A Visual Analogy Approach to Source Case Retrieval in Robot Learning from Observation

Tesca Fitzgerald, Keith McGreggor, Baris Akgun, Ashok K. Goel, Andrea L. Thomaz
School of Interactive Computing
Georgia Institute of Technology, Atlanta, GA 30332, USA
{tesca.fitzgerald, keith.mcgreggor}@gatech.edu, {bakgun3, goel, athomaz}@cc.gatech.edu

Abstract

Learning by observation is an important goal in developing complete intelligent robots that learn interactively. We present a visual analogy approach toward an integrated, intelligent system capable of learning skills from observation. In particular, we focus on the task of retrieving a previously acquired case similar to a new, observed skill. We describe three approaches to case retrieval: feature matching, feature transformation, and fractal analogy. SIFT features and fractal encoding were used to represent the visual state prior to the skill demonstration, the final state after the skill has been executed, and the visual transformation between the two states. We discovered that the three methods (feature matching, feature transformation, and fractal analogy) are useful for retrieval of similar skill cases under different conditions pertaining to the observed skills.

Introduction

Learning is an ability that lies at the core of AI research. Complete intelligent robots in the future must be able to learn quickly such that they can address problems within novel environments. We focus on robots that interact with end-users via Learning from Demonstration (LfD), which is an approach aimed at allowing human teachers to demonstrate new skills to a robot without requiring explicit programming (Argall et al. 2009). There have been decades of research in this domain, and a wide variety of approaches to extracting skill models from observation of a human performance (e.g., (Ijspeert, Nakanishi, and Schaal 2002; Nakanishi et al. 2004; Calinon and Billard 2007; Kuniyoshi, Inaba, and Inoue 1994; Deisenroth, Rasmussen, and Fox 2011; Jenkins and Matarić 2002)). The work we present here is not focused on how to extract a skill model from these observations, but instead on allowing the robot to use visual analogy to find the most similar cases of a demonstrated skill.

This kind of skill transfer is necessary to develop complete intelligent robots, and allows adaption of existing skill models rather than requiring that the robot learns new ones from scratch in a novel context. For example, if the robot has been taught to close a box (the "source" problem), it could learn to close a book (the "target" problem) by transferring its source knowledge to the new problem. Recalling an appropriate source case is therefore the first step necessary to transfer knowledge from a library of past (source) cases to a new, target problem (Kolodner 1992; Gick and Holyoak 1980; Gentner 1983).

Background, Motivation, and Goals

We seek to use visual case-based reasoning (CBR) (Davies, Glasgow, and Kuo 2006; Davies, Goel, and Yaner 2008; Perner 1999; Perner, Holt, and Richter 2005; Yaner and Goel 2006) to allow robots to relate new observations, such as those shown in Figure 1, to previously learned skills. We focus on using visual CBR to identify skill demonstrations that are provided in a controlled environment, as shown in Figure 1. These images are collected via an overhead camera located in front of the robot learner and above the tabletop workspace.

We use the method of fractal analogy (McGreggor and Goel 2012) because it allows automatic adjustment of the level of spatial resolution for evaluating similarity between two images; rather than encode the features detected within visual scenes, this method encodes the transformation between two visual scenes. One contribution of this paper is evaluating the fractal analogy method on a robot's visual observations during a real-life skill demonstration.

We also present two methods based on extracting SIFT features from the visual scene perceived by the robot (Lowe 1999, 2004). First, we use the SIFT algorithm in a way that is standard in computer vision for image classification problems. Due to the fractal analogy method's emphasis on visual transformations, we introduce a second SIFT method based on transformations of SIFT image features to evaluate the similarity between two skill demonstrations. The question we address in this paper is the extent to which each of these can address the task of source skill demonstration retrieval using a robot's visual observations.

We first formulate the problem of visual analogy for LfD. Then, we detail the three algorithms for solving this problem. We conduct experiments comparing the ability of all three methods to accurately retrieve analogous skills given a new skill demonstration, and provide a detailed analysis of cases in which each method performed better than the others.

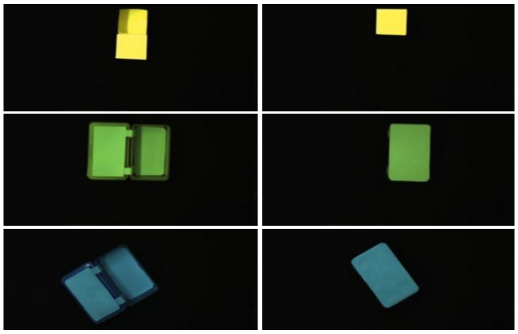


Figure 1: Analogical Box-Closing Skill Demonstrations

Approach: Skill Retrieval as Visual Analogy

Source retrieval for LfD demonstrations is a difficult problem for several reasons. A single skill can be demonstrated using a variety of objects, e.g., pouring using a tablespoon or cup, or a box-closing skill can be demonstrated with different sizes, colors, or shapes of boxes. Figure 1 illustrates three analogous demonstrations of a box-closing skill. A single object may be used to demonstrate multiple skills (e.g., opening and closing skills both demonstrated on the same box). Additionally, different teachers may demonstrate the same skill in various manners. While not addressed in this work, additional interference such as poor lighting or messy workspace add further complexities to this problem.

We define a skill demonstration as the combination of two distinct transformations:

- The *kinesthetic transformation* T_k such that when the physical action T_k is executed, the skill is completed and the goal state is achieved.
- The set of *visual transformations* T_v (the observable state change) such that when T_k is performed, T_v is observed.

We assume there is a correspondence between T_k and T_v, and skills with analogous visual transformations will require analogous kinesthetic transformations as well. For the purpose of this paper we only focus on the retrieval of an analogous skill demonstration from memory, given a novel skill demonstration.

We define a visual transformation as the tuple

$$< S_i, S_f, T >$$

where S_i is the observed initial state (the left-side images in Figure 1), S_f is the observed goal state that is reached following the skill completion (the right-side images in Figure 1), and T is the transformation between the two images S_i and S_f. The transformation T is not dependent on any particular features of either state observation. Rather, the relationship between the two images dictates the transformation T. Two visual transformations are determined to be analogous if their transformations T are similar. By encoding the transformation between initial and final state images, only the visual change resulting from the demonstration is encoded, rather than the specific approach used to execute the demonstration. We detail later exactly how each of the three approaches define the visual state, but importantly, all our methods are reasoning on pixel level features, not using any form of object-based segmentation or symbolic feature classification.

Analogical problems are commonly written in the form $A : B :: C : ?$ where each letter represents an element of the analogy, and denotes that the transformation from element A to element B is analogous to the transformation between element C and the unknown element. In the case of LfD skill retrieval, given one skill demonstration where A is the image depicting the initial state S_i and B is the image depicting the final state S_f, T is the visual transformation such that $A : B$ holds. The task for source demonstration retrieval is to find $S_{0i} : S_{0f} :: S_{1i} : S_{1f}$ by determining the similarity between the relation $S_{0i} : S_{0f}$ (represented by T_0) and $S_{1i} : S_{1f}$ (represented by T_1). In the next section, we detail three algorithms for solving this task.

Algorithms

Baseline: SIFT Feature-Counting

The Scale-Invariant Feature Transform (SIFT) algorithm selects keypoint features that can be identified regardless of the image's scale, translation, or rotation (Lowe 2004), using the following steps (Lowe 1999). First, candidate keypoints are chosen. These candidates are selected as interest points with high visual variation. Candidate keypoints are tested to determine their robustness to visual changes (i.e., illumination, rotation, scale, and noise). Keypoints deemed "unstable" are removed from the candidate set. Each keypoint is then assigned an orientation invariant to the image's orientation. Once each keypoint has been assigned a location, scale, and orientation, a descriptor is allocated to each keypoint, representing it in the context of the local image.

Our first approach to source demonstration retrieval using SIFT features is based on feature-counting. The target skill demonstration is represented by the image pair (S_i, S_f). Using the SIFT algorithm, features are extracted from each image and matched to features from the initial and final states of source skill demonstrations. Each feature consists of the 16x16 pixel area surrounding the feature keypoint. A brute-force method is used to determine that two features match if they have the most similar 16x16 surrounding area. The source demonstration sharing the most features with the target demonstration is retrieved using the following process:

Let D be a set of source skill demonstration images
$c \leftarrow null$; $s \leftarrow 0$
$U_i \leftarrow$ SIFT features extracted from S_i
$U_f \leftarrow$ SIFT features extracted from S_f
for each demonstration $d \in D$ **do**
    $C_i \leftarrow$ SIFT features extracted from d_i
    $C_f \leftarrow$ SIFT features extracted from d_f
    $T \leftarrow (U_i \cap C_i) \cup (U_f \cap C_f)$
    If $size(T) > s$, then: $s \leftarrow size(T)$, $c \leftarrow d$
end for
return c

Figure 4(e) illustrates a retrieval result, where the left-side image is S_i and the right-side image is the d_i selected with the highest number of matching SIFT features.

SIFT Feature-Transformation

Our second approach to source demonstration retrieval via the SIFT algorithm focuses on the transformation of SIFT features between a demonstration's initial and final states. Rather than retrieve a source demonstration based on the explicit features it shares with the target demonstration, this approach retrieves a source demonstration according to the similarities between its feature transformations and those of the transformations observed in the target demonstration.

Each feature of the demonstration's S_i is matched to its corresponding feature in the S_f, as shown in Figure 4(b). This method uses the same features and feature-matching method as in the SIFT feature-counting approach described previously. We define each SIFT feature transformation as the tuple

$$<< s_x, s_y >, \theta, l >$$

where s_x and s_y are the coordinates of the feature in the initial state, θ is the angular difference between the feature in the initial state and final state, and l is the distance between the feature location in the initial and end state images. Each feature transformation occurring between S_i and S_f in the target demonstration is compared to each transformation occurring between S_i and S_f in each of the source skill demonstrations. The difference between two SIFT feature transformations is calculated by weighting the transformations' source location change, angular difference, and distance.

Each comparison is performed over seven abstraction levels, at each a normalized box filter kernel is used to blur the target and source demonstrations' visual states, with the kernel size increasing by a factor of two at each level. This serves to reduce the number of "noisy" features that are irrelevant to the skill being performed. The SIFT feature-transformation method retrieves a source demonstration as follows:

Let D be a set of source skill demonstration images
$c \leftarrow null$; $s \leftarrow 0$; $x \leftarrow 0$
for each demonstration $d \in D$ **do**
    $n \leftarrow 0$
    while $n <$ maximum abstraction level **do**
        $x \leftarrow 0$
        Blur S_i, S_f, d_i, and d_f by a factor of 2^n
        $U_i \leftarrow$ SIFT features extracted from S_i
        $U_f \leftarrow$ SIFT features extracted from S_f
        $T_u \leftarrow getTransformations(U_i \cap U_f)$
        $C_i \leftarrow$ SIFT features extracted from d_i
        $C_f \leftarrow$ SIFT features extracted from d_f
        $T_c \leftarrow getTransformations(C_i \cap C_f)$
        for each transformation $t_u \in T_u$ **do**
            Find $t_c \in T_c$ that minimizes $diff(t_u, t_c)$
            $x \leftarrow x + diff(t_u, t_c)$
        end for
        If $x < s$, then: $c \leftarrow d$, $s \leftarrow x$
        $n \leftarrow n + 1$
    end while
end for
return c

Fractal Analogy

Our final approach uses fractals, which encode self-similarity between images, to represent the visual transformation function T between two images (McGreggor and Goel 2012), and is expressed as the set of operations that occur to transform the initial state image S_i into the final state image S_f. The similarity between two image transformations can be determined using the *ratio model*:

$$sim(T, T') = f(T \cap T') / f(T \cup T')$$

In this model, T encodes the first set of image transformations, T' encodes the second set of image transformations, and $f(x)$ returns the number of features in the set x (Tversky 1977), (McGreggor, Kunda, and Goel 2010). Thus, $f(T \cap T')$ returns the number of transformations common to both transformation sets, and $f(T \cup T')$ returns the number of transformations in either set. The following process encodes a visual transformation as a fractal (McGreggor, Kunda, and Goel 2010):

1. The initial state image is segmented into a grid containing a specified number of partitions, $P = \{p_0, p_1, ..., p_n\}$, where n is defined by the abstraction level.
2. For each sub-image $p \in P$, the destination image is searched for a sub-image s such that for a transformation k, $k(p)$ is most similar to s.
3. The transformation k and shift c, the mean color-shift between s and $k(p)$, are used to create a code c_p.
4. The resulting fractal is defined by $F = \{c_0, c_1, ..., c_n\}$

This encoding process is repeated for multiple values of n, resulting in an encoding of the transformation at seven levels of abstraction. A code is defined by the tuple

$$<< s_x, s_y >, < d_x, d_y >, k, c >$$

where s_x and s_y are the coordinates of the source sub-image, d_x and d_y are the coordinates of the destination sub-image, c is the mean color-shift between the two sub-images, and k represents the affine transformation between the source and destination sub-images where $k \in \{$ 90° clockwise rotation, 180° rotation, 270° clockwise rotation, horizontal reflection (HR), vertical reflection (VR), identity (I) $\}$. The transformation k referenced in step 2 is one of such transformations, such that sub-image s is converted into sub-image p minimally, while requiring minimal color changes.

A set of fractal features is derived as combinations of different aspects of each fractal code. While the fractal code does describe the transformation from a section of a source image into a target image, the analogical matching occurs on a much more robust set of features than merely the transformation taken by itself. The illustrations which visualize the fractal representation therefore demonstrate only those transformations, and not the features.

Experiment

These three algorithms were used to retrieve a source skill demonstration for three test sets of target demonstrations. Each skill demonstration is a pair of two recorded keyframe

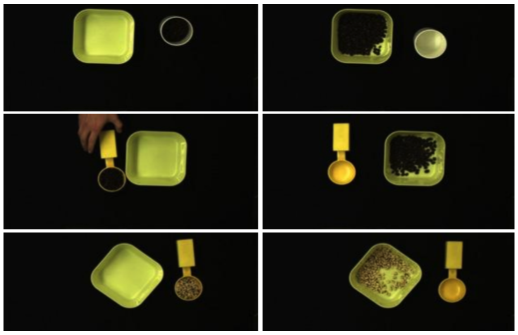


Figure 2: Analogical Pouring Skill Demonstrations

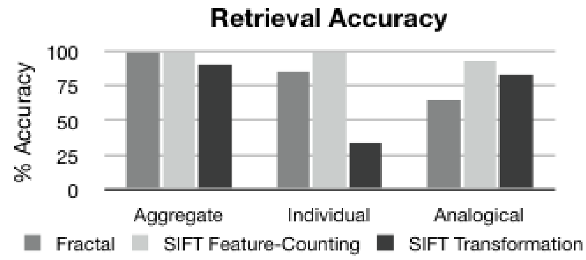


Figure 3: Source Demonstration Retrieval Test Results

images depicting the initial state and end state of a box-closing or cup-pouring skill performed by a human participant, as shown in Figures 1 and 2. Nine participants demonstrated the two skills, and were recorded using an overhead camera above the tabletop workspace. Participants indicated the initial, final, and intermediary keyframe states verbally, and removed their hands from view when the initial and final state images were recorded. Each participant's demonstration set consisted of nine demonstrations per skill, with each skill being performed using three different objects at three orientations as shown in Figures 1 and 2.

We evaluated the algorithms on three test sets. In the *aggregate* set, a source demonstration is retrieved for two participants' demonstrations (a total of 12 target demonstrations) from a library of 48 source demonstrations, which included 24 demonstrations of each skill. All box-closing and pouring demonstrations used the same box and pouring objects, respectively.

In the *individual* set, a source skill demonstration was retrieved for each of 54 target demonstrations (27 per skill). Within each participant's demonstration set, the target demonstration was compared to the other demonstrations by the same participant. As a result, a source was retrieved for each target demonstration from a library containing two source demonstrations of the same skill and three of the opposite skill. As in the aggregate test set, demonstrations used the same box and pouring objects.

In the *analogical* set, a source demonstration was retrieved for each of 161 target demonstrations (80 box-closing, 81 pouring). Within each participant's demonstration set, the target demonstration was compared only to other demonstrations performed by the same participant. Unlike the previous test sets, target demonstrations were compared to source demonstrations involving different objects, as in Figures 1 and 2. A source demonstration was retrieved for each target demonstration from a library containing six source demonstrations of the same skill and nine of the opposite skill. One box-closing demonstration was incomplete and could not be included in the test set; as a result, 17 target demonstrations were compared to one fewer box-closing demonstration. The purpose of the analogical test set was to test each retrieval method's ability to retrieve a source skill demonstration, despite containing a different set of objects than the target demonstration.

Experimental Results

Figure 3 illustrates the overall accuracy of each method's ability to retrieve a correct source skill demonstration when applied to each test set. Since the aggregate test contained a large set of source demonstrations and was most likely to contain a demonstration similar to the target problem, we expected that this test set would be the easiest test set for any of the three methods to address. The fractal transformation, SIFT feature counting, and SIFT feature transformation methods retrieved an appropriate source demonstration correctly for 100%, 100%, and 91.7% of the target demonstrations, respectively. When addressing the individual test set, which contained fewer source demonstrations, the fractal transformation, SIFT feature counting, and SIFT feature transformation methods retrieved an appropriate source demonstration correctly for 87%, 100%, and 35.2% of the target demonstrations, respectively. In the analogical test set, the fractal transformation, SIFT feature-counting, and SIFT feature transformation methods retrieved an appropriate source demonstration correctly for 65.3%, 93.8%, and 84.5%, respectively, of the target demonstrations included in the analogical test set. Overall, the SIFT feature-counting method yielded the highest accuracy.

Detailed Analysis

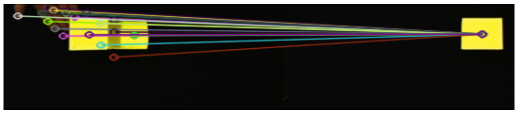
While the experimental results provide useful feedback about the accuracy of each retrieval method, we chose to further analyze three source demonstration retrieval problems, illustrating the strengths of each retrieval method.

Case Study: Fractal Analogy Success

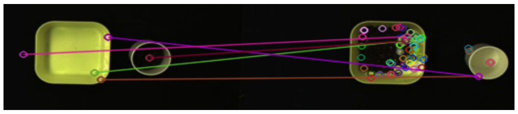
First, we analyze an example in which only the fractal analogy method retrieved an appropriate source demonstration. Figure 4(a) depicts the target problem demonstration, which the fractal analogy method correctly matched to the source demonstration shown in Figure 4(d). The fractal method offers both a decreased susceptibility to noise as well as a plethora of fractal features over which to calculate a potential match (beyond the transformation itself).

The SIFT feature-matching method incorrectly classified Figure 4(a) as a pouring skill demonstration, due to the many features matched between the target demonstration and pouring demonstration's final states. Features of the demonstrator's hand were incorrectly matched to features of the pouring instrument, as shown in Figure 4(e).

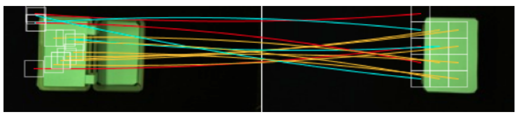
(a) Target Problem



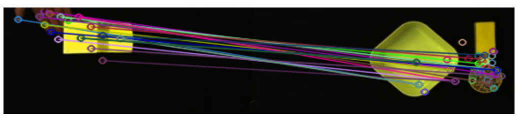
(b) SIFT Feature-Transformation Representation



(c) SIFT Feature-Transformation Result



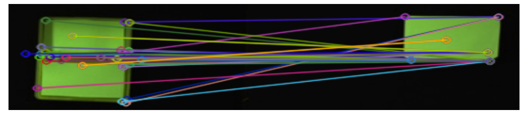
(d) Fractal Method Result



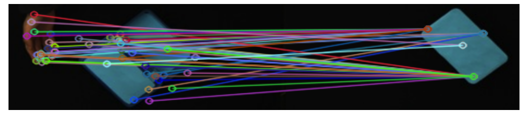
(e) SIFT Feature-Matching Result

Figure 4: Case Study 1: Retrieval Method Results

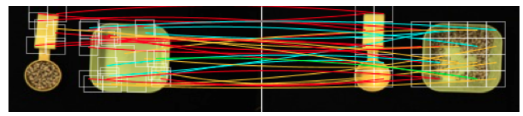
(a) Target Problem



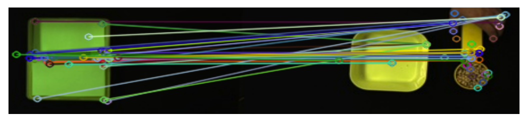
(b) SIFT Feature-Transformation Representation



(c) SIFT Feature-Transformation Result



(d) Fractal Method Result



(e) SIFT Feature-Matching Result

Figure 5: Case Study 2: Retrieval Method Results

The SIFT feature-transformation method also incorrectly classified the demonstration as a pouring skill demonstration. Figure 4(b) illustrates the feature transformations used to represent the target problem. Each feature in the initial state was matched to the single feature identified in the final state. Thus, the resulting feature transformations did not properly represent the skill being performed, which led to the retrieval of an incorrect source demonstration (see Figure 4(c)).

We can conclude that the fractal method can be applied to source retrieval problems in which the visual transformation, rather than keypoint features, are indicative of the skill being performed. The fractal method is also applicable to demonstrations that include "noise", such as the demonstrator's hand or other objects unrelated to the skill being performed. Additionally, we conclude that the feature-matching method is sensitive to noise, and that the feature-transformation method is less effective in classifying demonstrations in which there are few features in the initial or final state, or in which there is a poor correspondence between features matched between the initial and final state images.

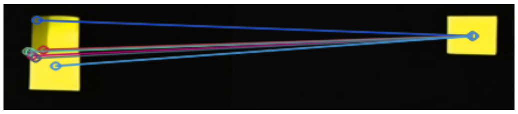
Case Study: SIFT Transformation Success

In the next case, only the SIFT feature-transformation method retrieved an appropriate source demonstration for the target problem shown in Figure 5(a). The SIFT feature transformation method retrieves visually analogical source demonstrations by identifying visual transformations at multiple abstraction levels. The transformations in Figure5(c) were deemed similar to those in the target problem. Features in the initial and final states were matched correctly, which is why this method was able to succeed.

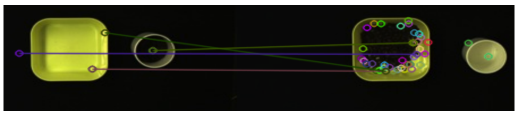
The fractal method incorrectly retrieved the source demonstration shown in Figure 5(d) due to its emphasis on visual transformations independent of features, and thus is less effective in distinguishing between skills that have similar visual transformations. The more similar the visual transformations, the more common and therefore the less salient are the fractal method's generated features derived from those transformations. The fractal method chose this source demonstration due to the similarity between the movement of the box lid from one part of the target demonstration image to another, and the movement of coffee beans from one part of the source demonstration image to another. The SIFT feature-matching method also returned an incorrect source demonstration in this case, as it erroneously matched features of the target demonstration's initial state to features of a pouring instrument (see Figure 5(e)).

This case study teaches us that the feature-transformation method is best applied to situations in which there are a large number of features in both the initial and final state images, and the two sets of features have been mapped correctly. Additionally, we find that the fractal method is less effective in distinguishing between skills that have similar visual transformations. Finally, this case study demonstrates how the feature-matching method relies on having a correct mapping between features of the target demonstration and features extracted from a potential source demonstration.

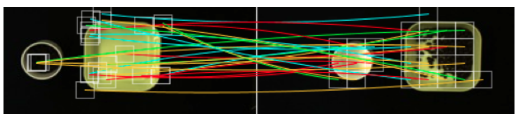
(a) Target Problem



(b) SIFT Feature-Transformation Representation



(c) SIFT Feature-Transformation Result



(d) Fractal Method Result

(e) SIFT Feature-Matching Result

Figure 6: Case Study 3: Retrieval Method Results

Case Study: SIFT Feature-Matching Success

In the final case, only the feature-matching method retrieved the correct source demonstration to address the target problem shown in Figure 6(a). This method correctly matches features between the target problem and source demonstration's initial and final state features. The initial state feature mapping is shown in Figure 6(e). Just as in the first case study example, the feature-transformation method does not retrieve the correct source demonstration because there are not enough features in the final state image. All features in the source demonstration's initial state are mapped to the single feature in the final state image, causing the feature transformations to poorly reflect the skill being performed. The fractal method retrieves an incorrect source demonstration due to its emphasis on the visual transformation between the two states, without any weight to the objects being moved. It determined the movement of the box lid to be analogical to the movement of coffee beans from the left side of the image to the right side as shown in Figure 6(d).

Thus, the feature-matching method is most effective when there is a correct correspondence between features of the target problem and matching features in the potential source demonstration, and there are enough features in both demonstrations to represent the objects being used. As it turns out, even our analogical test set used objects that were similar enough for feature-matching to achieve the highest success rate (e.g., even after switching from pouring coffee beans to white beans, black flecks made them look enough like coffee beans to match). We expect that for analogical images with less object feature correspondence, this result would dramatically change. This case study also demonstrates that the fractal method is less effective in cases where different skills entail similar visual transformations. Finally, this example shows how the feature-transformation method is less effective in classifying demonstrations in which there are few features in the initial or final state, or there is a poor correspondence between features matched between the initial and final state images.

Conclusions

A complete intelligent robot must be able to relate new observations to already-known skills. Retrieving an appropriate source skill demonstration could then aid the robot in learning new skills while requiring fewer demonstrations, as its knowledge of analogous skills could inform its representation of the new skill. As a first step toward enabling this, we have presented three approaches to analogical source skill retrieval. The SIFT feature-matching method represents well-known methods explored in the AI research field, while the fractal method is new to both AI and robotics. The SIFT feature-transformation method represents an intermediate method, combining the feature-identification of the SIFT algorithm with the visual transformations emphasized in fractal analogy reasoning.

Several variables may affect the accuracy of each skill classification method. The fractal method's accuracy improves when each skill can be represented by a distinct visual transformation. Additionally, this method is affected by the heuristic used to select the abstraction level at which two demonstrations should be compared. We currently use the heuristic of summing the similarity scores that are calculated at multiple abstraction levels. However, this heuristic may negatively impact the fractal method's overall accuracy if skill types are most accurately classified at a certain abstraction level. Additionally, the SIFT feature-transformation method is affected by the scoring function used to determine the similarity of two transformations. The weight values applied to the angular difference, change in transformation distance, and change in start location between two feature transformations will impact how accurately the method can determine the similarity between visual feature transformations. These two variables, the abstraction-level selection heuristic and the transformation similarity metric, may become the focus of future work.

No single method works best for all skill demonstration retrieval problems. Rather, each method discussed in this paper is best suited for a particular type of problem. In particular, the SIFT feature-matching method is best suited for retrieval problems in which enough visual features can be extracted to identify the skill, and little noise is present. The SIFT feature-transformation method is most effective in problems where many features can be extracted from the demonstrations, and correspondences between features can be identified correctly. Finally, the fractal analogy method is most effective in identifying skills in which the visual transformation should be emphasized, rather than features of the demonstration images themselves.

Acknowledgments

The authors wish to thank Dr. Maithilee Kunda for many valuable discussions and for her feedback. This work was supported by ONR Grant #N000141410120 and NSF IIS Grant #1116541.

References

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57(5):469–483.

Calinon, S., and Billard, A. 2007. Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, 255–262.

Davies, J.; Glasgow, J.; and Kuo, T. 2006. Visio-spatial case-based reasoning: A case study in prediction of protein structure. *Computational Intelligence* 22(3-4):194–207.

Davies, J.; Goel, A. K.; and Yaner, P. W. 2008. Proteus: Visuospatial analogy in problem-solving. *Knowledge-Based Systems* 21(7):636–654.

Deisenroth, M. P.; Rasmussen, C. E.; and Fox, D. 2011. Learning to control a low-cost manipulator using data-efficient reinforcement learning. In *Robotics: Science & Systems (RSS)*.

Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science* 7(2):155–170.

Gick, M. L., and Holyoak, K. J. 1980. Analogical problem solving. *Cognitive Psychology* 12(3):306–355.

Ijspeert, A. J.; Nakanishi, J.; and Schaal, S. 2002. Learning rhythmic movements by demonstration using nonlinear oscillators. In *Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS '02)*.

Jenkins, O. C., and Matarić, M. 2002. Deriving action and behavior primitives from human motion data. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, 2551–2556.

Kolodner, J. L. 1992. An introduction to case-based reasoning. *Artificial Intelligence Review* 6(1):3–34.

Kuniyoshi, Y.; Inaba, M.; and Inoue, H. 1994. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. In *IEEE Transactions on Robotics and Automation*, volume 10, 799–822.

Lowe, D. G. 1999. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, 1150–1157. IEEE.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2):91–110.

McGreggor, K., and Goel, A. 2012. Fractal analogies for general intelligence. In *Artificial General Intelligence*. Springer. 177–188.

McGreggor, K.; Kunda, M.; and Goel, A. 2010. A fractal approach towards visual analogy. In *Proceedings of the International Conference on Computational Creativity, Lisbon, Portugal, January 9Y11*.

Nakanishi, J.; Morimoto, J.; Endo, G.; Cheng, G.; Schaal, S.; and Kawato, M. 2004. Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems* 47:79–91.

Perner, P.; Holt, A.; and Richter, M. 2005. Image processing in case-based reasoning. *Knowledge Engineering Review* 20(3):311–314.

Perner, P. 1999. An architecture for a CBR image segmentation system. *Engineering Applications of Artificial Intelligence* 12(6):749–759.

Tversky, A. 1977. Features of similarity. *Psychological Review* 84(4):327.

Yaner, P. W., and Goel, A. K. 2006. Visual analogy: Viewing analogical retrieval and mapping as constraint satisfaction problems. *Applied Intelligence* 25(1):91–105.