Visual Case Retrieval for Interpreting Skill Demonstrations

Tesca Fitzgerald(✉), Keith McGreggor, Baris Akgun, Andrea Thomaz, and Ashok Goel

School of Interactive Computing, Georgia Institute of Technology, 30332 Atlanta, Georgia
{tesca.fitzgerald,bakgun3,athomaz,goel}@cc.gatech.edu, keith.mcgreggor@venturelab.gatech.edu

Abstract. Imitation is a well known method for learning. Case-based reasoning is an important paradigm for imitation learning; thus, case retrieval is a necessary step in case-based interpretation of skill demonstrations. In the context of a case-based robot that learns by imitation, each case may represent a demonstration of a skill that a robot has previously observed. Before it may reuse a familiar, *source* skill demonstration to address a new, *target* problem, the robot must first retrieve from its case memory the most relevant source skill demonstration. We describe three techniques for visual case retrieval in this context: feature matching, feature transformation matching, and feature transformation matching using fractal representations. We found that each method enables visual case retrieval under a different set of conditions pertaining to the nature of the skill demonstration.

Keywords: Visual case retrieval · Case-based agents · Imitation learning

1 Introduction

Learning by imitation is a well-researched methodology, both in human cognition and in cognitive robotics [2,18,26]. Robot learning by demonstration is an approach which aims to enable imitation by having the robot receive a demonstration of a skill from a human teacher. The robot perceives the workspace and objects involved in completing the skill during the demonstration, while also recording the actions required to complete the skill. At a later time, the robot may be asked to repeat the learned skill in the same or in a new workspace.

Case-based reasoning is an important paradigm for learning by imitation (e.g. [6,7]). In the case-based approach to imitation, the robot would (i) store the observed skill demonstrations as cases in a case memory, (ii) given a new, related problem, retrieve the most similar case from the case memory, (iii) adapt the demonstrated actions from the retrieved case to the new problem, and (iv) execute the adapted actions to address the new problem. We refer to the first two steps

E. Hüllermeier and M. Minor (Eds.): ICCBR 2015, LNAI 9343, pp. 119–133, 2015.
DOI: 10.1007/978-3-319-24586-7_9

of this approach as *skill demonstration interpretation*. Note that a necessary step in skill demonstration interpretation is for the robot to recall the skill demonstration most similar to the current configuration of objects. Thus, in this paper, we focus *solely on this task of case retrieval* to enable case-based interpretation of skill demonstrations in the context of interactive robot learning by imitation. The goal of case retrieval in this context is to return a source case demonstrating the same skill as shown in a new, uncategorized skill demonstration.

A critical question in case-based interpretation is that of case representation. A case of a previously observed skill should be represented such that, given a new skill demonstration, it is feasible for the robot to recognize the similarity between the two. In the rest of this paper, we make the following contributions:

1. Propose three visual representations for skill demonstration cases, with corresponding source case retrieval algorithms.
2. Present experiments testing each representation on skill demonstrations provided in a table-top environment.
3. Test the effectiveness of Fractal reasoning on real-world images perceived during skill demonstrations.
4. Compare the efficacy of the three case retrieval methods by providing an analysis of situations in which each method performs better than the others.

2 Background

Case-based reasoning is a cognitively inspired paradigm for reasoning and learning [1,11–13,22,23]; Thagard [25] views case-based reasoning as a paradigm for modeling human cognition. In case-based reasoning, new problems are addressed by retrieving and adapting solutions to similar problems stored as cases in a case memory. In case-based reasoning, (a) learning is incremental, (b) learning is problem-specific in that the robot adapts the most similar case to address the current problem, and (c) learning is lazy, meaning that the robot learns the abstraction only when needed.

Ontanon et al. [19] studied case-based learning from demonstration in the context of online case-based planning in real-time strategy games. While an important domain for case-based reasoning, games do not offer the low-level challenges of perception and action to the same degree that interactive robots immediately pose. Floyd, Esfandiari and Lam [7] describe a case-based method for learning soccer team skills by observing spatially distributed soccer team plays. Ros et al. [24] present a case-based approach to action selection in robot soccer. More recently, Floyd and Esfandiari [6] describe a preliminary scheme for separating domain-independent case-based learning by observation from domain-dependent sensors and effectors on a physical robot.

We seek to use visual case-based reasoning to recognize that a new target demonstration, such as the overhead view of a box-closing skill shown in the top row of Fig. 1, is similar to skill demonstrations previously stored in the robot's memory, such as the related box-closing demonstrations shown in the bottom two

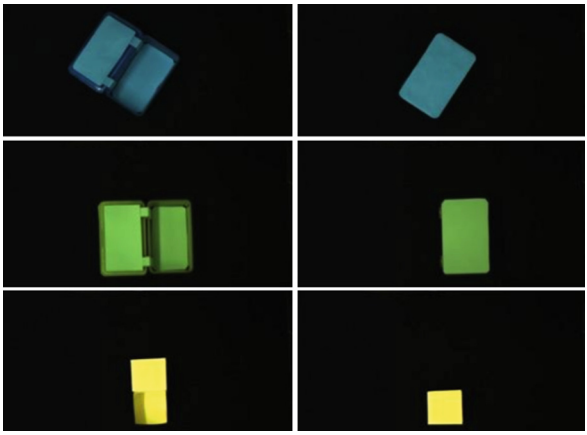


Fig. 1. Similar box-closing skill demonstrations

rows of Fig. 1. Visual case-based reasoning has been previously studied in tasks ranging from interpreting line drawings [5,28] to image interpretation [8,20] in domains ranging from molecular biology [4,10] to design [3,9,28]. Perner, Hold and Richter [21] provide a review of some of these applications. Techniques for visual case retrieval in these applications range from heuristic [9] to graph matching [5] to constraint satisfaction [28]. Images in these applications typically are static and often discrete (e.g., in the form of line drawings). In contrast, images in case-based interpretation of skill demonstrations are dynamic and continuous, requiring the development of new techniques for visual case retrieval.

The first column of Fig. 1 depicts the observed initial states of three demonstrations of the same skill, and the second column depicts the corresponding final states. As Fig. 1 illustrates, our current focus is on case-based interpretation of skill demonstrations in a table-top learning environment. Our aim is to first develop approaches for case-based interpretation, leaving the task of perception in cluttered, occluded, messy, or poorly-lit environments to future work.

We first approach the problem of case-based skill interpretation using a Fractal representation [17]. Instead of encoding the features detected within visual scenes, the Fractal method encodes the visual transformations between initial and final states of a skill demonstration. We wanted to use the Fractal method because it allows automatic adjustment of the level of spatial resolution for evaluating similarity between two sets of images. While the Fractal method has been applied to geometric analogies on intelligence tests, it has not yet been applied to real-world images such as those a robot would perceive. To fully evaluate the Fractal method for case-based interpretation, we chose to compare it to a baseline method which uses the Scale-Invariant Feature Transform (SIFT) algorithm to select image features. The SIFT algorithm identifies features regardless of the image's scale, translation, or rotation [14,15] and is widely used for computer vision tasks in robotics research.

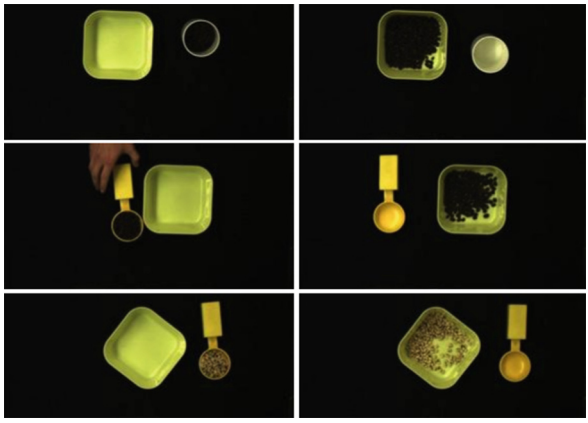


Fig. 2. Analogical pouring skill demonstrations

3 Problem Characterization

We refer to a *source case* as a skill demonstration that has been provided to the robot and is stored in the robot's case memory. Thus, we use the terms *demonstration*, *skill demonstration*, and *case* interchangeably. Each demonstration is defined as $d = <p, a>$, where p encodes the *problem* the demonstration seeks to address and a encodes the demonstrated action. We focus on representing demonstrations that illustrate only one *action label* (e.g. "pouring", "opening", "stacking"). The list of observed objects, o, and the list of observed features of the objects (color, size, etc.), f, are also elements of the demonstration representation. A skill demonstration then consists of the following elements:

- The problem description $p = <o, f, v>$, where o and f are as described above, and v is a set of parameters (e.g. initial object locations, initial end-effector position)
- The action model $a = \{j_0, j_1, \ldots, j_i\}$ encoding the robot's end-effector position at each time interval i.

The case-based interpretation process uses the problem descriptions of sources cases in memory as input, such as the demonstrations shown in the first two rows in Fig. 1, and a target problem, such as the third row in Fig. 1, and maps the target problem to the most similar case in memory. Case-based interpretation is completed by evaluating the similarity between the visual representations of the target problem and the source cases, i.e., on o and f, and the visual transformations in them, and does not require semantic information that specifies the demonstrated action label.

We define a visual transformation as the tuple $<S_i, S_f, T>$, where S_i is an overhead view of the initial state (the first column of images in Fig. 1), S_f is the observed goal state that is reached following the skill completion (the second

column of images in Fig. 1), and T is the visual relation, or transformation, between the two images S_i and S_f.

4 Algorithms

4.1 Fractal Method

Our first approach uses fractal representations to encode the visual transformation function T between two images [16], 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. Thus, the transformation function T encodes a set of sub-transformations between S_i and S_f. The Fractal method evaluates similarity at several levels of abstraction, allowing automatic adjustment of the level of spatial resolution. 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 [16,27]. 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 [16]:

1. The initial state image is segmented into a grid containing a specified number of partitions, $S = \{s_0, s_1, \ldots, s_p\}$, where p is determined by the abstraction level n.
2. For each sub-image $s \in S$, the destination image is searched for a sub-image d such that for some transformation $k \in K$, $k(s)$ is most similar to d.
3. The transformation k and shift c, the mean color-shift between d and $k(s)$, are used to create a fractal code f_s.
4. The resulting fractal is defined by $F = \{f_0, f_1, \ldots, f_p\}$

This encoding process is repeated for multiple values of n, resulting in an encoding of the transformation at n levels of abstraction, where n is derived from the images' pixel dimensions. Here, we partition each 300 px by 180 px image at $n = 7$ 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
- $k \in K$ represents the affine transformation between the source and destination sub-images where $K = \{$ 90 ° clockwise rotation, 180 ° rotation, 270 ° clockwise rotation, horizontal reflection (HR), vertical reflection (VR), identity (I) $\}$. k is the transformation that converts sub-image s into sub-image d minimally, while requiring minimal color changes.

– c is the mean color-shift between the two sub-images

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 that of 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.

4.2 SIFT Feature-Matching

The SIFT algorithm selects keypoint features using the following steps [14]. 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 second approach to source demonstration retrieval using SIFT features is based on feature-matching. 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 16×16 pixel area surrounding the feature keypoint. A brute-force method is used to determine that two features match if they have the most similar 16×16 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
$c \leftarrow null$; $m \leftarrow 0$
$U_i \leftarrow$ 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 $size(T) > m$, then: $m \leftarrow size(T)$, $c \leftarrow d$
end for
return c

Figure 3(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.

4.3 SIFT Feature-Transformation

Our final approach to source demonstration retrieval via the SIFT algorithm serves as an intermediate method which incorporates aspects of the Fractal method's emphasis on visual transformations, while adopting the same feature selection strategy as the previous SIFT feature-matching method. This approach 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 S_f, as shown in Fig. 3(b). This method uses the same features and feature-matching method as in the feature-matching 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 location in the initial and final states, 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 source skill demonstration. 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 blurring levels, which serves to reduce the number of irrelevant or noisy features comparably to the Fractal method's usage of multiple abstraction levels. At each blur level, 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. The SIFT feature-transformation method retrieves a source demonstration as follows:

Let D be a set of source skill demonstration images
$c \leftarrow null$; $m \leftarrow 0$; $x \leftarrow 0$
for each demonstration $d \in D$ **do**
    $n \leftarrow 0$
    while $n <$ maximum abstraction level **do**
        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 $diff(t_u, t_c)$

15: **end for**
16: $x \leftarrow 0$
17: **for** each transformation $t_u \in T_u$ **do**
18: $x \leftarrow x + \mathit{diff}(t_u, t_c)$
19: **end for**
20: If c is *null* or $x < m$, then: $c \leftarrow d$, $m \leftarrow x$
21: $n \leftarrow n + 1$
22: **end while**
23: **end for**
24: **return** c

5 Experiment

Each approach was used to retrieve a source skill demonstration for three test sets of target demonstrations. Each skill demonstration is a pair of two recorded keyframe images depicting the initial state and end state of a box-closing or cup-pouring skill performed by a human participant, as shown in Figs. 1 and 2. Nine participants demonstrated the two skills, and were recorded using an overhead camera above the tabletop workspace. Participants indicated the initial and final states verbally, and were asked to remove their hands from view when the initial and final states were recorded. Each participant's demonstration set consisted of nine demonstrations per skill, each skill being performed at the orientations shown in Figs. 1 and 2.

We evaluated the algorithms on three test sets, each representing retrieval problems of a different difficulty level. In the *aggregate* set, a source demonstration is retrieved for two participants' demonstrations (two skills each performed with two objects at three configurations, resulting in 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 two boxes and two pouring objects, respectively, shown in the first two rows of Figs. 1 and 2. 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 two boxes and two 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 Figs. 1 and 2. As a result, demonstrations involving a third kind of box and pouring object were introduced, shown in the last row of Figs. 1 and 2. A source demonstration was retrieved for each target demonstration from a library

Table 1. Source Case Retrieval Results

Test set	Fractal	SIFT feature-matching	SIFT feature-transformations
Aggregate	100 %	100 %	91.7 %
Individual	87 %	100 %	35.2 %
Analogical	65.3 %	93.8 %	84.5 %

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.

6 Experimental Results

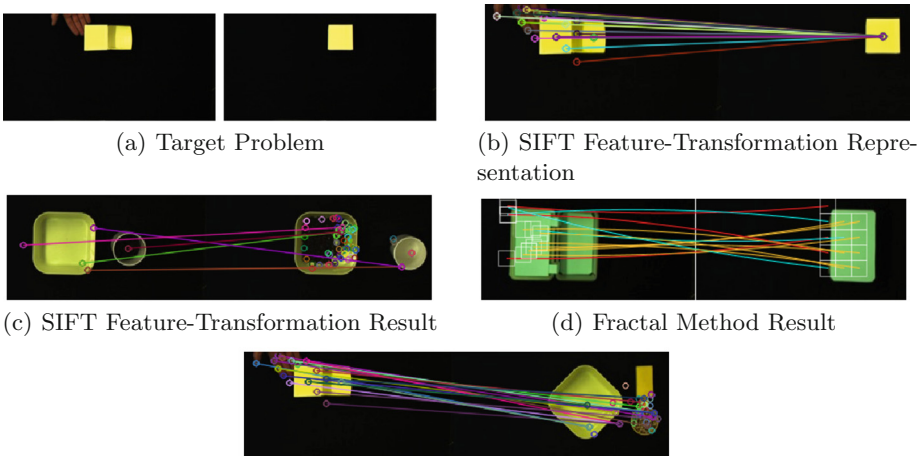
Table 1 lists the overall accuracy of each method when applied to each test set. Since the aggregate test contained a large set of source demonstrations and was most likely to contain a demonstration similar to the target problem, we expected that this test set would be the easiest test set for any of the three methods to address.

6.1 Detailed Analysis

While the experimental results provide useful information about the accuracy of the three methods, it is useful to also analyze the strengths of each method.

Case Study: Fractal Method Success. First, we analyze an example in which only the Fractal method retrieved an appropriate source demonstration. Figure 3(a) depicts the target problem demonstration, which the Fractal method correctly matched to the source demonstration shown in Fig. 3(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 Fig. 3(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 Fig. 3(e). The SIFT feature-transformation method also incorrectly classified the demonstration as a pouring skill demonstration. Figure 3(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 Fig. 3(c)).



(a) Target Problem

(b) SIFT Feature-Transformation Representation

(c) SIFT Feature-Transformation Result

(d) Fractal Method Result

(e) SIFT Feature-Matching Result

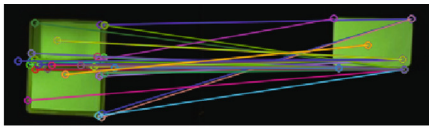
Fig. 3. Case study 1: retrieval method results

We 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 some clutter, such as the demonstrator's hand or other objects unrelated to the skill being performed. This case study also demonstrates that the feature-matching method is sensitive to clutter. Additionally, 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. As an example, the feature-transformation method would perform poorly given a demonstration of a book-closing skill, where initial-state SIFT features detected on the inside pages of the book cannot be matched to final-state SIFT features on the cover of the book.

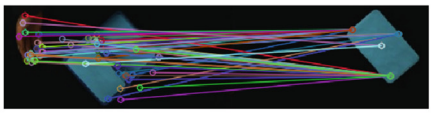
Case Study: SIFT Transformation Success. In the next case, only the SIFT feature-transformation method retrieved an appropriate source demonstration for the target problem shown in Fig. 4(a). The SIFT feature transformation method retrieves visually analogical source demonstrations by identifying visual transformations at multiple abstraction levels. The transformations in Fig. 4(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 Fig. 4(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

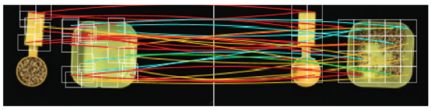
(a) Target Problem



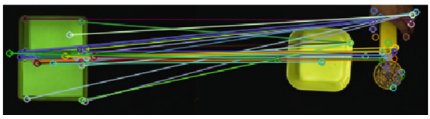
(b) SIFT Feature-Transformation Representation



(c) SIFT Feature-Transformation Result



(d) Fractal Method Result



(e) SIFT Feature-Matching Result

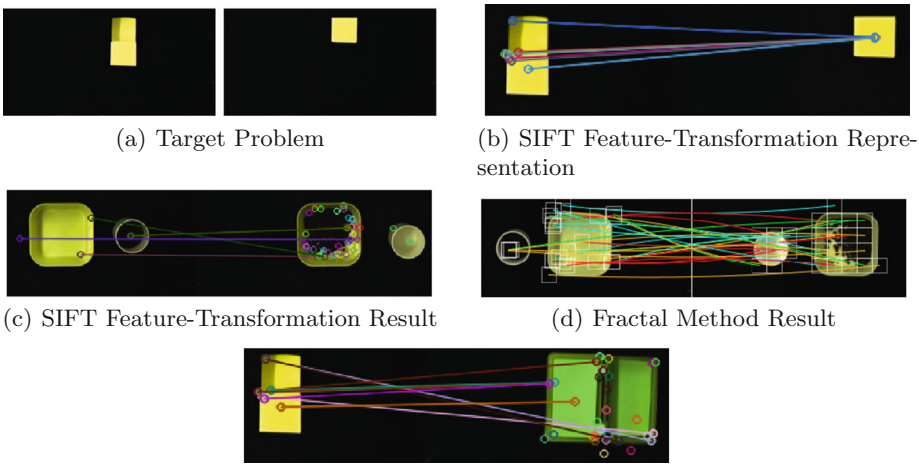
Fig. 4. Case study 2: retrieval method results

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 Fig. 4(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.

Case Study: SIFT Feature-Matching Success. In the final case study, only the feature-matching method retrieved the correct source demonstration to address the target problem shown in Fig. 5(a). This method correctly corresponds features between the target problem and source demonstration's initial and final state features. The initial state feature mapping is shown in Fig. 5(e).

Just as in the first case study, 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



(a) Target Problem

(b) SIFT Feature-Transformation Representation

(c) SIFT Feature-Transformation Result

(d) Fractal Method Result

(e) SIFT Feature-Matching Result

Fig. 5. Case study 3: retrieval method results

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. In this example, the Fractal method 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 Fig. 5(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.

The feature-matching method performed best on each test set. However, we anticipate that this method would not perform well on skill demonstrations in which irrelevant features are present, such as clutter or the demonstrator's hand. Additionally, this method would mistake skill demonstrations with the same feature set; block-sorting and block-stacking demonstrations could be performed using the same objects, and thus the two demonstrations would be matched as a result of having the same set of features.

6.2 Discussion

Several variables may affect the accuracy of each skill interpretation method. The Fractal 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.

7 Conclusion

We have explored visual case retrieval for case-based interpretation of skill demonstrations as a precursor to case-based robot learning by imitation. We have presented three methods for this task: SIFT feature-matching, SIFT feature-transformation, and Fractal feature-transformation. Although the general SIFT algorithm is widely used for computer vision tasks, the use of fractal and SIFT features in case-based skill interpretation is new insofar as we know.

No single method works best for all case-based skill interpretation problems. Rather, each method discussed in this paper is best suited for a particular type of problem. The feature-matching method is best suited for interpretation problems in which enough visual features can be extracted to identify the skill and no clutter 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 method is most effective in identifying skills in which the visual transformation should be emphasized, rather than features of the demonstration images themselves. This suggests the use of a multi-strategy technique for visual case retrieval in the domain of interpreting skill demonstrations.

Acknowledgments. This material is based upon work supported by the United States' National Science Foundation through Graduate Research Fellowship Grant #DGE-1148903 and Robust Intelligence Grant #1116541. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: foundational issues, methodological variations, and system approaches. AI commun. **7**(1), 39–59 (1994)
2. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robot. Auton. Syst. **57**(5), 469–483 (2009)

3. Cheetham, W., Graf, J.: Case-based reasoning in color matching. In: Leake, D.B., Plaza, E. (eds.) ICCBR 1997. LNCS, vol. 1266, pp. 1–12. Springer, Heidelberg (1997)
4. Davies, J., Goel, A.K., Nersessian, N.J.: A computational model of visual analogies in design. Cogn. Syst. Res. **10**(3), 204–215 (2009)
5. Ferguson, R.W., Forbus, K.D.: GeoRep: a flexible tool for spatial representation of line drawings. In: AAAI/IAAI, pp. 510–516 (2000)
6. Floyd, M.W., Esfandiari, B.: A case-based reasoning framework for developing agents using learning by observation. In: 2011 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 531–538. IEEE (2011)
7. Floyd, M.W., Esfandiari, B., Lam, K.: A case-based reasoning approach to imitating robocup players. In: FLAIRS Conference, pp. 251–256 (2008)
8. Grimnes, M., Aamodt, A.: A two layer case-based reasoning architecture for medical image understanding. In: Smith, I., Faltings, B.V. (eds.) EWCBR 1996. LNCS, vol. 1168, pp. 164–178. Springer, Heidelberg (1996)
9. Gross, M.D., Do, E.Y.L.: Drawing on the back of an envelope: a framework for interacting with application programs by freehand drawing. Comput. Graph. **24**(6), 835–849 (2000)
10. Jurisica, I., Glasgow, J.: Applications of case-based reasoning in molecular biology. AI Mag. **25**(1), 85 (2004)
11. Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann, San Mateo (1993)
12. Leake, D.B.: Case-Based Reasoning: Experiences, lessons and future directions. MIT press, Menlo Park (1996)
13. Lopez De Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M.L., Cox, M.T., Forbus, K., et al.: Retrieval, reuse, revision and retention in case-based reasoning. Knowl. Eng. Rev. **20**(03), 215–240 (2005)
14. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157. IEEE (1999)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004)
16. McGreggor, K., Goel, A.: Fractal analogies for general intelligence. In: Bach, J., Goertzel, B., Iklé, M. (eds.) AGI 2012. LNCS, vol. 7716, pp. 177–188. Springer, Heidelberg (2012)
17. McGreggor, K., Kunda, M., Goel, A.: Fractals and ravens. Artif. Intell. **215**, 1–23 (2014)
18. Meltzoff, A.N.: Imitation and other minds: the "like me" hypothesis. Perspectives on Imitation: From Neuroscience to Social Science, vol. 2, pp. 55–77 (2005)
19. Ontañón, S., Mishra, K., Sugandh, N., Ram, A.: Case-based planning and execution for real-time strategy games. In: Weber, R.O., Richter, M.M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp. 164–178. Springer, Heidelberg (2007)
20. Perner, P.: An architecture for a CBR image segmentation system. Eng. Appl. Artif. Intell. **12**(6), 749–759 (1999)
21. Perner, P., Holt, A., Richter, M.: Image processing in case-based reasoning. Know. Eng. Rev. **20**(3), 311–314 (2005)
22. Richter, M.M., Weber, R.: Case-Based Reasoning. Springer, Heidelberg (2013)
23. Riesbeck, C., Schank, R.: Inside Case-Based Reasoning. Lawrence Erlbaum Associates, Hillsdale (1989)
24. Ros, R., Arcos, J.L., De Mantaras, R.L., Veloso, M.: A case-based approach for coordinated action selection in robot soccer. Artif. Intell. **173**(9), 1014–1039 (2009)

25. Thagard, P.: Mind: Introduction to Cognitive Science. MIT press, Cambridge (2005)
26. Tomasello, M., Kruger, A.C., Ratner, H.H.: Cultural learning. Behav. Brain Sci. **16**(03), 495–511 (1993)
27. Tversky, A.: Features of similarity. Psychol. Rev. **84**(4), 327 (1977)
28. Yaner, P.W., Goel, A.K.: Analogical recognition of shape and structure in design drawings. Artif. Intell. Eng. Des. Anal. Manuf. **22**(02), 117–128 (2008)