Modeling Objects as Aspect Transition Graphs to Support Manipulation

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

Abstract Designing robots that can model unstructured environments and act predictably in those environments is a challenging problem. In this article we address two issues in object modeling and manipulation. First, in image-based models, how can we choose a discrete set of canonical views from an infinite set of possible choices to support recognition and manipulation? Second, how can we make actions repeatable and predictable in unstructured environments? We propose an object model that handles both of these issues in a coherent way and introduce a novel image-based visual servoing algorithm that works in conjunction with the object model. We then demonstrate our object model and visual servoing algorithm on a tool grasping task on the Robonaut 2 simulator.

1 Introduction

In the fields of human psychophysics and neurophysiology, the study of visual object recognition is often motivated by the question of how humans recognize 3-D objects while receiving only 2-D light patterns on the retina [29]. Two types of models for object recognition have been proposed to answer this question. The structural description model represents each object by a small number of view-invariant primitives and their position in an object-centered reference frame [23]. Alternatively, image-based models represent each object as a collection of viewpoint-specific local features. Since the development of these models, experiments in human psychophysics and neurophysiology have provided converging evidence for image-based models. In experiments done by Bülthoff and Edelman [6] [2], it was shown that when a new object is presented to a human subject, a small set of canonical views are formed despite the fact that each viewpoint is presented to the subject for

L. Ku · E. Learned-Miller · R. Grupen
College of Information and Computer Science, University of Massachusetts Amherst, MA, USA
e-mail: {lku,elm,grupen} @cs.umass.edu

the same amount of time. Experiments on monkeys further confirmed that a significant percentage of neurons in the inferior temporal cortex responded selectively to a subset of views of a known object [20]. However, how an infinite set of possible views can be effectively reduced to a smaller set of canonical views remains an open question. Different approaches such as view interpolation [24] and linear combinations of views [31] have been proposed.

Closely related to the image-based models in the field of psychophysics, aspect graphs were first introduced as a way to represent 3-D objects using multiple 2-D views in the field of computer vision [16]. An aspect graph contains distinctive views of an object captured from a viewing sphere centered on the object. Research on aspect graphs has focused on the methodologies for automatically computing aspect graphs of polyhedra [10] and general curved objects [17]. The set of viewpoints on the viewing sphere is partitioned into regions that have the same qualitative topological structure as an image of the geometric contours of the object. However, work done in this field was mostly theoretical and was not applicable in practice [7]. One of the difficulties faced in this work concerned the large number of aspects that exist for normal everyday objects. An object can generate millions of different aspects, but many of these may be irrelevant at the scale of the observation. In this work, we propose an object model that provides a consistent treatment for classifying observations into aspects within a practically-sized subset of all possible aspects for most types of objects including deformable objects.

Object and tool manipulation are essential skills for a humanoid robot, and recognizing known objects and tools is often a first step in manipulation tasks. In computer vision and robotics, object recognition is often defined as the process of labeling segments in an image or fitting a 3-D model to an observed point cloud. The object models used to accomplish these tasks usually include information about visual appearance and shape. However, what these object recognition systems provide is merely a label for each observed object. The sequence of actions that the robot should perform based on the object label are often manually defined. Without linking actions to object labels these object models themselves have limited utility to the robot.

Both aspect graphs and image-based models attempt to model 3-D objects with multiple 2-D views. Research in aspect graphs has encountered difficulties in determining the threshold to differentiate two distinctive views while for image-based models how to generalize from unfamiliar to canonical views remains an open question. In this article we propose an object model that addresses both of these issues and incorporates actions in a coherent way. In particular, we show how aspects can be chosen in a unique and repeatable way that is defined by the object itself, and in a way that supports manipulation.

While many of our examples use images and visual processing, our methodology applies to other modes of perception such as audition and haptics. Below, we use the terms “observation” and “aspect” instead of “view” and “canonical view” to reflect the more general nature of our approach beyond just visual processing.

The three main contributions of this paper are the following. 1) We define a principle that determines whether two observations should be differentiated or gen-

eralized to one aspect based on the actor's capability. 2) We propose an image-based visual servoing algorithm that allows the actor to manipulate an object to cause the features in an image to conform with an aspect in memory. 3) We introduce a method for determining whether a sequence of non-deterministic manipulation actions can, under certain assumptions, be guaranteed to transition between two aspects. We demonstrate our object model and our visual servoing algorithm on a tool-grasping task using the Robonaut 2 simulator.

2 Related Work

Besides work done in aspect graphs and image-based models mentioned in the last section, our work also relates to a body of work in hybrid control theory. In [3], a controller is described as a funnel that guides the robot state to convergence; multiple controllers can be combined to funnel robot states to a desired state that no one single controller can reach alone. In [30], an algorithm that combines linear quadratic regulators into a nonlinear policy was also introduced. However under certain situations the goal state may not be reachable through a combinations of controllers that act like funnels. For example, the visual servoing controller implemented in our experiment controls the end effector to a certain pose based on the robot hand's visual appearance. However to reach the goal state, a controller that transitions from a state where the robot hand is not visible to one in which the visual servoing controller can be executed is required. Such a controller can be an open loop controller that moves the end effector to a memorized pose and may not necessarily converge to a certain state like a funnel.

In this work we introduce the notion of a *slide* as a metaphor for this kind of action that transitions from one set of states to another (see Figure 1). Uncertainty of the state may increase after transitioning down a slide, but may still reach the goal state if a funnel-slide-funnel structure is carefully designed. We investigate how a sequence of these two kinds of controllers will change how an object is observed. In previous (on-going) work we have referred to funnels as *track* control actions and slides as *search* control actions [11]. The search control action orients the visual sensor to where the target is likely be found therefore transitioning states like a slide; the track control action keeps the target in the visual center and converges to a subset of states like a funnel. Figure 1 illustrates the funnel-slide-funnel concept using the same style of figure demonstrated in previous work by Burridge et al [3].

There is also a good deal of related work in visual servoing. This work can be classified into two major types: position-based servoing, where servoing is based on the estimated pose; and image-based servoing, where servoing is based directly on visual features [14]. The image-based servoing approach has the advantage that it performs with an accuracy independent of extrinsic camera calibration and does not require an accurate model of the target object or end effector. Our visual servoing approach belongs to this class of image-based servoing techniques.

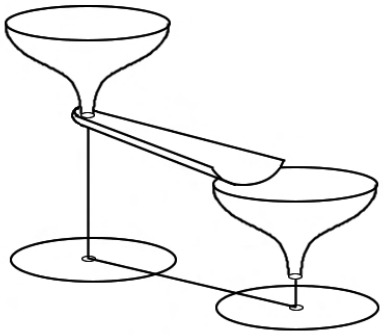

Fig. 1 Funnel-slide-funnel structure. We use the funnel metaphor introduced in [3] to describe a closed-loop controller or a track control action [11] that converges to a subset of states and the slide metaphor to describe an open-loop controller or a search control action [11] that causes state transitions.

Our work is inspired by Jägersand and Nelson [15], in which Broyden's method is used to estimate the visuomotor Jacobian online. Our algorithm uses a similar update approach but is implemented on top of a changing set of features. Some other work in visual servoing has also investigated approaches that do not rely on a predefined set of features. In [26], a set of robust SIFT features are selected to perform visual servoing. In [12] moments of SIFT features that represent six degrees of motion are designed. An approach that is based on the image entropy was also introduced in [4]. However these approaches all assume a setting in which the camera is mounted on the end effector. In this article we are interested in a setting that is more similar to human manipulation. Unlike a system where the camera is mounted on the end effector, only part of the observed features move in correspondence with the end effector. Our algorithm is used to guide the robot end effector, within the field of view, to a pose that is defined relative to an object that was memorized. The features that are controllable are learned and reused.

Our work also has many connections to prior work on *affordances*. The term affordance [9] has many interpretations. We prefer the definition of affordance as "the opportunities for action provided by a particular object or environment" [8]. Affordances can be used to explain the functionality and utility of things in the environment. 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. An approach to bind affordances of objects with the robot was also introduced by Stoytchev [27]. In this work, the robot learns sequences of actions that will lead to invariant features on objects through random exploration. In the object model introduced in [33], predefined base affordances are associated with object surface types. Instead of defining object affordances from a human perspective, our object models memorize how robot actions change perception with a graph representation.

The aspect transition graph model employed in this work was first introduced by Sen [25]. In our previous work [18] [19], we introduced a mechanism for learning these models without supervision, from a fixed set of actions and observations. We used these models to support belief-space planning techniques where actions are chosen to minimize the expected future model-space entropy, and we showed that these techniques can be used to condense belief over objects more efficiently. In this article we extend the aspect transition graph model to handle an infinite variety

of observations and to handle continuous actions. We start with a discussion of our aspect transition graph model.

3 Object Model

The aspect transition graph (ATG) object model discussed in this paper is an extension of the original concept of an aspect graph. In addition to distinctive views, the ATG object model summarizes how actions change viewpoints or the state of the object and thus, the observation. We define the term "observation" to be the combination of all sensor feedback of the robot at a particular time and the "observation space" as the space of all possible observations. This limits the model to a specific robot, but allows the model to present object properties other than viewpoint changes. Extensions to tactile, auditory and other sensors is possible with this representation. An ATG model of an object can be used to plan manipulation actions for that object to achieve a specific target aspect. For example, in order for the robot to pick up an object, the target aspect is a view where the robot's end effector surrounds the object. We expect that this view will be common to many such tasks and that it can be the expected outcome of a sequence of slides (i.e. like moving the effector to the same field of view as the target object) and funnels (like visually servoing features from the hand into the pregrasp configuration relative to the object).

Definitions

We define an "aspect" as a single observation that is stored in the object model. This usage is consistent with the term "canonical view" coined in the psychophysics literature to describe image-based models. As we will see below, many observations will not be stored in the object's memory and hence will not be categorized as aspects. We will discuss in detail below how a given observation is categorized as an aspect or not.

An ATG object model is represented using a directed *multigraph*[1] $G = (\mathcal{X},\mathcal{U})$, 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. 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. Note that there can be multiple action edges (associated with different actions) that transition between the same pair of nodes. In contrast to aspect graphs and image-based models that differentiate views based on visual appearance, we argue that, in general, discriminating between object observations should depend on whether the actor is capable of manipulating the object such that the observation converges to a target aspect. That is, we define aspects that are functions of the visual servoing and action abilities of the robot.

[1] A multigraph allows multiple edges between a given pair of vertices.

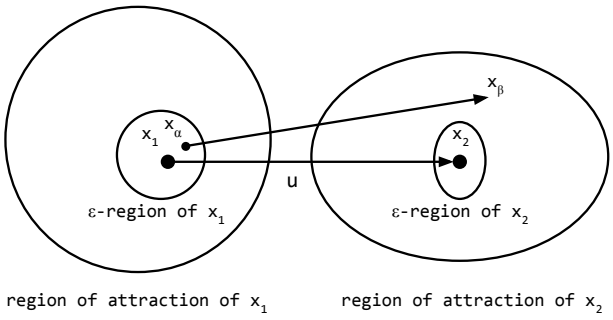

Fig. 2 An ATG model containing two aspects x_1 and x_2, each a likely result of applying a funnel action within their respective regions of attraction. The edge labeled u is a model-referenced "slide" action that reliably maps the ε-region of x_1 to the interior of the region of attraction of x_2.

Figure 2 shows an example of an ATG model that contains two aspects x_1, x_2 and one action edge u connecting the two aspects in the observation space. An aspect is represented as a single dot in the figure. The smaller ellipses around x_1, x_2 represent the ε-region of the corresponding aspect. Inside the ε-region, the observation is close to the target aspect, and the funnel action is considered to have "converged". The ε-region is task dependent; a task that requires higher precision such as picking up a needle will require a smaller ε-region. Each aspect x is located in the ε-region but does not have to be in the center. The location and shape of the ε-region also depends on the given task since certain dimensions in the observation space might be less relevant when performing certain tasks.

The larger ellipses surrounding the ε-regions are the region of attraction of the "funnel" controller referenced to aspects x_1 and x_2. Observations within the region of attraction converge to the ε-region of the target aspect by running a closed-loop controller that does not rely on additional information from the object model. In our experiment, a visual servoing controller is implemented to perform gradient descent to minimize the observation error. The region of attraction for using such a controller is the set of observations from which a gradient descent error minimization procedure leads to the ε-region of the target aspect.

Slides

The arrow in Figure 2 that connects the two aspects is an action edge (x_1, x_2, a) that represents a "slide" action. Action a is an open-loop controller that causes aspect transitions. Instead of converging to an aspect, "slide" actions tend to increase uncertainty in the observation space. If a funnel is used to describe a convergent controller then a slide is suitable for describing this type of action. Figure 1 illustrates this metaphor with an example structure that allows transitions from a converged aspect to the mouth of another funnel.

We implement slide actions as open-loop controllers. In our experiments, a slide action a can be represented in the form $a = \phi|_{\tau}^{\tilde{\sigma}}$ where ϕ represents the potential function that the controller tries to minimize, $\tilde{\sigma}$ represents a set of memorized controller parameters, and τ represents the motor resources the action controls. An example is an end point position controller that moves to a relative pose with respect to the center of an object point cloud. Under situations when there is no randomness in observation, action execution and the environment, executing action a from aspect x_1 will transition reliably to aspect x_2.

Convergence

The arrow in Figure 2 that connects the observation x_α within the ε-region of x_1 to observation x_β represents a scenario where action a is executed when x_α is observed in a system in which actions have stochastic outcomes. We define ε_u as the maximum error between the aspect x_2 and the observation x_β when action a is executed while the current observation is within the ε-region of aspect x_1. ε_u can be caused by a combination of kinematic and sensory errors generated by the robot or randomness in the environment. If the region of attraction of the controller that converges to aspect x_2 covers the observation space within ε_u from x_2, by running the convergent controller we are guaranteed to converge within the ε-region of aspect x_2 under such an environment. Figure 1 illustrates this using the funnel and slide metaphor. As long as the end of the slide is within the mouth of the next funnel we can guarantee convergence to the desired state even when open loop controllers are within the sequence. The target aspect x_2 is determined by estimating the most likely observation after executing action a through the Bayesian filtering algorithm.

Completeness and Sufficiency

We call an Aspect Transition Graph model *complete* if the union of the regions of attraction over all aspects cover the whole observation space and a path exists between any pair of aspects. A complete ATG object model allows the robot to manipulate the object from any observation to one of the aspects. Complete ATG object models are informative but often hard to acquire and do not exist for irreversible actions. On the other hand, it is not always necessary to have a complete ATG to accomplish a task. For example, a robot can accomplish most drill related tasks without modeling the bottom of the drill. Therefore, we define an Aspect Transition Graph object model to be *sufficient* if it can be used to accomplish all required tasks of the object. In this work we will focus on sufficient ATG object models.

4 Visual Servoing

In this section we introduce an image-based visual servoing algorithm under the control basis framework [13]. This visual servoing controller is used to converge

from an observation within the region of attraction to the ε-region of the corresponding aspect. An action is written in the form $\phi|_{\tau}^{\sigma}$, where ϕ is a potential function, σ represents sensory resources allocated, and τ represents the motor resources allocated [13]. The control basis framework provides a means for robot systems to explore combinations of sensory and motor controls. Although only visual data are used in this work, the control basis framework allows us to combine controllers that utilize sensory resources of different modalities in future work. In our experiment the visual servoing controller is used to control the end effector of the robot to reach a pose relative to a target object using visual sensor feedback. Unlike many visual servoing approaches, our visual servoing algorithm does not require a set of predefined visual features on the end effector or target object nor does it require an inverse kinematic solution for the robot. The only information required is the current observation and the target aspect. Figure 3 shows a trial of our visual servoing algorithm converging to a stored target aspect.

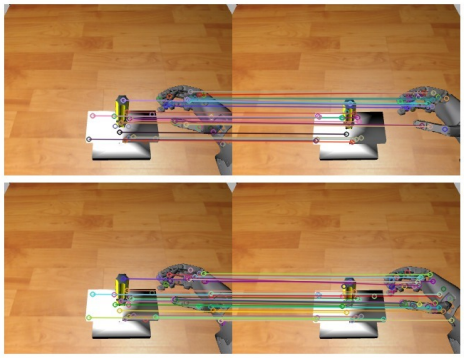

Fig. 3 Visual servoing sequences. Each image pair shows the target aspect (left) and the current observation (right). A line in between represents a pair of matching keypoints. The top image pair represents the starting observation and the bottom image pair represents when the controller converged.

Potential Function

In the control basis framework, a potential function ϕ represents an error function that the controller minimizes. To reach minimum error a closed loop controller performs gradient descent on the potential function to converge to a minimum. Artificial potential functions that guarantee asymptotically stable behavior are usually used to avoid local minima [11]. However in visual servoing, potential functions with a unique minimum often do not exist due to occlusion, lighting and noisy sensory data. Instead of trying to define a potential function with a unique minimum, we define a potential function with possibly many local minima and call the region in which gradient descent converges to a particular minimum the region of attraction. If the current aspect is within the region of attraction we can guarantee convergence to the target aspect through gradient descent.

Our potential function is defined as the weighted squared Euclidean distance between the signature of the current observation $\tilde{s}$ and the signature of the target aspect s. This approach can be used with most feature detectors and feature descriptors. In our experiment the Fast-Hessian detector and the SURF descriptor [1]

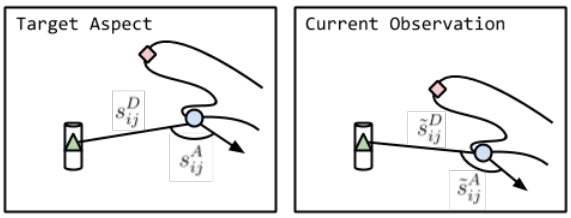

Fig. 4 Components of the signature of the target aspect (left) and the current observation (right). The circle and the triangle represent the *i*th and *j*th matched keypoints.

are implemented. A depth filter that uses the depth image is first used to filter out most keypoints that belong to the background. The first step to calculate the signature of an observation is to find a subset K of keypoints in the current observation that match to keypoints in the target aspect. The signature of an observation can then be calculated based on this subset K of keypoints. The signature is a combination of the distance signature vector $\boldsymbol{s}^D$ and the angle signature vector $\boldsymbol{s}^A$. $\boldsymbol{s}^D$ is a signature vector that consists of Euclidean distances s_{ij}^D between all pairs of keypoints (k_i, k_j) in K: $s_{ij}^D = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. Here x_i, y_i are the X Y image coordinates of keypoint $k_i \in K$. The angle signature vector $\boldsymbol{s}^A$ consists of angle differences s_{ij}^A between all pairs of keypoints (k_i, k_j) in K: $s_{ij}^A = \omega_{ij} - \theta_i$. Here ω_{ij} represents the orientation of the ray from keypoint k_i to keypoint k_j and θ_i represents the orientation of keypoint k_i. Figure 4 illustrates examples of s_{ij}^D and s_{ij}^A of the target aspect and the current observation.

The potential ϕ is then the scaled squared Euclidean distance between distance signature vectors of the target aspect $\boldsymbol{s}^D$ and the current observation $\tilde{\boldsymbol{s}}^D$ plus the weighted squared Euclidean distance between angle signature vectors of the target aspect $\boldsymbol{s}^A$ and the current observation $\tilde{\boldsymbol{s}}^A$;

$$\phi = \frac{1}{N_D} \cdot \sum_{\{i,j|k_i,k_j \in K\}} (s_{ij}^D - \tilde{s}_{ij}^D)^2 + \sum_{\{i,j|k_i,k_j \in K\}} w_{ij}^A \cdot (s_{ij}^A - \tilde{s}_{ij}^A)^2,$$

where $N_D = |K| \cdot (|K| - 1)/2$ and $w_{ij}^A = s_{ij}^D / \sum_{\{i,j|k_i,k_j \in K\}} s_{ij}^D$. Here $|K|$ is the number of matched keypoints between the current observation and the target aspect and w_{ij}^A is a normalized weight proportional to the keypoint pair distance s_{ij}^D in the target aspect. The purpose of w_{ij}^A is to weight angle differences more heavily for keypoints that are far apart.

Gradient Descent

In order to perform gradient descent on the potential function we need to be able to estimate the potential-motor Jacobian defined as $J = \partial\phi(\sigma)/\partial\tau$. A seven degree freedom arm is used in our experiment, therefore $\tau = [q_1, q_2, ..., q_7]$ where q_i represents the *i*th joint in Robonaut-2's right arm. The control signal that leads to the greatest descent can then be calculated by the expression: $\Delta\tau = -c(J^{\#}\phi(\sigma))$, where c is a positive step size and $J^{\#}$ is the Moore-Penrose pseudoinverse [22].

In order to calculate the partial derivative of the potential function ϕ with respect to each joint q, we introduce the visuomotor Jacobian defined as $J_v = \partial V / \partial \tau$, where V is the X Y positions and orientations of the set of keypoints detected in the current observation that match to keypoints in the target aspect based on its feature descriptor. Given $\Delta\tau$ and J_v we can calculate the change in the keypoint positions and angles through $\Delta V = J_v \cdot \Delta\tau$. Since the potential only depends on matched pairs we can calculate an estimated potential for every joint value.

Learning the Visuomotor Jacobian

Our visuomotor Jacobian that models how features change with respect to joint values is inspired by work done in understanding how humans obtain a sense of agency by observing their own hand movements [32]. Our approach learns that certain feature positions on the robot end effector are controllable while features in the background are not. Our visuomotor Jacobians for each aspect are updated on-line using a Broyden-like method $J_{v_{t+1}} = J_{v_t} + (\mu(\Delta V - J_{v_t}\Delta\tau)\Delta\tau^T / \Delta\tau^T\Delta\tau)$, where J_{v_t} is the visuomotor Jacobian at time t and $\mu \in (0,1]$ is a factor that specifies the update rate [21]. When $\mu = 1$ the updating formula will converge to the correct Jacobian J_v after m noiseless orthogonal moves and observations, where m is the dimension of J_v. In our experiment we set $\mu = 0.1$ to make the estimation more robust. The visuomotor Jacobians for each aspect are initialized randomly for the first run and memorized afterwards. The more trials the controller runs the more accurate the estimated J_v is on average. Using Broyden's method to estimate Jacobians on-line for visual servoing was first introduced in [15].

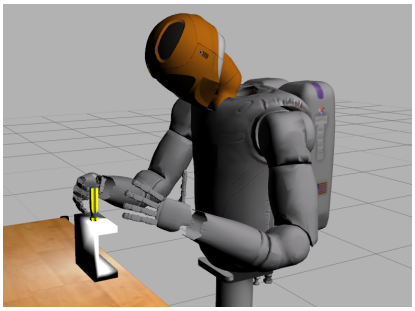

Fig. 5 Robonaut 2 approaching a pregrasp pose for a screwdriver on a tool stand in simulation.

5 Experimental Results

The aspect transition graph object model in conjunction with the visual servoing algorithm introduced in previous sections are tested on a tool grasping task on the NASA Robonaut-2 simulator [5]. The goal of the task is to control Robonaut-2's right hand to a pose where a screwdriver on a tool stand is in between the robot's right thumb, index finger and middle finger as shown in Figure 5. An ATG object model consisting of three aspects, that is sufficient for this task, was built through

demonstration. We show that the "slide-funnel-slide-funnel" controller sequence decreases the average pose error over a "slide-slide" controller sequence.

Building ATG Models

In this experiment our ATG object model is built through a teleoperated demonstration. An interface was implemented to allow the demonstrator to indicate when to create a new aspect in the object model. The demonstrator can control the robot end effector through interactive markers implemented by the MoveIt! platform [28]. When a new aspect is created, the action edge that connects the previous aspect to this new aspect can be inferred.

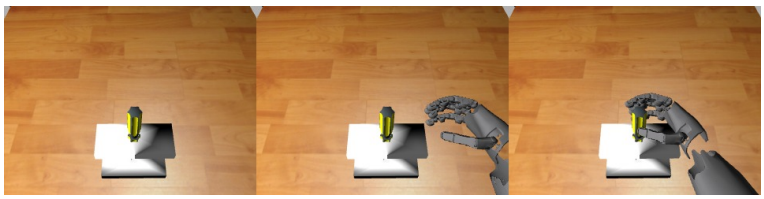

Fig. 6 The first, second, and third aspect stored in the ATG model through demonstration are shown from left to right. In the first aspect, the object on top of the table is a screwdriver on a tool stand. In the second aspect, the robot hand is in a position where a straight movement toward the screwdriver would lead to a pregrasp pose. The third aspect represents a pregrasp pose. This is the goal aspect for the pregrasp task designed in this experiment.

The ATG object model used in this experiment consists of three aspects. The first aspect represents an observation in which the screwdriver is on a tool stand on a table and is 0.6 meters in front of the robot. In addition, no parts of the robot are visible. The left image in Figure 6 is the corresponding observation of this aspect. The second aspect represents an observation where the robot's right hand is about 0.07 meters right of the screwdriver. The action edge between the first and second aspects represents an action that moves the robot's right hand to a pose relative to the center of the segmented point cloud observed in the first aspect. This point cloud is segmented based on the distance to the camera. The middle image in Figure 6 is the corresponding observation of this aspect. The third aspect represents an observation where the robot's right thumb, index and middle finger surrounds the screwdriver handle. The right image in Figure 6 is the corresponding observation of this aspect. The action edge in between the second and third aspects represents an action that moves the robot's right hand to a pose relative to the right hand pose of the previous aspect. The relative action frame is determined based on the closest observable feature to the end effector. An even better approach would be to assign action frames based on the intention of the demonstrator but this is beyond the scope of this paper.

Region of Attraction

The region of attraction of the second and third aspect of the ATG object model

with respect to the visual servoing controller can be analyzed. It is possible to also have a controller that is capable of converging to the first aspect through controlling joints in the robot's neck and waist, however since we assume the robot starts in a similar pose with similar observation this controller is not implemented in this experiment. The region of attraction of an aspect is defined as the observation space in which a closed loop convergence controller that does not rely on additional information from the object model can converge to the ε-region of the aspect. An aspect or observation lies in a high dimensional observation space and can be varied by multiple different parameters or noise. In this experiment we are interested in two types of noise. 1) Noise in the relative pose between the robot hand and the object. This kind of noise can be caused by kinematic errors from executing an action or imperfect object positions calculated from a noisy point cloud. This type of noise will result in a different end effector pose relative to the object. 2) Noise in the object position. This kind of noise can be caused by placing the tool stand and screwdriver in a different position than the position previously observed in the demonstration. This type of noise can cause the estimated object center position to vary and will affect the visual servoing controller since the object and the robot end effector will look visually different from a different angle. In this experiment our goal is to find the region of attraction of the second and third aspects with respect to these two kinds of noise.

These two kinds of noise are artificially added to our experiment and the number of gradient descent iterations required to reach the ε-region of the aspect are recorded. In this experiment we only consider noise on the X-Y plane for easier visualization and analysis. For each type of noise and each aspect we tested 289 different combination of noise in the X and Y axes roughly within the scale that the visual servoing controller can handle. The results for adding noise in the relative pose between the robot hand and the object to the second aspect are shown in Figure 7. The plot on the left indicates how many iterations the visual servoing controller executed till convergence for different noise values. Each color tile is one single experiment and dark blue means the controller converges fast while orange means the controller took longer to converge. A yellow tile means that the controller could not converge within the 1000 iteration threshold. We call the region of attraction the set of observations that include the aspect plus the set of noise positions that corresponds to a non yellow tile connected to the origin. The plot on the right is a visualization of the same result in 3D which has some resemblance to the funnel metaphor used in Figure 1.

The results for adding noise in the relative pose between the robot hand and the object to the third aspect are shown in Figure 8. Note that this aspect has a smaller region of attraction with more tolerance in the direction perpendicular to the hand opening. If there is a large error in the Y axis the robot's hand may end up in front or behind the screwdriver. Under such situations without additional information the visual servoing controller will not be able to avoid colliding with the screwdriver while trying to reach the goal. The results for adding noise in the object position are shown in Figure 9. Notice that the regions of attraction are much larger for this type of noise.

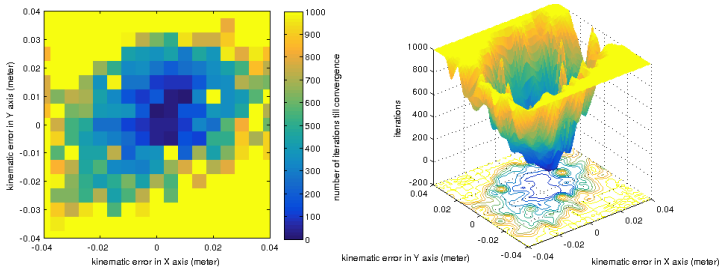

Fig. 7 Iteration till convergence with respect to noise in the relative pose between the robot hand and the object for the second aspect.

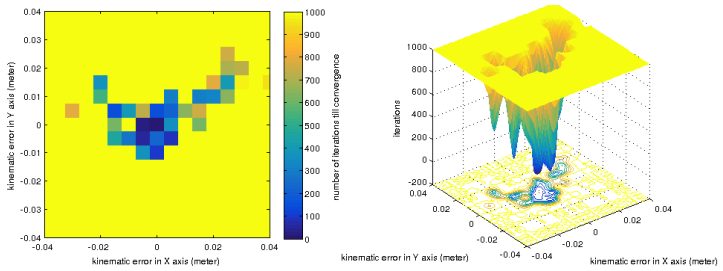

Fig. 8 Iteration till convergence with respect to noise in the relative pose between the robot hand and the object for the third aspect.

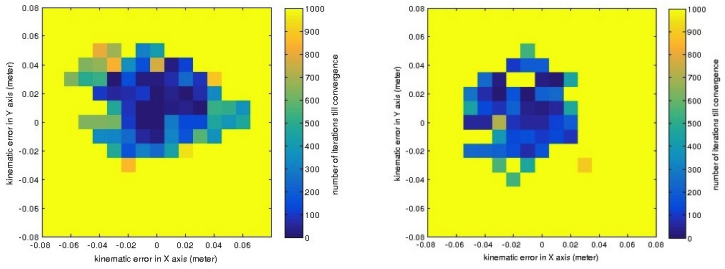

Fig. 9 Iteration till convergence with respect to noise in the object position for the second aspect (left image) and the third aspect (right image).

Convergence and Accuracy

By analyzing the observed regions of attraction of the visual servo controller that converges to the two aspects we can estimate the magnitude of noise this "slide-funnel-slide-funnel" controller sequence can tolerate. Through Figure 7 and Figure 8 we can see that the visual servo controller has a region of attraction with about 1.5 centimeter radius of kinematic noise around the second aspect and about 0.5 centimeter radius of kinematic noise around the third aspect. We evaluate these sequences of actions by comparing the final end effector position in the X-Y plane to

the demonstrated pose relative to the screwdriver. We tested noise of three different magnitudes to each open-loop action; 0.5, 1.0, and 1.5 centimeters for the action that transitions from the first aspect to the second aspect and 0.1, 0.2, and 0.3 centimeters for the action that transitions from the second aspect to the third aspect. For each combination of noise we test eight uniformly distributed directions. Among the 72 test cases 100% of them converged to the second aspect and 87.5% of them converged to the third aspect.

We did not reach a 100% overall convergence rate for two possible reasons. First, in addition to the artificial noise, randomness in the action planner and simulator also exist in the system. Second, the region of attractions shown in the previous section are estimated based on visual similarity. Two observations can be visually similar but position wise quite different therefore causing a false estimate of convergence. Figure 10 shows the test cases that the controller fails to converge on; most of the failed test cases are located in the lower right corner. This is consistent with the shape of the region of attraction of the controller with respect to the third aspect shown in Figure 8. The final poses of the end effector relative to the screwdriver are recorded and compared to the demonstrated pose.

We further compare the result to a sequence of "slide-slide" controllers without visual servoing acting as a funnel. The average position error is shown in Table 1. The "slide-funnel-slide-funnel" structure reduces the error by 55.8% and has an average error of 0.75 cm in the X-Y plane when only considering test cases that converged.

	complete test set	"slide-funnel-slide-funnel" structure converged test set
"slide-slide" structure	2.24 cm	2.06 cm
"slide-funnel-slide-funnel" structure	0.99 cm	0.75 cm

Table 1 Average position error in the X-Y plane in centimeters.

6 Conclusion

In this paper we introduce an image-based object model that categorizes different observations of an object into a subset of aspects based on interactions instead of only on visual appearance. We further propose that a sequence of controllers that form a "funnel-slide-funnel" structure based on this object model can have high rates of success even when open-loop controllers are within the sequence. To demonstrate this proposition we created an aspect transition graph object model that represents a pregrasp action through a teleoperation demonstration. In addition, we introduced a novel visual servoing controller that funnels the current observation to a memorized aspect using a changing set of visual features. The regions of attraction with respect to the end effector pose of the visual servoing controller are then identified by manually adding kinematic noise to the end effector position. Based on this region of attraction we identified the magnitude of kinematic noise this sequence of controllers is capable of handling and showed that under an environment

with a similar magnitude of noise this sequence of actions decreases the average final position error significantly.

The biggest drawback of the current approach is its scalability to model more complex objects. In this work we define aspects by manually indicating meaningful observations. In future work we plan to identify transitions autonomously and investigate hierarchical models that reuse exisiting sub-structures.

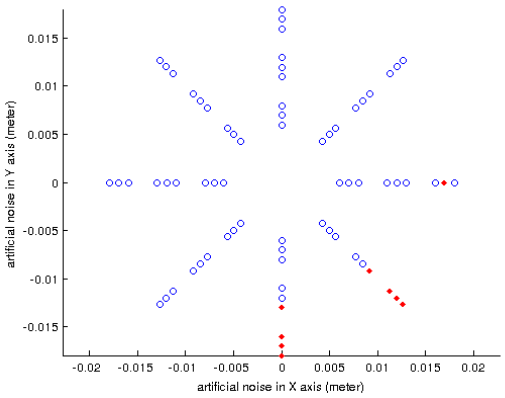

Fig. 10 Convergence with respect to artificial noise added to the test cases. Each dot represents a test case where the X Y value represents the summed magnitude and direction of the manually added kinematic noise. A red diamond indicates that the controller fails to converge to the third aspect while a blue circle indicates that the action sequence converged.

Acknowledgements The authors would like to thank Mitchell Hebert, Hee-Tae Jung, Michael Lanighan, Dirk Ruiken, Shiraj Sen, 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. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.
2. Heinrich H Bülthoff and Shimon Edelman. Psychophysical support for a two-dimensional view interpolation theory of object recognition. *Proceedings of the National Academy of Sciences*, 89(1):60–64, 1992.
3. Robert R Burridge, Alfred A Rizzi, and Daniel E Koditschek. Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6):534–555, 1999.
4. Amaury Dame and Eric Marchand. Entropy-based visual servoing. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 707–713. IEEE, 2009.
5. Paul Dinh and Stephen Hart. NASA Robonaut 2 Simulator, 2013. [Online; accessed 7-July-2014].
6. Shimon Edelman and Heinrich H Bülthoff. Orientation dependence in the recognition of familiar and novel views of three-dimensional objects. *Vision research*, 32(12):2385–2400, 1992.
7. Olivier Faugeras, Joe Mundy, Narendra Ahuja, Charles Dyer, Alex Pentland, Ramesh Jain, Katsushi Ikeuchi, and Kevin Bowyer. Why aspect graphs are not (yet) practical for computer vision. *CVGIP: Image Understanding*, 55(2):212–218, 1992.

8. James J Gibson. The perception of the visual world. 1950.
9. James J Gibson. Perceiving, acting, and knowing: Toward an ecological psychology. *chap. The Theory of Affordance). Michigan: Lawrence Erlbaum Associates*, 1977.
10. 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.
11. Stephen W Hart. *The development of hierarchical knowledge in robot systems*. PhD thesis, University of Massachusetts Amherst, 2009.
12. Frank Hoffmann, Thomas Nierobisch, Torsten Seyffarth, and Günter Rudolph. Visual servoing with moments of sift features. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, volume 5, pages 4262–4267. IEEE, 2006.
13. Manfred Huber. A hybrid architecture for adaptive robot control. 2000.
14. Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *Robotics and Automation, IEEE Transactions on*, 12(5):651–670, 1996.
15. Martin Jägersand and Randal Nelson. On-line estimation of visual-motor models using active vision. *image*, 11:1, 1996.
16. 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.
17. David J Kriegman and Jean Ponce. Computing exact aspect graphs of curved objects: Solids of revolution. *International Journal of Computer Vision*, 5(2):119–135, 1990.
18. 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.
19. 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.
20. Nikos K Logothetis, Jon Pauls, and Tomaso Poggio. Shape representation in the inferior temporal cortex of monkeys. *Current Biology*, 5(5):552–563, 1995.
21. Jorge J More and John Arthur Trangenstein. On the global convergence of broydens method. *Mathematics of Computation*, 30(135):523–540, 1976.
22. Yoshihiko Nakamura. Advanced robotics: redundancy and optimization, 1991.
23. Thomas J Palmeri and Isabel Gauthier. Visual object understanding. *Nature Reviews Neuroscience*, 5(4):291–303, 2004.
24. Tomaso Poggio and Shimon Edelman. A network that learns to recognize 3d objects. *Nature*, 343(6255):263–266, 1990.
25. Shiraj Sen. *Bridging the gap between autonomous skill learning and task-specific planning*. PhD thesis, University of Massachusetts Amherst, 2013.
26. Azad Shademan and Farrokh Janabi-Sharifi. Using scale-invariant feature points in visual servoing. In *Optics East*, pages 63–70. International Society for Optics and Photonics, 2004.
27. Alexander Stoytchev. Toward learning the binding affordances of objects: A behavior-grounded approach. In *Proceedings of AAAI Symposium on Developmental Robotics*, pages 17–22, 2005.
28. Ioan A. Sucan and Sachin Chitta. Moveit! [Online].
29. Michael J Tarr and Heinrich H Bülthoff. Image-based object recognition in man, monkey and machine. *Cognition*, 67(1):1–20, 1998.
30. Russ Tedrake. Lqr-trees: Feedback motion planning on sparse randomized trees. 2009.
31. Shimon Ullman and Ronen Basri. Recognition by linear combinations of models. *IEEE transactions on pattern analysis and machine intelligence*, 13(10):992–1006, 1991.
32. Esther Van Den Bos and Marc Jeannerod. Sense of body and sense of action both contribute to self-recognition. *Cognition*, 85(2):177–187, 2002.
33. Karthik Mahesh Varadarajan and Markus Vincze. Afrob: The affordance network ontology for robots. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1343–1350. IEEE, 2012.