Towards Sketch Recognition by Mirroring*

(Extended Abstract)

Mor Vered
Computer Science Department
MAVERICK Group
Bar Ilan University, Israel
veredm@cs.biu.ac.il

Gal A. Kaminka
Computer Science Department &
Gonda Brain Research Center
MAVERICK Group
Bar Ilan University, Israel
galk@cs.biu.ac.il

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous

Keywords

On-line plan recognition, sketch recognition/planning, mirroring

1. INTRODUCTION

Humans increasingly 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. To recognize shapes in sketches, most existing work focuses on offline (post-drawing) recognition methods, trained on large sets of examples . 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.

We present an *online* shape recognizer that identifies multi-stroke geometric shapes without a plan library. Inspired by mirroring processes hypothesized to take place in socially-intelligent brains , the recognizer uses a shape-drawing planner for drawn-shape recognition. It is a form of plan recognition from planning .

Mirror neurons have first been discovered in the early 90's . These neurons were seen to fire both when a monkey manipulated an object and also when it saw another animal manipulate an object. Recent neuro-imaging 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. Indeed, 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 other agents from a series of observations of the other agents' actions.

There are many advantages to a mirroring-based shape recognizer. Some immediate technical advantages from a recognition point of view include: (1) no need for storing a library of shapes to be matched against drawings; and (2) fast on-line recognition.

However, the most important 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.

*Supported in part by ISF Grant #1511/12. Thanks to K. Ushi.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

2. USING A SHAPE-DRAWING PLANNER TO RECOGNIZE DRAWN SHAPES

We demonstrate our approach on the problem of shape recognition of hand drawn shapes on paper. To perform the recognition, our system will utilize its own existing, shape-drawing planner instead of referring to an existing plan library.

2.1 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, is a set of 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, given the stream of observations, *as early as possible*. Each observation is an edge of the polygon, connected to a previously observed edge. The output of the shape recognizer is an ordered list, of all of the goals (i.e., shapes) that match 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 matches the first observations seen. This is done incrementally, so that 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 ranked list of candidate goals.

We build on the principles of plan-recognition by planning. The 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 this approach 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.

Therefore, instead of explicitly folding observations into the initial state, we do so implicitly. In our case, to generate a new shape, 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.

2.2 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. We describe these below, briefly.

Input Preparation Component

The goal shape and observations input are fed into the *Input Preparation* component, whose task is to prepare the input to be sent to the *Planner*. This is the key step in the mirroring approach. The *Input Preparation* component incorporates the available observation history into the goal being sent to the *Planner* by creating a new goal, that removes the edges already seen from the original goal, and is comprised only of the remainder of the polygon, i.e., the part expected to be completed if the observations are to be a part of intended goal.

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 *Input Preparation* component 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 size 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.

Planner

Each possible goal is sent to the *Shape Planner*. 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.

If the planner is unable to generate a plan for drawing the polygon remainder, it issues an error which indicates that it is not possible to draw the specific goal. This indicates that the goal does not match 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 or relevance.

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. The idea is to measure the geometric errors for each possible plan, between the instantiated shape, derived by the hypothesized goal and the corresponding shape derived by the original goal.

3. EXPERIMENTS AND CONCLUSIONS

We conducted a series of preliminary experiments utilizing a drawing planner for regular polygons. This allows up to explore how the recognition process works, and how the different components interact. The basis for the experiment was a data-base of scanned hand-drawn regular polygons. 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. We ran these shapes through our recognizer and on a group of human participants.

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. However, in general the ranking recognizer performed on par, or just below, human levels of recognition.

We look forward to conducting further experiments, in order to be able to draw further lessons regarding the recognition process both in our model and in the human context.

REFERENCES

[1] G. A. Kaminka. Curing robot autism: A challenge. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 801–804. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[2] L. Olsen, F. F. Samavati, M. C. Sousa, and J. A. Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, 33(1):85–103, 2009.

[3] G. Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti. Understanding motor events: A neurophysiological study. *Experimental Brain Research*, 91(1):176–180, 1992.

[4] M. Ramırez and H. Geffner. Plan recognition as planning. In *International Joint Conference on Artifical Intelligence*, pages 1778–1783, 2009.