If You Can Draw It, You Can Recognize It: Mirroring For Sketch Recognition

Mor Vered and Gal A. Kaminka

Computer Science Department and Gonda Brain Research Center
MAVERICK Group
Bar Ilan University, Israel
{veredm,galk}@cs.biu.ac.il

Abstract. Humans use sketches drawn on paper, on a computer, or via hand gestures in the air as part of their communications. To recognize shapes in sketches, most existing work focuses on offline (post-drawing) recognition methods, trained on large sets of examples which serve as a plan library for the recognition method. These methods do not allow on-line recognition, and require a very large library (or expensive pre-processing) in order to recognize shapes that have been translated, rotated or scaled. Inspired by mirroring processes in human brains we present an online shape recognizer that identifies multi-stroke geometric shapes without a plan library. Instead, the recognizer uses a shape-drawing planner for drawn-shape recognition, i.e., a form of plan recognition by planning. This method (1) allows recognition of shapes that is immune to geometric translations, rotations, and scale; (2) eliminates the need for storing a library of shapes to be matched against drawings (instead, only needs a set of possible Goals and a planner that can instantiate them in any manner); and (3) allows fast on-line recognition. The method is particularly suited to complete agents, that must not only recognize sketches, but also produce them, and therefore necessarily have a drawing planner already. We compare the performance of different variants of the recognizer to that of humans, and show that its recognition level is close to that of humans, while making less recognition errors early in the recognition process.

1 Introduction

Humans use sketches, drawn on paper, on a computer, or via hand gestures in the air, as part of their communications with agents, robots, and other humans. Whether in computer graphics applications that require sketch-based modeling [9], or in innovative assistive robotics applications [27, 15, 20], or in the increasing use of sketch-based user interfaces in tablets and other ubiquitous computing devices [13].

Gesture signalling has always been one of the primal, basic ways for humans to interact and humans have perfected the ability to recognize online gestures quickly and efficiently. In order to understand how humans perform this recognition we draw from neuroscience, psychology and cognitive science. There has been evidence that humans ability to do online shape recognition comes from the newly discovered *mirror neuron system* for matching the observation and execution of actions within the adult human brain [19]. The mirror neuron system gives humans the ability to infer the intentions leading to an observed action using their own internal mechanism.

Mirror neurons have first been discovered to exist in macaque monkeys in the early 90's [16]. These neurons for manipulation were seen to fire both when the monkey manipulated an object in a certain way and also when it saw another animal manipulate an object in a similar fashion. Recent neuroimaging data indicates that the adult human brain is also endowed with a mirror neuron system where it is attributed to high level cognitive functions such as imitation, action understanding, intention attribution and language evolution. The human mirror neuron system may be viewed as a part of the brains' own plan recognition module and can be used to recognize the actions and goals of one or more agents from a series of observations of the other agents' actions.

To recognize shapes in sketches, most existing work focuses on offline (post-drawing) recognition methods, trained on large sets of examples [2, 23, 22, 25]. Given the infinite number of ways in which shapes can appear—rotated, scaled, translated—and given inherent inaccuracies in the drawings, these methods do not allow on-line recognition, and require a very large library (or expensive pre-processing) in order to recognize even a small number of shapes [1].

Inspired by mirroring processes hypothesized to take place in socially-intelligent brains [16, 11] we present an *online* shape recognizer that identifies multi-stroke geometric shapes without a plan library. The recognizer uses a shape-drawing planner for drawn-shape recognition. In that aspect our work relates closely to the work done in [17], i.e.,a form of plan recognition by planning, [7], goal recognition through the analysis of goal graphs, agent tracking [24], or mirroring in virtual humans [12, 21, 20]. In Section 2 we will adress these works and discuss how, while our approach relates to these works, our method differs signifiantly from them.

There are many advantages to a planner-based shape recognizer, inspired by the mirror neuron system principles. Some immediate technical advantages from a recognition point of view include: (1) considerable reduction in storage space - no need for storing both a library of shapes to be matched against drawings and a separate library of shapes to be sent to the shape planner; the recognizer will use the knowledge of its own plan generated by its own shape planner in an online manner while eliminating the need for different plans according to different scales and rotations; and (2) fast on-line recognition.

However, the key advantage rises from the point of view of the complete agent that uses the recognition as part of its interactions: an agent that actively communicates using sketches—thus necessarily possessing a sketch planner—will be able to use our methods to recognize sketches, without relying on a separate shape recognition library. This is the motivation for our work.

We carefully evaluate the approach. We compare the performance of different variants of the recognizer to that of humans, and show that its recognition level is close to—in some cases better than—that of humans, while making less recognition errors early in the recognition process. The results also demonstrate that humans utilize additional knowledge in their recognition of shapes, as they are able to guess—sometimes correctly—the shape being drawn even based on a single drawn edge (which logically can be a part of any polygonal shape).

2 Related Work

Intelligent systems increasingly rely on sketching, hand-drawing and gestures as input. In its essence sketching and gesturing are one of the fundamental ways for humans to interact and is therefore often an important part of any intelligent system. The problem of sketch recognition may be also viewed as a very important instance of plan recognition, since part of what makes a system intelligent is the ability to foresee the needs and intentions of its users.

A particular aspect of drawn shapes is that they can be drawn in an infinite number of ways within the drawing area. Furthermore, given that the shapes are drawn by humans, both edges and vertices are drawn with quite a bit of inaccuracy, in edge curvature (i.e., they are not straight lines), in the accuracy of angles between edges, or even in the drawing of angles themselves (for instance, whether two edges actually intersect in a vertex).

Thus shape recognition—whether offline or on-line—faces the following key challenge: there exist essentially infinite numbers of possible sketches of each goal shape. From the point of view of plan- or goal- recognition, this poses the challenge of recognizing a small set of goals, given an infinitely-large plan library. Naturally, the challenge is exacerbated in on-line shape recognition, as the agent cannot easily tell whether it has seen all observations.

Most approaches to shape recognition use global geometric properties extracted from the drawings, and specialized to the recognition task. For instance, Paulson and Hammond designed a system that works by computing specific tests for all possible shapes, then sorts the matching hypotheses (matching shapes) in order of best fit [14]. Ulgan et al. use a neural network, trained on the relation between the internal angles of a shape and its classification [26].

All of these are offline approaches: they carry out the recognition process only once the drawing is completed.

Online methods in the same spirit (relying on specialized geometric features) include [6, 10]. These implement methods that are invariant to scale and rotation. They use global geometric properties extracted from input shapes ahead of time and associated with certainty degrees using fuzzy logic. These methods required substantial work ahead of time in selecting the best feature to identify a given shape while the initial shape selection process takes into account specific shape related properties. Another method for on-line recognition that also requires considerable training, i.e., an extensive set of examples, is the use of HMMs for sketch recognition [22].

The advantage of our approach over these methods is in the utilization of an existing planner in order to perform the recognition process. Thus eliminating the need for training ahead of time and for specific preparatory analysis for each individual plan.

Our idea relates to the model tracing approach [4] in the sense that the system must possess a computational model capable of solving the problems given to the student. The difference lies in the complexity of implementing a similar mechanism that will be able to work in a continuous, unpredictable domain that has to deal with missing knowledge, noise, and an infinite possibility of solutions.

We follow previous work [21, 20] in treating the problem of on-line recognition of shapes, as they are being drawn, as a problem of on-line goal recognition by mirroring.

However, In [21, 20] Sadeghipour et al. explicitly represent (and store) shape drawing *plans*, that can be used both for recognition and execution by the agent. In contrast, we do not store plans, but instead use a *planner* to generate plans on the fly. Thus in these previous works different rotations of the same shapes have to be stored as separate plans in the plan library, and the plan library must account for all rotations.

A technique similar in spirit to that of Sadeghipour et al., is that of agent tracking [24], which uses a virtual agent's own BDI plan to recognize a BDI plan being executed by another agent. And similarly, this approach stores plans, rather than utilize a planner as we do.

Indeed, the key to our approach is the use of a planner, instead of a plan library, in order to avoid the problem of explicit representation of all possible plans for drawing the goal shapes. In this, we are somewhat inspired by work on *plan recognition by planning* [17, 18] and in a similar manner by work on *goal recognition through goal graph analysis* [7]. The idea is to dynamically generate plans that match existing observations, narrowing down the list of possible goals. This way, a very large set of plans is implicitly stored—by being generated as needed, and without the need for prior training on examples. However, while they depend on PDDL-capable planners in discrete domains with no uncertainty, our approach cannot; shape recognition necessarily takes place in continuous domains, with inaccuracies and noise.

In [7] they contest that recognizing the intended goals should aim at explaining past actions (rather than predict future actions). Therefore they distinguish partially or fully achieved goals from the other possible goals according to the observed actions. While we aim to predict the future rather than explain the past, we build on this principle in our approach by our recognizers' ability to eliminate goals that are impossible to achieve ahead of time, according to the observations.

3 Using a Shape-Drawing Planner to Recognize Drawn Shapes

In order to demonstrate our approach we chose to address the problem of shape recognition through the platform of shapes drawn on paper. Therefore in order to perform the recognition, our system will utilize its own existing, shape-drawing planner instead of referring to an existing plan library.

3.1 Overview

We treat the problem of on-line shape recognition as a problem of on-line goal recognition. Here, the set of known goals to be recognized, G, is a set of k polygon labels, distinguished by the number of sides (edges) and the size of the internal angles between them.

The agent's goal recognition task is to accurately select the intended goal $g_x \in G$, given the stream of observations O, *as early as possible*, i.e., with the shortest subset of O. O_i denotes the $i^{th} (i \geq 0)$ observation in the stream, so the task is really to minimize i while correctly identifying $g_x \in G$. The entire stream O is an instantiated plan for drawing a specific instantiated goal shape g_x.

Each observation $O_i \in O$ is an edge of the polygon, connected to a previously observed edge (with the exception of O_0, which consists of a single edge drawn arbitrarily in the drawing area). Each edge is represented by its line equation in the form $O_i = a_i x + b_i$, where a_i is the slope of the i^{th} edge and b_i is the intercept. As the sketch

recognizer is supposed to work with scanned images from paper, gestures in the air, or computer graphics input, the recognizer does not commit to a particular technique for translating sensor data into edge line equations. For example, in our experiments with scanned drawings, we utilized a Hough-transform technique (see Section 4).

The shape recognizer also takes as input two anchor points (x_0, y_0) representing the first observed point and (x_n, y_n), $(n \geq 0)$, the latest observed coordinate, marking the open end of the latest observed edge. Each point is comprised of a location and a separate unit vector indicating the direction in which the edge is being built. The direction of each point is updated gradually as the shape is being observed while maintaining a default direction of the last known direction - or random for the initial point. These points will be regarded as anchor points and will later indicate to the planner where to start and end constructing the shape.

The output of the shape recognizer is an ordered list, R, of all of the goals (i.e., shapes) matching the observations, in order of likelihood. The key to mirroring comes in the reuse of a shape-drawing planner in the recognition process. The idea is to use the planner to generate shapes whose prefix of length i (the first i edges) matches the first i observations $O_0 \ldots O_i$. This is done incrementally, with each increasing i. Each new observation further constrains the possible shapes that could be drawn. Thus as the observations come in, the list of possibly matching shapes slowly stabilizes to a (potentially ranked) list of candidate goals.

We build on the principles outlined in [17, 18] in that our mechanism forces a planner to utilize specific observed edges as a prefix for the plan. Their approach is to explicitly fold past observations (edges and anchor points) into the initial state given to the planner. Thus the shape-drawing planner accepts an initial state that is comprised of a partially-drawn shape, anchored to a specific origin point and with at least one clear open end where the next edge should be connected. The planner accepts a goal shape, and returns a plan—a set of edges—that will complete the drawing of the goal shape, from the initial state (or it may return a result that indicates no plan is possible). By iterating over all possible goal shapes, one can systematically check all possible shapes (out of those still not ruled out), for each new observation.

One difficulty with the approach of [17] is that with each observation, and for every goal, the planner needs to provide a complete plan, from the initial state to the goal state. The initial state only differs from one observation to the other in that it adds constraints—the generated plan prefix must necessarily comprise of the steps already observed. Thus the planner's task is computationally intensive.

We differ from this approach in that instead of explicitly folding observations into the initial state, we instead do so implicitly. Instead of asking the shape planner, In our approach we build on both of these principles. In our case, to generate a new shape, from scratch, with edges as observed in $O_0 \ldots O_i$, we only ask the planner to produce *a remainder shape*—the part of the shape that completes the current observations into the goal considered. As observations become available incrementally, the remainder shape necessarily grows smaller and smaller, and thus easier and easier to compute.

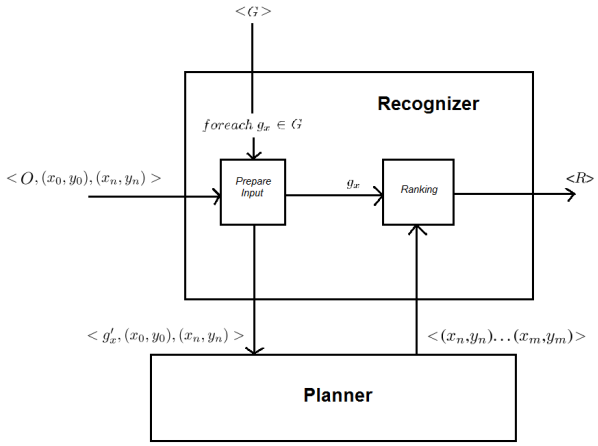


Fig. 1: The components of the shape recognition process: *Prepare input*, *Ranking*, and *Recognizer*. The *planner* is external, but is relied on by the recognizer.

3.2 A Regular Polygon Recognizer

To carry out this recognition process, in particular re-using a shape drawing planner in service of recognition, several components are needed. These are shown in Figure 1, and described below in detail.

In the experiments presented in this paper, we utilized a drawing planner for regular polygons only (equilateral, equiangular). Thus we use this type of shapes to describe how the recognition process works, and how the different components interact. Figure 1 shows the two inputs of the recognition process. The first is the stream of observations O and two anchor points, and the second is the set of possible goals G (normally given once, but can dynamically change). We have already discussed how each observation is given by a line equation for the edge it represents, and the two anchor points. For regular polygons, each goal, $g_x \in G$, is distinguished (and represented) by its unique number of sides and the size of the internal angles between edges; i.e., $\langle 3, 60^\circ \rangle$is the goal denoting equilateral triangles.

Prepare Input The goal and observations input are fed into the *Prepare Input* component, whose task is to prepare the input to be sent to the *Planner*. This is the key step in the mirroring approach.

With each new observation O_i, and anchor points (x_0, y_0) and (x_n, y_x) the task of the *Prepare Input* component is to iterate over all possible goals, folding the known observations $O_0 \ldots O_i$ and the anchor points into the input given to the planner. As there is no standard language for shape planning, different planners will require slightly different input preparation.

The *Prepare Input* component incorporates the available observation history O_i into g_x by creating a new goal, g'_x, that removes from g_x edges already seen, and is com-

prised only of the remainder of the polygon, i.e., the part expected to be completed if the observations $O_0 \ldots O_i$ are to be a part of g_x. We refer to g'_x as the *polygon remainder*.

For regular polygons, computing the polygon remainder involves calculating the expected angles in vertices, and the expected size of each remaining edge. Under ideal conditions, all edges already observed are equally-sized, and all observed angles are identical. In reality, however, inaccuracies in the drawing of shapes leads to edges that are not all the same size, and shapes that similarly are not ideal. Because of this, the prepare input must make some assumptions in its prediction of how the polygon will be completed (i.e., in what actual edge sizes and internal angles will be utilized).

We chose an optimistic heuristic for this assumption. We ignore the length of observed edge, and instead divide up the remaining angles equally among the remaining vertices. As the angles are thus fixed, and the open ends of the polygon are known, the edge sizes become fixed.

For example, suppose the observations so far have been of two edges, joined by a 90° angle. Suppose g_x (the target polygon) is a pentagon (5 sides, and necessarily a sum of all internal angles of 540°, (five angles of 108°). Then the polygon remainder g'_x would consist of three edges, with the remaining four internal angles each of $(540^\circ - 90^\circ)/4 = 112.5^\circ$.

There may be shapes in which the recognizer cannot incorporate the history into the goal, for instance, if we have seen three edges of a square with 90° angles between them, this cannot be incorporated into the goal of a triangle. In this case the recognizer automatically dismisses that goal and does not utilize the planner at all.

Planner After the goals have been adjusted, each possible goal is sent to the *Shape Planner*, along with the initial and current anchor points. Because the goal already incorporates the history of previously seen observations, the planner need only plan the rest of the shape, excluding the part already seen. It starts at the current point and adds edges until completing the rest of the polygon. The output of the planner is a completely planned shape polygon remainder, starting from the last observed point (x_n, y_n) and described by the consecutive vertex points $(x_n, y_n) \ldots (x_m, y_m)$.

If the planner is unable to generate a plan for drawing the polygon remainder g'_x, it issues an error which indicates that it is not possible to draw the specific g_x from which g'_x was derived. This indicates that g_x is *not* a possible goal, given the observations.

Thus taken together, the *Prepare Input* and the *Planner* components work essentially as a generate-and-test process. The *Prepare Input* component sets up possible hypotheses, and the *Planner* tests them, returning a plan to indicate the hypothesis passed, or error (no plan) to indicate the hypothesis should be discarded.

The end result of this process is a set (thus, unordered) of hypothesized shapes that match the observations thus far, generated without relying on a stored set of examples, or instantiated shapes. This set may be analyzed in various ways, to generate a ranked list of shapes, e.g., in order of likelihood [18] or relevance [24].

Ranking Recognition Hypotheses One way of determining a ranking order over the set of recognition hypotheses (i.e., the set of possible shapes matching the observations) is to rank them based on errors, when compared to the ideal goal shapes in G. Remember, recognition hypotheses are based on *instantiated shapes*, i.e., the actual drawings with

all of the inaccuracies. But hypotheses are of goal shapes in G. Thus the idea is to measure the geometric errors for each possible plan, between the instantiated shape, derived by the hypothesized goal g'_x and the corresponding shape derived by the original goal g_x.

The *Ranking* component compares each shape with the original goal shape and measures similarity according to the differences in edge relations and in overall angles comprising the shapes. The shape with the minimal amount of difference is ranked highest. Shapes which differ from the goal shape with an average internal error angle of more than 30° will be automatically disqualified. In this way, our recognizer is able to account for noise, allowing that even shapes with an error of 30° may still be possible. As we increase or decrease this threshold we also relax or restrict the recognition conservativeness when accounting for inaccuracy in the drawing.

Incidentally, this use of error in angle, specifically, as the ranking criteria, seems to also agree with studies of human estimates of intentionality and intended action [3]. Such studies have shown a strong bias on part of humans to prefer hypotheses that interpret motions as continuing in straight lines, i.e., without deviations from or corrections to, the heading of movements.

4 Experiments and Results

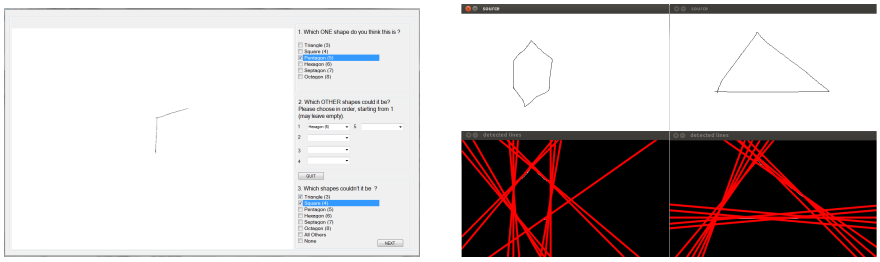
This section presents the results of empirically evaluating the performance of the shape recognizer, contrasting it with the performance of 20 human subjects in the exact same tasks. Section 4.1 describes the experiment setup. Section 4.2 presents the key results.

4.1 Experiment Setup

To evaluate the shape recognizer we designed an experiment to test its recognition ability in different ways, and contrast its performance with that of humans. The key to the experiment is to utilize the same hand-drawn inputs for both human recognition as well as machine recognition. The performance of the recognizer (and humans) on this data can be potentially used to generate two types of insights. First, by noting failures and successes in specific cases, we can learn about recognition capabilities and weaknesses. Second, by contrasting human and machine recognition, we can make some deductions as to how humans do or do not carry out the recognition process.

The basis for the experiment is in a data-base of scanned hand-drawn regular polygons. We asked three people to create a data base of 18 hand drawn regular polygons : 3 triangles, 3 squares, 3 pentagons, 3 hexagons, 3 septagons and 3 octagons. The people were instructed to draw the shapes as accurately as the could, making them as regular as possible (i.e., equilateral, equiangular). Shapes were drawn in various scales, rotations, and translations with respect to the center of the page. Naturally, hand drawings, even under these ideal conditions, reflect quite a bit of inaccuracy (see Figure 2b, top).

Each drawn shape was then scanned. To show them to humans in the same way that the recognizer perceives them, we separated the edges into separate images so when presenting them sequentially it would appear that they were constructed edge by edge: In the first image only the first edge would appear, in the second image the first and the second edges would appear and so on.



(a) Interface for the human subjects experiment. Shown are two edges (part of the polygon), and question interface to the right.

(b) Drawn shapes (above) and their Hough transforms (below).

Fig. 2: Experiment visual and data preparation.

Human recognition data collection Using these images we then conducted a human recognition experiment in order to collect data about human recognition performance. 20 human subjects (14 men and 6 women ages 19–52, with a mean age of 29) participated in the experiment. The participants were instructed to observe each edge and then to answer the following questions (after watching each edge), all using software built for this purpose (Figure 2a).

1. Which *one* shape do you think it is ? Only one option must be chosen.
2. Which other shapes could it be ? The participants were asked to rank, in consecutive order, the remaining shapes they thought likely. This field may be left empty.
3. Which shapes it definitely could not be ?

The participants were asked to take into consideration that the shapes may appear in any size and rotation and that, as the shapes were drawn by humans, may be drawn inaccurately. It was also explained to the participants that questions 2 and 3 were not necessarily complimentary in the sense that in question 2 one might pick one other shape that you seem most likely and in question three you may enter that all shapes were possible or only some of the remaining shapes.

From images to machine observations We wanted to use our recognizer on the same images as humans, allowing observations of one edge at a time. In order to obtain the line information from each image we used OpenCV to implement a Hough Transform [5], a feature extraction technique commonly used in computer vision. The performance of the technique on two example drawings is shown at the bottom of Figure 2b.

For each detected edge we were able to extract the following information: the initial and final x,y coordinates, the ρ parameter, which is the algebraic distance between the line and the origin, and Θ, the angle of the vector orthogonal to the line and pointing toward the half upper plane. From this it was easy to find the slope and intercept of each line along with the initial and end point coordinates.

Because of noise in perception (e.g., scanning noise) and drawing inaccuracy, the Hough transform often generates several candidate lines for each edge (can be seen

in Figure 2b). To find the common lines we used open-source hierarchical clustering software [8]. We defined each node to have equal weight and used Euclidean distance to measure the distances between each node and gave a threshold of 100 to check for affinity between nodes. Following this we had the number of lines recognized and the slope and intercept of each line.

This input, along with the initial and end point coordinates (the anchor points), were fed to the recognizer. To be able to test the significance of each of the major components, we contrast the results of the recognition with the ranking method described above, the *ranking recognizer*, and without it (i.e., no ordering on the results, all possible goals have an equal chance of being chosen), the *non-ranking recognizer*.

4.2 Results

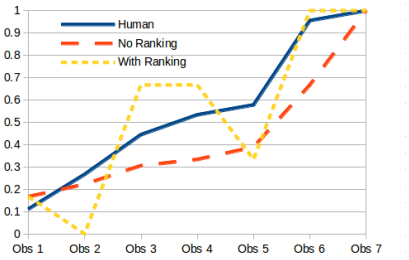
All the data (human subject results, ranking recognition results, non-ranking recognition results) was initially examined separating each polygon goal (i.e., all the data for triangles was separated from the data for squares). This is because necessarily, the number of observations in the observation stream O differs between these. A triangle has a maximum of three observations; a septagon has seven. We naturally examine the results for each question separately.

Question 1. Which shape is it? Question 1 allows the recognizer (machine or human) only a single guess as to the goal shape. Thus this is a conservative test of accuracy, as the guess either matches or does not match the ground truth.

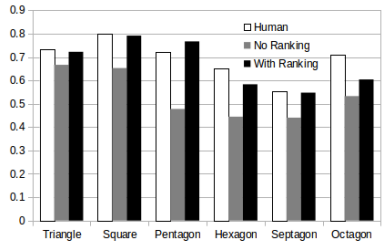
To illustrate the progression of a recognition process as observations accumulate, Figure 3a shows the results for question 1, for septagons. The figure contrasts the performance of the non-ranking recognizer, with that of the ranking recognizer, with the mean performance of the human subjects. The horizontal axis in Figure 3a counts the incrementally accumulating observations (edges). Thus the marking "Obs 3" denotes three edges that are revealed to the recognizer. The vertical axis measures success, as the ratio of correct guesses to the total number of guesses: 0 means the recognizer in question never succeeded in guessing the correct shape at the given point, 1 means it always did. In the case of non-ranking recognition, which cannot choose a top hypothesis, the statistically expected success rate for random selection is used ($1/k$ for k choices).

The figure shows clear monotonically increasing success for both human recognition as well as the non-ranking recognizer, with the human recognition consistently better than the non-ranking recognizer. The monotonic behavior of the graph reflects the fact that with each observed edge, less goal shapes are possible, and thus can be ruled out. The ranking recognizer performs inconsistently. For two observed edges, it consistently ranks the ground truth hypothesis (the septagon) below the top, likely because the average angle errors in the drawings, and thus *never* succeeds at this conservative test, with only two edges visible. However, with more edges becoming observable, it can (and does) perform better than human recognition.

Figure 3b shows the results for question 1, over all shapes. To "normalize" for the different number of observations, we examine *the convergence rate* by looking at the area-under-the-curve for each line, for any shape, and divided it by the number of observations. Successful results, early on, result in high convergence values.



(a) Convergence achieved for the septagon shape along all of the observations.



(b) Mean convergence achieved for each shape, by each recognizer.

Fig. 3: Question 1 results. Higher values are better.

Figure 3b also shows clearly that human and ranking recognition are superior to non-ranking recognition, in all shapes but triangles. On close examination of triangles, it turns out after observing two edges, the ranking procedure was in fact generating better rankings, but was consistenly putting triangle (the correct hypothesis) in the second rank. So its score there was 0 for two edges, while the non-ranking recognizer was expected statistically to be correct at least part of the time, and thus scored better. In all other shapes, the ranking recognizer performed on par with human recognition success, slightly below it (and for pentagons, slightly above it).

To evaluate the relationship between the human results and our recognizers' we performed a z-test comparing the human results to both the ranking recognizer results, Figure 4a and the non-ranking recognizer results, Figure 4b.

	Human vs. Ranking Recognizer		
	Question 1	Question 2	Question 3
Triangles	0.1846305185	0.9598504315	0.1506408531
Squares	0.3193511582	0.9998503387	0.017189658
Pentagons	0.650604497	0.9939619089	1.7729465635E-005
Hexagons	0.0028643917	0.999827136	2.3553546891E-009
Septagons	0.3885795877	0.2922368636	1.6929433364E-006
Octagons	0.000003899	1.092129719993E-010	1.7653697948E-009

(a) Compared to ranking recognizer.

	Human vs. Non-Ranking Recognizer		
	Question 1	Question 2	Question 3
Triangles	0.0083263085	0.9853769938	0.1506408531
Squares	1.20448661E-007	0.9999999979	0.017189658
Pentagons	0	1	1.772946563E-005
Hexagons	1.70419234E-014	0.9999999922	2.355354689E-009
Septagons	2.03179482E-005	0.9998563512	1.692943336E-006
Octagons	4.30211422E-014	0.0051791896	1.765369795E-009

(b) Compared to non-ranking recognizer.

Fig. 4: Z-Test values measured when comparing the human results to the recognizers.

When choosing a significance level of 5% we can see that the values agree with the qualitative analysis of Figure 3b. For the first question we will reject H0 for the Triangle,Hexagon and Octagon shapes showing a significant difference between the ranking recognizers' results and the human results. However, for the non-ranking recognizer we will reject H0 for all of the values.

Finally, we also evaluated the results in terms of the percent of the shape that had to be disclosed to the user before a definite identification, shown in Figure 5. Here, a lower value is better. The figure shows a clear superior performance of the ranking rec-

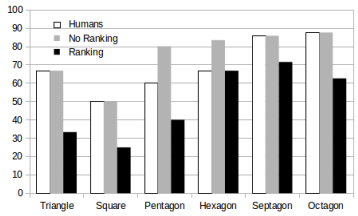


Fig. 5: Percent of shape uncovered before definite identification for each of the tested shapes. Lower percentage is better.

ognizer over human recognition and over the non-ranking recognizer, in all cases except Hexagons, where the human and ranking techniques are essentially equally successful.

Question 2: How likely is the shape? We now relax the test of accuracy a little. Rather than the conservative success/failure test of question 1, we now ask the recognizers to provide a ranked ordering of the hypotheses, and mark the rank of the correct hypothesis in their response. Thus putting the correct hypothesis at the top of the list generates a score of 1, and putting it at the bottom of list generates a score of 6 (i.e., here, a lower score is better).

While the ranking recognizer techniques always provide full rankings, the human subjects did not. Human subjects sometimes did not rank a shape, when they deemed it unlikely (but not necessarily impossible—which is why the answers to questions 2 and 3 are not complementary for humans). We thus differentiate between two cases where the correct hypothesis did not appear in the human subject's answer to question 2:

- The correct shape was not ranked, and explicitly stated in question 3 as a shape that was not a possibility. This was marked as an error. We show these errors separately in Figure 7 below.
- The correct shape was not ranked, but also not chosen in question 3. In this case we assumed the human allowed for its possibility, but dismissed it as unlikely (but not impossible). We then use the statistically expected rank of the correct shape in the fully-ranked list, over all combinations where it occupied non-ranked slots. For example, if the participant ranked three other shapes as possible but not the correct shape, then the correct shape would be given a rank of $(4 + 5 + 6)/3$.

For the non-ranking recognizer, we used the statistically expected rank, given all possible orderings of the hypotheses. For example, if the recognizer ranked two shapes as possible (with the correct shape as one of them), the new ranking of the correct shape would be calculated as $(1 + 2)/2$. If the correct shape was not ranked it was necessarily chosen as impossible and regarded as an error.

Figure 6 presents the average convergence of the ranking score of each shape along all of the observations received. The vertical axis marks the convergence (lower score here is better). The figure shows that human recognition stays fairly consistent and is mostly superior to both ranking and non-ranking recognition methods. For the Octagon

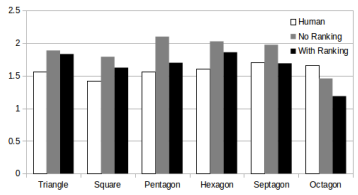


Fig. 6: Mean ranking convergence achieved in question 2 for each shape.

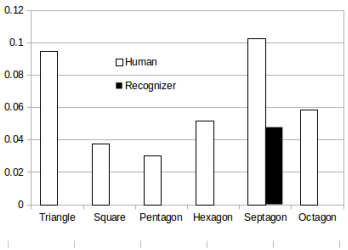


Fig. 7: The average ratio of errors received for each shape. Data range 0-1.

shape humans perform less well than both recognizers raising future questions relating to humans' perception of larger shapes in general. We can also see that the ranking recognition was ultimately better than the non-ranking recognition.

From the Z-Test results displayed in Figures 4 we can see that with a significance level of 5% we don't reject H0 for either the ranking and the non-ranking algorithms establishing a clear connection between the results.

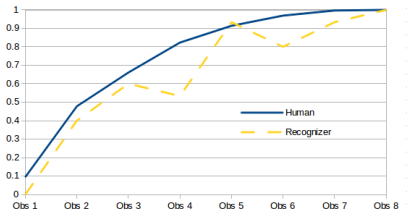
As mentioned before we excluded the errors from our result analysis and addressed them separately. Figure 7 shows the errors made by all three recognizers, i.e., correct hypotheses that have been actively disqualified by the recognizer. With the exception of septagons, only humans make such errors, as the results demonstrate. This resulted from a great deal of noise in the septagon shapes, expressed as a large difference in expected angle size. Furthermore, even in septagons, the machine shape recognizers make far less errors than humans.

Question 3: What shape couldn't it be? In the final question, we examined a separate factor that can be utilized to improve recognition. While questions 1 and 2 focused on positive recognition—the ability to correctly guess (hypothesize) at the correct goal shape, question 3 addresses negative recognition—the ability to rule out hypotheses from consideration. The two, as explained earlier, are not complementary. For instance, one could rule out 4 of 6 hypotheses, and still completely fail on questions 1 and 2 (ranking the correct hypothesis second).

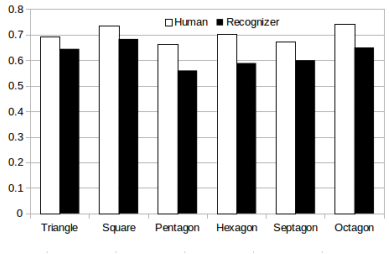
We measure performance in this question as follows. We note how many of the incorrect shapes (necessarily, 5) are ruled out in the response to question 3. This is specified in ratio form, with the worst score being $0 = 0/5$ when no shapes are ruled out, and the best score being $1 = 5/5$. Thus a higher score (1 is the maximum) is

better. And as before, with less observations this is better. Thus we again utilize the convergence measure.

For example, Figure 8a shows the results for octagons. The horizontal axis measures the number of observed edges. The vertical axis marks performance score (high is better) as described above. Because ranking has no significance in this particular question the calculation for the performance of both the ranking and non-ranking recognizer is identical. Therefore we utilize only one line (marked arbitrarily as "Recognizer") to denote their performance in the figure.



(a) Mean convergence of incorrect shapes for each observation in the octagon shape.



(b) Mean convergence achieved in question 3 for each shape.

Fig. 8: Question 3 results - negative recognition. Higher values are better.

There are several interesting observations based on this figure. First, the machine recognizers surpass human performance at one point (5 observed edges), but otherwise offer inferior (if close) performance to human recognition. Thus there is, apparently, the possibility that humans are not disqualifying hypotheses in these stages, even when they are in fact no longer relevant. Second, humans also make another kind of mistake, where they make an incorrect disqualification of an hypothesis with the first observation. Necessarily, observing a single edge, no shape hypothesis can be disqualified. Yet humans temd to jump to conclusions, eliminating at least one hypothesis from being considered.

Figure 8b presents the mean convergence for each shape in respect to Question 3. Clearly, human disqualification is faster in all cases—but this result should be taken with some caution, as we note that such early disqualification of hypotheses results in the errors discussed above (in addressing question 2).

The Z-Test results displayed in Figures 4 again agree with the convergence results. And we can see that with a significance level of 5% we reject H0 for Septagons,Hexagons and Octagons.

5 Conclusions and Future Work

We presented a mirroring approach to on-line shape recognition, where by a planner is re-used by a recognition process, allowing drawn-shape recognition by drawn-shape

planning. The approach has a number of technical advantages specific to recognition (such as no plan library and fast on-line computation with no pre-processing), but most importantly, is particularly suited to agents, where a complete agent is expected to have a planner for its own goals, and this can be utilized for recognition, without the need for a separate source of recognition knowledge.

We instantiated the shape recognition approach in the recognition of regular polygons, and evaluated the performance of different ranking and non-ranking variants of the recognizer against human subjects' recognition of scanned hand-drawn regular polygons. The evaluation utilized several different evaluation criteria. Across the board, the ranking recognition proved superior to the non-ranking recognition. In some cases, the ranking recognizer surpassed human recognition results (e.g., it required less of the polygon to be observed before settling on the correct response). However, in general the ranking recognizer performed on par, or just below, human levels of recognition. Through one of the evaluation tests (question 3) we show that humans make negative recognition mistakes, both in disqualifying hypotheses too early, or in holding on to them even once it is proven they are incorrect. However, it might be that the tendencies leading to these mistakes might also account for the better performance of humans.

As stated, the planner's input is comprised of the anchor points (initial and ending open ends of the polygon) and line parameters (slope and intercept). Because of this, translating shapes in 2D space, and rotating them, are completely transparent to the recognizer. Indeed, the results presented here are based on shapes drawn by hands in various scales, rotations, and translated. Thus we expect the mirroring approach to be particular scalable to realistic scenarios, where these transformations are to be expected.

In future work, we hope to study the differences between human and machine recognition, especially in the ability to disqualify hypotheses early on. The biases that humans exhibit may prove useful. Additionally, we are interested in finding methods for automatically calibrating the thresholds in the ranking procedure, to better handle inaccuracy and noise in perception. We also hope to be able to eventually incorporate learning into our mechanism, so as to combine the benefits of mirroring with the benefits of learning.

References

1. Amanatiadis, A., Kaburlasos, V., Gasteratos, A., Papadakis, S.: Evaluation of shape descriptors for shape-based image retrieval. Image Processing, IET 5(5), 493–499 (2011)
2. Anderson, D., Bailey, C., Skubic, M.: Hidden Markov model symbol recognition for sketch-based interfaces. In: AAAI Fall Symposium. pp. 15–21 (2004)
3. Bonchek-Dokow, E., Kaminka, G.A.: Towards computational models of intention detection and intention prediction. Cognitive Systems Research 28, 44–79 (2014)
4. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction 4(4), 253–278 (1994)
5. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Communications of the ACM 15(1), 11–15 (1972)
6. Fonseca, M.J., Jorge, J.A.: Using fuzzy logic to recognize geometric shapes interactively. In: Proceeding of The Ninth IEEE International Conference on Fuzzy Systems. vol. 1, pp. 291–296. IEEE (2000)

7. Hong, J.: Goal recognition through goal graph analysis. J. Artif. Intell. Res. (JAIR) 15, 1–30 (2001)
8. de Hoon, M.J., Imoto, S., Nolan, J., Miyano, S.: Open source clustering software. Bioinformatics 20(9), 1453–1454 (2004)
9. Jorge, J., Samavati, F.: Sketch-based interfaces and modeling. Springer (2010)
10. Jorge, J.A., Fonseca, M.J.: A simple approach to recognise geometric shapes interactively. In: Graphics Recognition Recent Advances, pp. 266–274. Springer (2000)
11. Kaminka, G.A.: Curing robot autism: A challenge. In: Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems. pp. 801–804. International Foundation for Autonomous Agents and Multiagent Systems (2013)
12. Kopp, S., Wachsmuth, I., Bonaiuto, J., Arbib, M.: Imitation in embodied communication - from monkey mirror neurons to artificial humans. Embodied Communication in Humans and Machines pp. 357–390 (2008)
13. Olsen, L., Samavati, F.F., Sousa, M.C., Jorge, J.A.: Sketch-based modeling: A survey. Computers & Graphics 33(1), 85–103 (2009)
14. Paulson, B., Hammond, T.: A system for recognizing and beautifying low-level sketch shapes using NDDE and DCR. In: ACM Symposium on User Interface Software and Technology (UIST2007) (2007)
15. Pavlovic, V.I., Sharma, R., Huang, T.S.: Visual interpretation of hand gestures for human-computer interaction: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 677–695 (1997)
16. Pellegrino, G., Fadiga, L., Fogassi, L., Gallese, V., Rizzolatti, G.: Understanding motor events: A neurophysiological study. Experimental Brain Research 91(1), 176–180 (1992)
17. Ramırez, M., Geffner, H.: Plan recognition as planning. In: International Joint Conference on Artifical Intelligence. pp. 1778–1783 (2009)
18. Ramırez, M., Geffner, H.: Probabilistic plan recognition using off-the-shelf classical planners. In: International Joint Conference on Artificial Intelligence (2010)
19. Rizzolatti, G., Fadiga, L., Gallese, V., Fogassi, L.: Premotor cortex and the recognition of motor actions. Cognitive brain research 3(2), 131–141 (1996)
20. Sadeghipour, A., Kopp, S.: Embodied gesture processing: Motor-based integration of perception and action in social artificial agents. Cognitive Computation 3(3), 419–435 (2011)
21. Sadeghipour, A., Yaghoubzadeh, R., Rüter, A., Kopp, S.: Social motorics–towards an embodied basis of social human-robot interaction. In: Human Centered Robot Systems, pp. 193–203. Springer (2009)
22. Sezgin, T.M., Davis, R.: Hmm-based efficient sketch recognition. In: Proceedings of the 10th International Conference on Intelligent User Interfaces. pp. 281–283. ACM (2005)
23. Sharon, D., Van De Panne, M.: Constellation models for sketch recognition. In: Proceedings of the Third Eurographics Conference on Sketch-Based Interfaces and Modeling. pp. 19–26. Eurographics Association (2006)
24. Tambe, M., Rosenbloom, P.: RESC: An approach for real-time, dynamic agent tracking. In: International Joint Conference on Artificial Intelligence. vol. 14, pp. 103–111 (1995)
25. Tumen, R.S., Acer, M.E., Sezgin, T.M.: Feature extraction and classifier combination for image-based sketch recognition. In: Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium. pp. 63–70. Eurographics Association (2010)
26. Ulgen, F., Flavell, A.C., Akamatsu, N.: Geometric shape recognition with fuzzy filtered input to a backpropagation neural network. IEICE Transactions on Information and Systems 78(2), 174–183 (1995)
27. Wachs, J.P., Kölsch, M., Stern, H., Edan, Y.: Vision-based hand-gesture applications. Communications of the ACM 54(2), 60–71 (2011)