

Error Detection and Surprise in Stochastic Robot Actions

Li Yang Ku, Dirk Ruiken, Erik Learned-Miller and Roderic Grupen

Abstract—For an autonomous robot to accomplish tasks when the outcome of actions is non-deterministic it is often necessary to detect and correct errors. In this work we introduce a general framework that stores fine-grained event transitions so that failures can be detected and handled early in a task. These failures are then recovered through two different approaches based on whether the error is “surprising” to the robot or not. Surprise transitions are used to create new models that capture observations previously not in the model. We demonstrate how the framework is capable of handling uncertainties encountered by a robot in “pick-and-place” tasks on the uBot-6 mobile manipulator using both visual and haptic sensor feedback.

I. INTRODUCTION

Building robots that can handle uncertainty is necessary for autonomous robots to accomplish tasks in a non-deterministic environment. However, the vast majority of failures either evade detection completely or are detected only after a high level action fails to reach the target state. This approach makes robots inefficient and can lead to catastrophic failure if the robot continues to execute a plan when the actual state is quite different from what the robot expected. In this work, plans are monitored during the execution of low-level actions to create a fine-grained observer for error detection. This allows the robot to compare observations to stored models frequently and to handle unexpected outcomes immediately after they are detected.

In [24], environment, sensors, robots, models, and computation are considered as the five factors that give rise to uncertainty in robotic applications. Instead of categorizing uncertainty based on the cause, we focus on how it is perceived. Uncertainties a robot may encounter are classified into two categories: 1) random transitions that are within the model domain but may still be unlikely given transition probabilities, and, 2) “surprise” transitions that lead to outcomes that are not represented in the model domain. An example of a random transition is the outcome of rolling a loaded die that should come up 6 with high probability, but instead comes up 1. The robot will not be able to predict the exact outcome, but by having a model that captures low probability outcomes it is capable of handling each of them accordingly. An example of a surprise transition is if the outcome of rolling a 6 sided die is 13 while the robot’s die model has possible outcomes from 1 to 6. In [3], Casti describes that “surprise can arise only as a consequence of models that are unfaithful to nature.” Therefore we define a “surprise” transition as when the robot observations are not

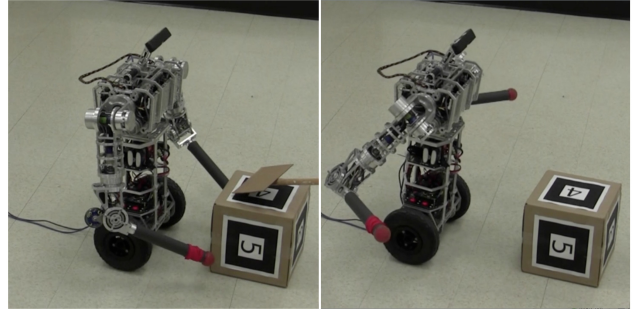


Fig. 1. The uBot-6 mobile manipulator performing a “what’s up?” gesture to convey that it is surprised after an unexpected event.

explainable by any model in the robot memory. During the surprise transition a new model of the environment that can explain this surprising outcome is learned. Failures detected by the fine-grained observer are then recovered differently based on this classification.

In this work, we use the interaction with objects as an example task with such uncertainties. We model each object using a directed multigraph where each node corresponds to a stored observation and each edge corresponds to a stored action. We call a stored observation an aspect and this object model an aspect transition graph (ATG) model [13]. An ATG model captures how actions change observations. We discuss how an ATG model is capable of handling uncertainties encountered by a robot in “pick-and-place” tasks. This framework is tested on the balancing mobile manipulator uBot-6 [22]. We demonstrate that the framework can handle random transitions and recover from surprise transitions at fine-grained time scales.

II. RELATED WORK

In the work done by Rodriguez *et al.* [21] a classifier is used to predict whether a grasp is a successful grasp for completing a task based on haptic feedback. If it is classified as a failed grasp the robot aborts and retries. Our approach is based on a similar concept but instead of running a classifier at a specific step we present a general framework that constantly checks if the observation is within expectations. In research done by Donald [4], a theory for error detection and recovery strategies based on geometry and physical reasoning are introduced. In our work, error detection and recovery is based on an observation-based model of the environment.

Baldi introduced a computational theory of surprise where surprise is defined by the relative entropy between the prior and the posterior distribution of an observer [1]. This formula

Li Yang Ku, Dirk Ruiken, Erik Learned-Miller and Roderic Grupen are with the College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst, MA 01003, USA {lku, ruiken, elm, grupen}@cs.umass.edu

is shown to be consistent with what attracts human gaze in natural video stimuli [10]. However, this theory of surprise would identify informative robot actions that reduces the entropy over models significantly as surprising. In this work we introduce a simpler definition that agrees better with intuition.

Intrinsic motivators for behavior have long been studied in psychology. Hull introduced the concept of “drives” such as hunger, pain, sex, or escape in human behavior, in terms of deficits that the organism wishes to reduce to achieve homeostatic equilibrium [9]. Later, researchers extended the Hullian theory by introducing drives for manipulation [7], and for exploration [17]. Berlyne proposed a number of other intrinsically motivating factors such as novelty, habituation, curiosity, surprise, challenge, and incongruity [2]. In our work, we use surprise as an intrinsic motivator to learn new models of the environment.

Our work also has many connections to prior work on *affordances* defined as “the opportunities for action provided by a particular object or environment.” [5] Affordances are associative and can be used to infer control actions from observations. Our object models are based on this interactionist view of perception and action that focuses on learning relationships between objects and actions specific to the robot.

Aspect graphs were first introduced to represent shape in the field of computer vision [12], [6]. An aspect graph contains distinctive views of an object captured from a viewing sphere centered on the object. The aspect transition graph (ATG) introduced in this paper is an extension of this concept. In addition to distinctive views, the object model summarizes how actions change viewpoints and, thus, the state of the coupled robot/object system. In addition to visual sensors, extensions to tactile, auditory and other sensors also become possible with this representation. The aspect transition graph model was first introduced by Sen [23]. In previous work a mechanism for learning these models by autonomous exploration was introduced using a fixed set of actions and observations [14], [15]. These models support belief-space planning techniques where actions are chosen to minimize the expected future model-space entropy and can be used to condense belief over objects more efficiently. In [13], the aspect transition graph model was extended to handle an infinite variety of observations and continuous actions.

Our work also relates closely to a body of work on Partially Observable Markov Decision Process (POMDP) [11]. Since different objects may result in a similar observation from certain viewpoints, there is uncertainty in the state of the interaction defined by aspects. The goal of a POMDP problem is to find the optimal policy that maximizes reward over a finite or infinite horizon. However the computational complexity was shown to be PSPACE-complete over finite horizons [19] and is undecidable over infinite horizons [16]. In this work, the first action on the shortest path to the goal state is executed based on the most likely current state.

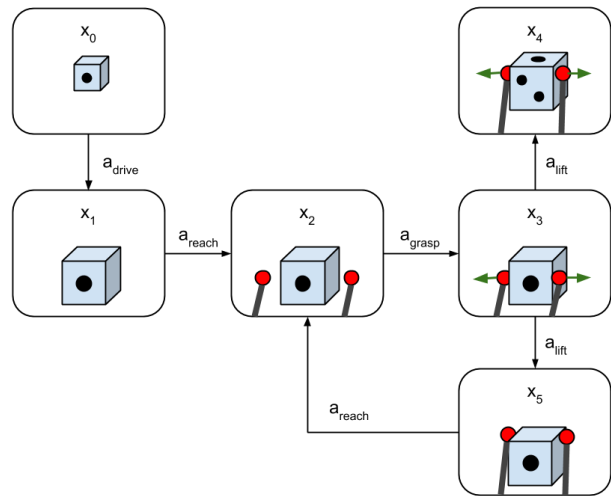


Fig. 2. Part of an aspect transition graph model of a dice. The top right node indicates the observation when the robot successfully flipped the dice while the bottom right node indicates when the dice slipped. The red circles indicate the robot hands and the green arrows indicate haptic feedback.

III. MODEL

The finite state controller (FSC) [20] used to solve many POMDP problems is defined as a directed graph where each node is an action and each edge is an observation. The aspect transition graph (ATG) model used in our work uses the dual of this graph where each node is an observation and each edge is an action. An aspect transition graph (ATG) object model is represented using a directed *multigraph* $G = (\mathcal{X}, \mathcal{U})$, composed of a set of aspect nodes \mathcal{X} connected by a set of action edges \mathcal{U} that capture the probabilistic transition between aspects. We define an “aspect” as a distinctive set of multiple features called an observation that is stored in the object model. An action edge U is a triple (X_1, X_2, A) consisting of a source node X_1 , a destination node X_2 and an action A that transitions between them. Figure 2 shows an example of part of an ATG model that captures possible observations when interacting with a die.

Unlike hidden Markov models (HMM) that assume observations are generated from a “true” hidden state such as the robot position, the ATG is an observation-based model that uses a single observation and the corresponding aspect node to represent a robot state directly. The observation-based approach avoids modeling and planning with hidden robot states and uses a model based on observation and memory. More details are described in the Recursive Bayesian Estimation subsection.

A. Observation

An ATG model is used to estimate robot state transitions based on the robot’s observations; however defining robot states directly in the observation space of raw sensor outputs is problematic due to the large volume of data in raw sensor feedback such as images. Therefore, instead of defining observation based on raw sensor feedback directly, we define an observation as a list of features detected by feature

detectors. Features are patterns in signals that reflect structure in the interaction between the robot and the environment. We assume that feature detectors are robust and output consistent patterns under the same conditions. Each feature is presented with a binary feature state that indicates whether the feature exists or not and are associated with a Cartesian position in R^3 . Position references for multiple features provide an estimate for the object pose with which to scale and orient actions that transition between aspects.

Visual features in the experiment designate: a feature type; the mean Cartesian position for the feature μ in R^3 ; and the estimated covariance for the feature location Σ in $R^{3 \times 3}$. For example, the ARtag feature type identifies a unique tag id, the Cartesian coordinate of the center of the tag in the robot (sensor) frame, and the Cartesian covariance of the position. A red blob feature uniquely identifies the position and uncertainty of the red robot hand in the visual signal. Haptic feature types include both positions and normals. For positions, μ designates contact positions with covariance Σ and for normals, μ designates a unit vector in R^3 and covariance Σ in $R^{2 \times 2}$ represents the uncertainty cone of the unit normal vector. The *grasp* feature type is computed over a population of contacts. It has two types that designate the force and moment residuals μ in $R^{3 \times 3}$ with uncertainty Σ in $R^{3 \times 3}$. Aspects that define intermediate states in interactions with an object, for example, are patterns of responses in these types of random variables that fit the Cartesian template stored in memory of past interactions with the object. In the experiments presented in this paper, a simple 28 cm cube object is employed with these features distributed over the six square surfaces of the cube. The size of the set of features can be arbitrarily large. For example, the set of features that belongs to the ARtag feature type includes features that represent all possible tag patterns. We describe how we detect feature types that have a large set of features in the Top Down Inference subsection.

An aspect node X in an ATG model represents one observation of an object O and stores a list of feature states described above. We say that two observations are equal if all features that have positive states in one of the observations also have positive states in the other observation. There could be multiple aspect nodes with the same observation within one or more ATG models.

B. Action

In an ATG model an action edge U represents an action that causes an aspect transition. Actions can be implemented with open-loop or closed-loop controllers. For each action, its type, parameterization, and reference frame are stored. A robot has a set of actions available to interact with objects. These actions include visual and haptic servoing actions as well as gross motor (mobility) actions and fine motor (arm and hand) actions. While mobility actions can be used to bring an object into reach for manipulation, they also alter the view of an object, possibly revealing previously hidden visual features. Actions such as pick-up and lift can be used to manipulate objects, but they can also gather

```

1: procedure BAYES FILTER( $bel(x_{t-1}), a_t, z_t$ )
2:   for all  $x_t$  do
3:      $\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t|a_t, x_{t-1}) \cdot bel(x_{t-1})$ 
4:      $bel(x_t) = p(z_t|x_t) \cdot \overline{bel}(x_t)$ 
5:   end for
6:   NORMALIZE( $bel(x_t)$ )
7: end procedure

```

Fig. 4. Bayes Filter Algorithm

haptic feedback and change the current viewpoint to reveal otherwise unavailable features (e.g. surface markings on the bottom face of a box).

High-level manipulation actions can be represented as a sequence of primitive actions and aspect transitions. For example, the “flip” macro action is implemented as a sequential composition of the following four primitive actions: 1) reach, 2) grasp, 3) lift, and 4) place. These four actions connect five fine-grained aspect nodes that represent expectations for intermediate observations. Figure 3 shows these intermediate stages. These fine-grained aspect nodes allow us to monitor and detect unplanned transitions at many intermediate stages of the flip interaction. In Section IV, we describe how failures are handled in the ATG model.

C. Recursive Bayesian Estimation

In our framework the Bayesian filtering algorithm [24] is used to estimate which aspect node likely generates the current observation. The recursive Bayesian estimation described in Figure 4 is used to update the posterior belief bel over all aspect nodes x_t when action a_t is executed from state x_{t-1} and new observation z_t is acquired. The initial belief is set based on a prior distribution over all aspect nodes.

D. Top Down Inference

The number of different feature types that could exist in an environment could be enormous. A bottom up approach that runs all possible detectors would be inefficient and often infeasible in robotic applications. Although there is evidence that human vision is processed in parallel in the primary visual cortex [8], there is also evidence that secondary and tertiary cortical areas strongly focus perceptual effort on features that are related to tasks and expectations. We argue that mid-level perceptual features may be strongly influenced by top down expectations so that only a subset of detectors must be active at a time. This is consistent with a hypothesis introduced by Notredame *et al.* that explains how visual illusions could be caused by prior expectations overtaking the actual stimulus [18]. A lot of research in computer vision has also been done on building models that perform top down inference [26] [25]. However, when dealing with single images independently the ability of top down inference is limited. Most work uses information from larger image regions to infer low level features. In our work, we focus on top down inference that is based on past observations and how the actor interacts with objects and the environment.

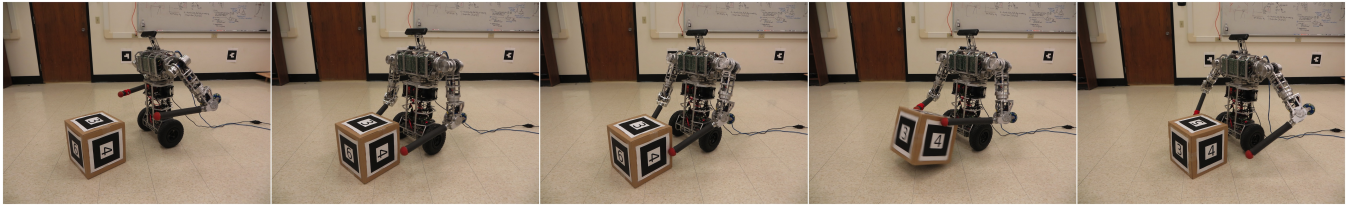


Fig. 3. Fine-grained flip action. The photos shown from left to right are the five intermediate stages of a flip action. The robot checks whether the sub-action succeeded for each stage.

Given a new set of raw sensor feedback, our framework does not run all available feature detectors. Instead, only feature detectors corresponding to features in the feature list of an aspect node with significant posterior probability are executed. We define a posterior probability to be significant if it is within a fixed percentage difference to the maximum posterior probability among all aspect nodes.

E. Acquiring the Model

In our previous work [14], [15], we demonstrated that ATG models can be learned without supervision from a fixed set of actions and observations. However, with a set of actions that take continuous parameters, learning ATG models without supervision may require a significant amount of exploration over the parameter space. In [13], we introduced a way to learn ATG models through imitation learning by teleoperating the robot and memorizing action parameters that transition between aspect nodes. The work presented in this paper focuses on how previously acquired ATG models are used to monitor for transition errors during the execution of a task. Therefore, we employ hand-built ATGs for objects as well as action for transitioning between aspect nodes. During surprise transitions new models that capture new transitions are created based on these hand-built models.

IV. HANDLING UNCERTAINTIES

In a non-deterministic environment it is inevitable that a robot will encounter randomness and surprises while performing tasks. In this section, we describe approaches to handling uncertainty in robotics, including a new definition of surprise.

A. Modeling Randomness

We define a random transition as an outcome of an action that is non-deterministic but is within a previously seen set of outcomes that can be captured in the model. For example, due to uncertainty, the “flip” macro in Section III-B may place the object in an ungraspable pose and, thus, causing a different observation in the terminating state. These random outcomes are represented in the ATG using transition probabilities $p(x_t|a_t, x_{t-1})$, which is nonzero for all possible outcome states. As a consequence, the Bayesian filtering algorithm will update the belief accordingly. For simplicity, we assign an equal transition probability to all of the outcome states possible under action a_t . In general, each outcome of a randomized transition requires a different action to achieve the goal aspect.

B. Recovery from Surprises

In [3], surprise is described as “the difference between expectation and reality.” Although one might associate surprise or unexpected events with low probability events, observing a rare event does not always lead to surprise. For example, a person that observes a lottery drawing will likely not be surprised if the outcome is 29 - 41 - 48 - 52 - 54 even though this outcome has a low probability.

Baldi [1] measures the degree of surprise in a Bayesian setting in which there is a well-defined distribution $P(M)$ over the space of models \mathcal{M} . He uses a measure S of how much the new information in an observation d changes the distribution over models. Specifically, he computes the KL-divergence between the current belief state $P(M)$ and the belief state $P(M|d)$ induced by the new observation:

$$S(d, \mathcal{M}) = \int_{\mathcal{M}} P(M) \log(P(M)/P(M|d)) dM.$$

An event is then defined to be a surprise if this divergence is greater than some threshold θ .

However, this formula considers transitions where an informative action reduces the entropy over models significantly as surprising. If there is a uniform distribution over models, observing an outcome that concentrates the posterior probability on one model should not be surprising. We offer a simpler definition of surprise that agrees better with intuition. Our definition can be applied to a distribution over models, but it also applies when there is just a single model M , and we start with this simpler case.

The entropy $H_M(D) = E[-\log P_M(D)]$ associated with a model M of observation D defines the expected negative log probability of an observation D . We define a specific observation d to be *surprising* if its *information*, $-\log P_M(d)$, is much greater than the entropy:

$$-\log(P_M(d)) \gg H_M(D).$$

Consider the example of a perfect fair die with six outcomes. No outcome can be surprising since the information of each outcome is equal to the entropy of the die. On the other hand, if we acknowledge that there is some minute probability of a die landing on its corner and balancing, then a better estimate of the probability of each face is $\frac{1}{6} - \epsilon$, and the probability of landing on a corner extremely small. If the die does in fact land on a corner, it is not the rarity of this event that makes it surprising, but the probability *relative to the entropy of the*

die. That is, the event is thousands of times less likely than what we expected, which is $\frac{1}{6} - \epsilon$.

In the Bayesian setting, where the model M is unknown, we extend our definition of surprise by simply computing expectations over the unknown models:

$$E_{\mathcal{M}}[-\log(P_M(d))] \gg E_{\mathcal{M}}[H_M(D)].$$

Intuitively, a “surprising event” is still an outcome whose probability is far lower (and information is much higher) than what was expected. And again, when all events are a uniform low probability, there can be no surprise.¹

In practice, rather than trying to define a precise probability for extremely rare events, we simply define events that are not part of a model as having extremely low probability. For example, while the robot is grasping an object if someone covered the robot’s camera and such a transition is not modeled in any of the object models, the observation will be inconsistent with what the robot expects (i.e., has an extremely low probability) and therefore be classified as a surprise transition. During the surprise transition a new ATG model that includes this new transition is created by extending the ATG model with the highest belief in memory. If an identical transition reoccurs the robot will be able to expect the outcome with the new model.

Since surprise transitions are unpredictable and do not lie within a bounded set of possible outcomes, they are handled by resetting the belief among all aspect nodes to the prior distribution. If a possible plan that will lead to the goal still exists, the robot will continue to accomplish the task. In our experiment, when a surprise transition occurs the robot also shows a “What’s up?” gesture that conveys confusion as in Figure 1. This gesture has proven to be useful for indicating the robot’s current state in experiments.

V. ALGORITHM

The action selection algorithm that uses the ATG to achieve goals and handle surprise transitions is described in Figure 5. Here, \mathcal{M} represents a set of ATG models stored in the robot memory and x_{target} is the target aspect given to the robot. The function $INITIALIZEBELIEF(\mathcal{M})$ initializes prior probability specified in the set of ATG models \mathcal{M} . The function $INFERASPECTS(\mathcal{M}, \overline{bel})$ infers a set of possible aspects $X_{candidate}$ based on the belief \overline{bel} . The function $GETOBSERVATION(X_{candidate})$ returns the current observation $z_{current}$ and the corresponding feature positions in R^3 f_{pose} from active feature detectors derived from the set of possible aspects $X_{candidate}$. The function $BAYESFILTEROBS(\overline{bel}, z_{current})$ refers to part of the Bayesian filtering algorithm that updates the belief given an observation. The function $SURPRISE(bel)$ returns true if the belief in all aspect nodes are zero. If a surprise transition occurs, the robot shows the “What’s up?” gesture, backs up and move arms to the ready pose. The

¹This addresses the case of “television snow” that Baldi raises. When snow occurs in the context of a low entropy process, it is surprising, since it has much lower probability. However, when it occurs in the context of a snow distribution, it has probability equal to the other outcomes.

```

1: procedure ALGORITHM( $x_{target}, \mathcal{M}$ )
2:    $\overline{bel} = INITIALIZEBELIEF(\mathcal{M})$ 
3:   while true do
4:      $X_{candidate} = INFERASPECTS(\mathcal{M}, \overline{bel})$ 
5:      $z_{current}, f_{pose} = GETOBSERVATION(X_{candidate})$ 
6:      $bel = BAYESFILTEROBS(\overline{bel}, z_{current})$ 
7:     if  $SURPRISE(bel)$  then
8:        $WHAT'SUPGESTURE()$ 
9:        $CREATENEWMODEL(\mathcal{M}, \overline{bel}, a, z_{current})$ 
10:       $\overline{bel} = RESETBELIEF(\overline{bel}, \mathcal{M})$ 
11:       $BACKUPANDREADYPOSE()$ 
12:      continue
13:    end if
14:     $a = INFERACTION(\mathcal{M}, x_{target}, bel)$ 
15:     $EXECUTEACTION(a, f_{pose})$ 
16:     $\overline{bel} = BAYESFILTERACT(bel, a)$ 
17:  end while
18: end procedure

```

Fig. 5. Algorithm for achieving goal aspect and handling surprise transitions.

function $CREATENEWMODEL(\mathcal{M}, \overline{bel}, a, z_{current})$ creates a new ATG model by adding a new aspect node representing $z_{current}$ and an action edge representing action a to the ATG model with the highest belief in \mathcal{M} . The belief over aspects are reset to the prior distribution through function $RESETBELIEF(\overline{bel}, \mathcal{M})$. The function $INFERACTION(\mathcal{M}, x_{target}, bel)$ finds a shortest path to the target aspect x_{target} . The function then returns the first action on the shortest path. $EXECUTEACTION(a, f_{pose})$ executes the action based on information stored in the action edge a and feature positions f_{pose} . The function $BAYESFILTERACT(bel, a)$ refers to part of the Bayesian filtering algorithm that updates the belief given an action.

VI. EXPERIMENTAL RESULTS

In order to evaluate the techniques we introduced to handle randomness and uncertainty, two different experiments were performed. For both cases, the uBot-6 mobile manipulator has to perform simple manipulation tasks. The performance is compared with and without the presented techniques.

A. Experimental Setting

The goal of task is to manipulate an ARcube until the requested features are observed on the corresponding faces. An ARcube is a cube box with a different ARtag on each of its face as shown in Figure 3. An ARtag is a square fiducial marker commonly used in robotics due to ease of use and robust detection. In this work an ATG model representing an ARcube has 174 nodes and 408 edges.

B. Surprise Recovery

For the first experiment, the model is provided with a fairly detailed ATG model of the handled ARcube. The robot has to reach the desired state with ARtags 1 and 2 on the top and front faces of the cube respectively. The robot has to first perform this task solely based on the provided model and the contained transitions. Despite the model being fairly complete, a small perturbation in the visual and

haptics sensors, as well as slightly different dynamics when manipulating the ARcube can result in an observation not contained in the model or not agreeing with the model (e.g. an action resulting in new outcome). Without the ability to recover, reaching such a state, the robot will not be able to complete the task.

The robot was repeatedly provided with random configurations of the ARcube and the performance of both methods was compared. The robot was able to complete the task reliably 10 out of 10 times when provided with the ability to recover from unforeseen observations. Without this capability, the robot succeeded 5 out of 10 runs.

C. Error Detection Through Fine-Grained Actions

In the second experiment, the robot was presented with an ARcube such that one flip action of the cube will lead to the desired goal state. The ARcube was then perturbed during the grasping action to cause a grasp failure. The performance of the robot was evaluated based on the number of actions it needed to complete the task despite this perturbation.

We tested two different setups. In the first setup the robot has access to a “flip” macro which combines four primitive actions described in Section III-B into one macro action. In the second setup, the robot uses the fine-grained actions and aspects directly. The experiment was performed 10 times for each setup. Using the flip macro, the robot needed 16.1 ± 3.37 primitive actions to complete the task, whereas with the fine-grained actions it only needed 10.6 ± 0.69 primitive actions. The difference in the required number of primitive actions is due to the capability of detecting the error much earlier and reacting appropriately when using fine-grained actions and aspects.

VII. CONCLUSION

In this work we describe how failures in stochastic robotic actions can be detected and handled properly. We introduce a general framework that stores fine-grained event transitions for early error detection. We further categorize detected failures into random transitions or surprise transitions based on how it is perceived by the robot. These two types of uncertainties are then handled and recovered separately. We have shown that the overall framework increases the robustness towards handling uncertainties and decreases the number of actions needed on manipulation tasks experimented on the uBot-6 mobile manipulator.

VIII. ACKNOWLEDGMENT

The authors would like to thank Mitchell Hebert, Michael Lanighan, Tiffany Liu, and Takeshi Takahashi for their contributions. This material is based upon work supported under Grant NASA-GCT-NNX12AR16A and a NASA Space Technology Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Aeronautics and Space Administration.

REFERENCES

[1] Pierre Baldi. A computational theory of surprise. In *Information, Coding and Mathematics*, pages 1–25. Springer, 2002.
 [2] Daniel E Berlyne. Conflict, arousal, and curiosity. 1960.

[3] John L Casti. Complexification explaining a paradoxical world through the science of surprise. 1994.
 [4] Bruce R Donald. A geometric approach to error detection and recovery for robot motion planning with uncertainty. *Artificial Intelligence*, 37(1):223–271, 1988.
 [5] J Gibson. Perceiving, acting, and knowing: Toward an ecological psychology. *chap. The Theory of Affordance*. Michigan: Lawrence Erlbaum Associates, 1977.
 [6] Ziv Gigus and Jitendra Malik. Computing the aspect graph for line drawings of polyhedral objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(2):113–122, 1990.
 [7] Harry F Harlow. Learning and satiation of response in intrinsically motivated complex puzzle performance by monkeys. *Journal of Comparative and Physiological Psychology*, 43(4):289, 1950.
 [8] DH Hubel, TN Wiesel, and MP Stryker. Orientation columns in macaque monkey visual cortex demonstrated by the 2-deoxyglucose autoradiographic technique. *Nature*, 269(5626):328, 1977.
 [9] Clark Leonard Hull. Principles of behavior: an introduction to behavior theory. 1943.
 [10] Laurent Itti and Pierre F Baldi. Bayesian surprise attracts human attention. In *Advances in neural information processing systems*, pages 547–554, 2005.
 [11] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.
 [12] Jan J Koenderink and Andrea J van Doorn. The internal representation of solid shape with respect to vision. *Biological cybernetics*, 32(4):211–216, 1979.
 [13] Li Yang Ku, Erik G Learned-Miller, and Roderic A Grupen. Modeling objects as aspect transition graphs to support manipulation. *International Symposium on Robotics Research*, 2015.
 [14] Li Yang Ku, Shiraj Sen, Erik G Learned-Miller, and Roderic A Grupen. Action-based models for belief-space planning. *Workshop on Information-Based Grasp and Manipulation Planning, at Robotics: Science and Systems*, 2014.
 [15] Li Yang Ku, Shiraj Sen, Erik G Learned-Miller, and Roderic A Grupen. Aspect transition graph: an affordance-based model. *Second Workshop on Affordances: Visual Perception of Affordances and Functional Visual Primitives for Scene Analysis, at the European Conference on Computer Vision*, 2014.
 [16] Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In *AAAI/IAAI*, pages 541–548, 1999.
 [17] Kay C Montgomery. The role of the exploratory drive in learning. *Journal of Comparative and Physiological Psychology*, 47(1):60, 1954.
 [18] Charles-Edouard Notredame, Delphine Pins, Sophie Deneve, and Renaud Jardri. What visual illusions teach us about schizophrenia. *Frontiers in integrative neuroscience*, 8, 2014.
 [19] Christos H Papadimitriou and John N Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.
 [20] Pascal Poupart and Craig Boutilier. Bounded finite state controllers. In *Advances in neural information processing systems*, page None, 2003.
 [21] Alberto Rodriguez, Matthew T Mason, Siddhartha S Srinivasa, Matthew Bernstein, and Alex Zirbel. Abort and retry in grasping. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1804–1810. IEEE, 2011.
 [22] Dirk Ruiken, Michael W Lanighan, Roderic Grupen, et al. Postural modes and control for dexterous mobile manipulation: the umass ubot concept. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 280–285. IEEE, 2013.
 [23] Shiraj Sen. *Bridging the gap between autonomous skill learning and task-specific planning*. PhD thesis, University of Massachusetts Amherst, 2013.
 [24] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.
 [25] Sinisa Todorovic and Narendra Ahuja. Region-based hierarchical image matching. *International Journal of Computer Vision*, 78(1):47–66, 2008.
 [26] Tianfu Wu and Song-Chun Zhu. A numerical study of the bottom-up and top-down inference processes in and-or graphs. *International journal of computer vision*, 93(2):226–252, 2011.