connected components," 2013.