Human and Robot Perception in Large-scale Learning from Demonstration

Christopher Crick
Brown University
Department of Computer Science
Providence RI 02912
chriscrick@cs.brown.edu

Sarah Osentoski
Brown University
Department of Computer Science
Providence RI 02912
sosentos@cs.brown.edu

Graylin Jay
Brown University
Department of Computer Science
Providence RI 02912
tjay@cs.brown.edu

Odest Chadwicke Jenkins
Brown University
Department of Computer Science
Providence RI 02912
cjenkins@cs.brown.edu

ABSTRACT

We present a study of using a robotic learning from demonstration system capable of collecting large amounts of human-robot interaction data through a web-based interface. We examine the effect of different perceptual mappings between the human teacher and robot on the learning from demonstration. We show that humans are significantly more effective at teaching a robot to navigate a maze when presented with information that is limited to the robot's perception of the world, even though their task performance measurably suffers when contrasted with users provided with a natural and detailed raw video feed. Robots trained on such demonstrations learn more quickly, perform more accurately and generalize better. We also demonstrate a set of software tools for enabling internet-mediated human-robot interaction and gathering the large datasets that such crowdsourcing makes possible.

Categories and Subject Descriptors: I.2.9 [Computing Methodologies, Artificial Intelligence, Robotics]Operator interfaces

General Terms: Experimentation, Human Factors

Keywords: Crowdsourcing, interface design, learning from demonstration

1. INTRODUCTION

Although the study of human-robot interaction has made great strides in recent years, robots are still far from being able to interact autonomously with humans in unstructured human environments. If we are to bring robots into our homes and offices, it is not likely that their builders will be able to engineer appropriate behaviors for every circumstance they might encounter, nor will the robots come with convenient Wizards of Oz to teleoperate them at every turn.

In order to perform multiple complex unforeseen tasks, robots will need to learn tasks taught to them by users who are not themselves roboticists or programmers. One promising way to accomplish this learning is to demonstrate desired tasks to a robot observer. This approach has several advantages: the teacher needs no expertise in programming the robot, nor any pedagogical insight into the learning process. Nothing is required of the user beyond the ability to complete the task, in some fashion that the robot can interpret.

However, this places the onus on inference by the learner. Interpreting and learning from the demonstration is by no means straightforward. One particular difficulty is the "correspondence problem" [16], the fact that a human teacher and robotic student have very different physical affordances and sensory modalities. The problem of how to establish a trustworthy mapping between the two is very much still an open question.

This work studies one aspect of this problem, the mismatch between human and robot perceptual abilities. Although a robot with a camera may be able to detect, identify and locate a few particularly salient visual cues, no existing algorithms can possibly make sense of all of the context, spatial relationships, identities and qualities of every object in a scene, all of which are apparent at a glance to a human. Thus, a robot's attempt to learn a policy may be thwarted by the simple fact that the human makes decisions based on features of the environment which the robot's sensory apparatus is currently inadequate to perceive.

Perhaps this perceptual mismatch will eventually be rendered moot by the gradual advance of robotic sensors and processing techniques. For now, however, the answer may well lie in manipulating the other side of the problem. Suppose we limit human perception to match the robot's. Humans are often able to make sense of profoundly impoverished stimuli [10]. They may be able to produce more reliable demonstrations for robot learning if they are forced

HRI'11, March 6–9, 2011, Lausanne, Switzerland.

to operate in a robot-like sensory environment. Conversely, their performance may be so hindered by the lack of familiar sensory information that their actions are of no help at all.

We present evidence that suggests that users are indeed able to provide reliable demonstrations for robot learning when operating within such an impoverished sensory environment. 132 subjects were given the opportunity to interact online with a robot, teaching it via teleoperation to maneuver through a simple maze and come to a stop at a goal. If they were only shown a visualization of augmented-reality tags, recognized as landmarks by the robot's visual system, their progress through the maze was much slower and less adroit than if they were given a full video representation of the scene. Nevertheless, the robot was able to learn far faster and more reliably from the demonstrators who shared the robot's perceptual environment, and was even able to generalize the lessons to related but novel tasks more successfully.

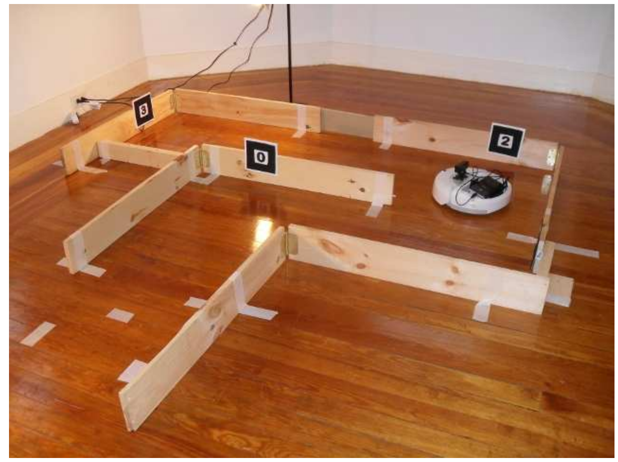

Figure 1: A demonstrator shows the robot how to navigate the maze via teleoperation

2. RELATED WORK

Learning from demonstration is an approach to robot programming in which the policy is learned using demonstrations provided by a teacher [5, 2]. Learning from demonstration can be formulated in a variety of ways; teachers and learners may employ a variety of mechanisms. Learning from demonstration has been examined as a supervised learning problem, including Gaussian mixture models [8, 9], a comparison of parametric and nonparametric classifiers applied to the pendulum swing-up task [3], and regression applied to robot soccer tasks [12]. As a basis for reinforcement learning, demonstration has been used for value function intialization [24] and inverse reinforcement learning [1, 27]. Demonstration has also motivated learning in behavior-based approaches [17] and in socially-driven interactions [26, 6, 21]. It has been used as a mechanism for initializing and bootstrapping subsequent machine learning [15] in a different paradigm. In this paper, we use decision trees to learn a policy from demonstrations provided using remote teleoperation of a robot over the web. Decision trees are a core machine learning technique known for producing classification schemas that are particularly straightforward for humans to understand [20]. This is not the first work to use decision trees for learning from demonstration; early work used them in learning to fly a simulated airplane [22].

This paper focuses on allowing a large number of users to provide demonstrations to a robot over the web and evaluating what type of visualizations result in the best policy. A recent trend in machine learning has examined the use of truly large data sets for learning rather than attempting to generalize from a small amount of data. Researchers in data mining and machine translation have able to take advantage of Google's index of billions of crowdsourced documents and trillions of words to show that simple learning algorithms that focus upon recognizing specific features outperform more conceptually sophisticated ones [13]. We conjecture that similar successes would be observed if large amounts of data could be collected for learning from demonstration. Human-robot interaction studies, to date, more often number in the dozens of subjects [4]. Opening up robots to the vast number of users on the world wide web provides the opportunity to gain a large number demonstrations from many different users.

The robotics community has made a few forays into human robot interaction over the internet. Goldberg et al. placed a robot in a garden and allowed users to view and interact with the robot over the web. Users were able to plant seeds, water, and monitor the garden [11]. Taylor and Trevelyan created a remote lab in which users perform tasks involving brightly colored blocks [25]. Schulz et al. examined the use of web interfaces to remotely operate mobile robots in public places [23]. This worked focused on letting remote users interact with humans within the robots' environment and did not examine the effect of the visualizations in a learning task. Burgard and Schulz have explored handling delay in remote operation/teleoperation of mobile robots using predictive simulation for visualization [7]. This work examined how robots could be controlled when there was a large delay in the visualization presented to the user. The work we present in this paper may result in less delay since users are presented with a reduced visualization; however the main motivation is to examine if the resulting demonstrations in the reduced feature spaces are better for learning.

3. ROBOT

The robot used in the experiments is the iRobot Create platform, on which is mounted a FitPC2 small-form-factor computer and a Sony PS3 Eye camera capable of running at 320x240 resolution at 120 Hz. The computer maintains a wireless connection to the internet for servicing rosjs connections (see Section 4). The various robot services communicate with each other via messages passed by the Robot Operating System (ROS) [19].

The system runs our ROS-compatible drivers for the Create platform and for the camera, as well as our extensions to the AR Toolkit [14], allowing the robot's visual system to detect, localize identify augmented reality tags on which it has trained.[1]

[1]Toward the promotion of reproducibility in robotics, we have made these software packages available at http://code.google.com/p/brown-ros-pkg. Our available tools include drivers for the Create and USB webcameras, the AR recognition package, rosjs (described in Section 4), and others.

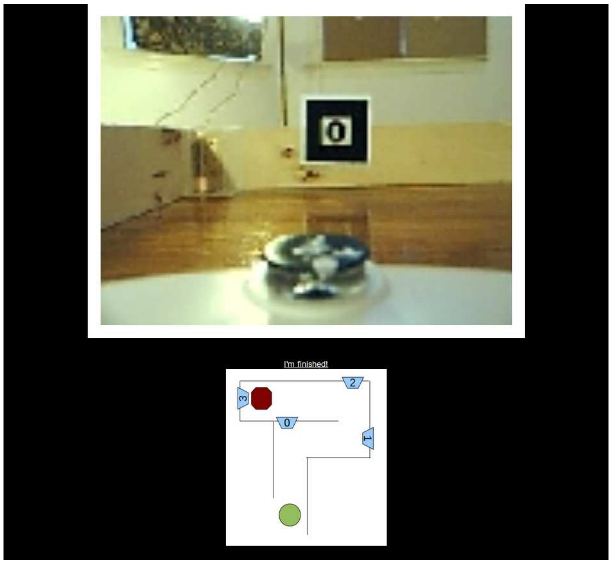

Figure 2: The user's view in the video condition

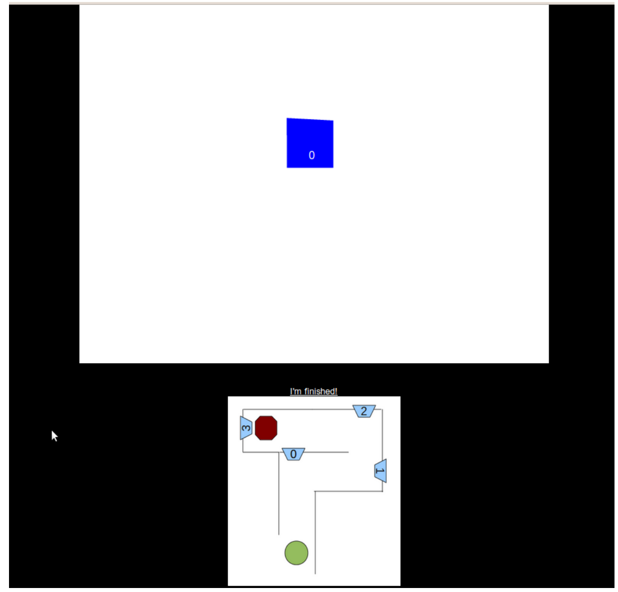

Figure 3: The user's view in the tag-only condition

The robot is able to move forward and backward and to rotate. The frame rate of its camera is sufficient to minimize blur induced by its motion, which aids in its ability to search for and identify augmented reality tags placed in its environment (depicted in Figure 2). If such a tag appears in its visual field, it is able to identify the pixel locations of the tag's corners and, usually, the tag's unique identification. In addition, the robot can use the observed size of the tag to generate a rough guess of its distance. The visual processing involved in recognizing these tags is comparatively robust and reliable. They are very salient, and their representation is low-dimensional and computationally tractable, in contrast to raw visual pixel arrays.

The tags, along with a bump sensor that detects collisions with walls, represent the totality of the perceptual space available to the robot for learning. We show that humans can operate as competent teachers within this space, as well.

4. WEB INTERFACE

This work is the first experimental deployment of our rosjs technology [18] developed in part to facilitate large-scale distributed data collection for human-robot interaction. rosjs is a lightweight Javascript interface to ROS which exposes a robot's data streams and controllers as web services accessible anywhere via the Internet, as well as providing visualization tools and security mechanisms. It allows robot application developers and researchers to produce robot controllers and interfaces in the same manner as creating web content – developers author webpages to interact with remote robot services. In addition, rosjs allows users to access and run robot applications without any installed software beyond a simple web browser, in hopes of expanding the potential pool of robot users and research subjects as wide as possible.

132 individual subjects participated in the robot training task from all over the world. They were invited to participate as many times as they liked; in all 276 human demonstration trials were collected. In the context of human-robot interaction studies, this is a very large subject pool; we hope that making rosjs and similar services available to the community will encourage many more such studies.

To account for possible habituation effects among the human demonstrators, all of our data analyses were performed on both the data as a whole and the subset of data that represented only the individual's first exposure to the experimental setting. The only measurable difference found was in the demonstration failure rate, to be discussed in Section 5. For this reason, no distinction is otherwise made between a subject's first trial and any subsequent ones in which she may have participated.

Upon connecting to the test website, subjects read a description of the experiment and what it was intended to accomplish. They were encouraged to drive the robot through a maze as quickly and deftly as possible, and told that the robot would be observing their actions and learning from their example. They were instructed to click a link and begin the experiment when ready.

Each subject was randomly assigned one of two conditions:

1. **Video View:** A camera view where the subject received a live color video feed from the robot's camera, and could see the floor, the walls of the maze, the lighting and the actual augmented-reality tags marking each leg of the maze (Figure 2), or

2. **Tag-only View:** The only information the subject received was a visualization of the robot's estimate of the position and identity of any of the four tags currently within the robots field of view (Figure 3). This representation took the form of blue polygons positioned in the appropriate place on a white visual field, along with a number matching the robot's estimation of the tag's identity.

In addition, the subjects received information on whether the robot had hit an obstacle – upon activation of a bump

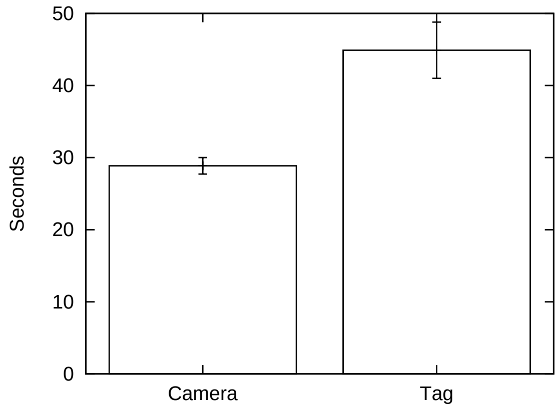

Figure 4: Time for humans to complete the maze. NB: For all figures in this paper, error bars represent 95% confidence intervals

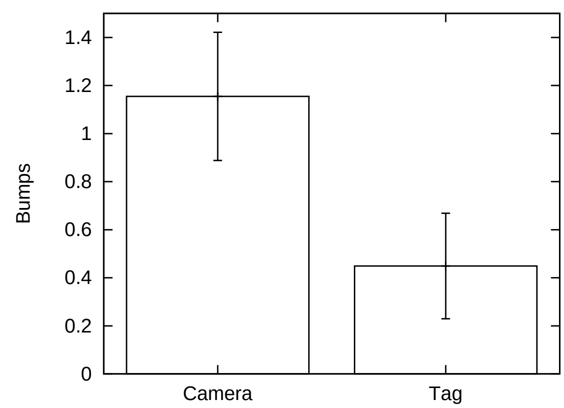

Figure 5: Collisions per trial

sensor, the screen image would flash red. Finally, each subject had available a map of the maze marked with the position of the various AR tags, to help with orientation and navigation.

Subjects controlled the robot using the arrow keys on their keyboard. Upon finishing, they were instructed to click a link ending the trial, whereupon they were told whether they had successfully navigated the maze, as well as their performance in terms of time and number of collisions.

5. HUMAN PERFORMANCE

As expected, humans were much more adept at navigating the maze in the video condition than in the tag-only one. On average, those completing the maze with intact vision were able to do so 16.03 seconds, or 36.3%, faster than those who could only make out disembodied blue squares (see Figure 4).

Somewhat surprisingly, though, while users with video were far faster at negotiating the maze, they were in fact *less* able to maneuver through the walls without running into them. Users who were not even able to see the walls hit them significantly less often. As shown in Figure 5, drivers with cameras were more than twice as likely to crash into a wall – 1.15 crashes per demonstration as opposed to 0.45. Part of this might be explainable by the simple fact that, since they were driving faster, they were more limited by their reaction time or by network latency. Indeed, transmitting full video puts far more burden on a network connection than does sending a few coordinates with which rosjs can reconstruct a visualization.

However, it is also the case that users with video were simply able to engage in the environment more fluidly and skillfully, in a manner more akin to controlling a car in a video game. Users in the video condition were more apt to send the robot along curved trajectories; they commanded simultaneous forward and turning motion 22.5 *times* as often as users navigating with only tags. Reduced to picking their way from tag to tag, demonstrators in the latter condition used curves only 1.62% of the time – the rest was given over to straight lines and rotating in place. This not only kept them from hitting walls, but as we shall see, proved far better pedagogy for the robot.

The final performance metric we applied to the human demonstrators was to analyze their rate of failure to navigate the maze at all. We expected that the unfamiliar and difficult-to-interpret tag condition would lead to more users abandoning the task. It did: 22.83% of tag trials failed as opposed to 17.44% of camera trials (see Figure 6), but the difference was not significant.

Overall trial failure was the only measure on which the user's familiarity with the task had a significant effect. Just over twice as many (38 to 17) of the failures came from people who were trying the task for the first time – a difference right at the 95% confidence threshold. Of those 38, 26 returned to participate again.

We can point to several reasons for failure. Users failed to consider the directions, or decided they were not interested in participating after all, or attempted to move around and became disoriented and lost. In addition, six of the failures were due to hardware problems – five from the robot's temporary loss of network connectivity, and one instance of the robot actually bumping into a maze wall hard enough to tear it loose from its moorings and block the path to the goal. If the robot determined that the demonstration was a failure – that is, it did not end with the robot stopped in the goal area, having identified the tag marking the end of the final maze leg – it marked the data as invalid and did not consider it further. Invalid trials played no part in the learning process to which we now turn.

6. DECISION TREES

Let $\mathbb{X}$ be a set of observation vectors and $X_j \in \mathbb{X}$ a set containing all of the observations of the jth variable in the vector, namely, one of:

- *last_tag_seen:* The identity of the most recent tag observed
- *tag_visible:* Whether that tag is currently visible
- *tag_x_coord:* The tag's most recently observed horizontal pixel coordinates (due to the fact that the camera has fixed tilt, the y coordinate is purely a function of distance, and so can be ignored)

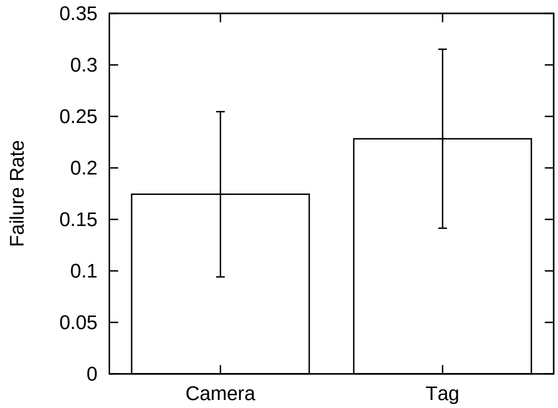

Figure 6: Percentage of trials which failed to reach goal

- *tag_distance:* The tag's most recently observed distance
- *bumping:* Whether or not the robot is currently bumping into a wall (as reported by a physical bumper on the front of the robot)

Let $\bar{X} \in \mathbb{X}$ be a single observation vector containing one instance of each of the variables above, and let Y be the set of motor commands associated with each of those vectors, as demonstrated by human teachers in one of the two experimental conditions. Let $y \in Y$ be the discrete individual motor states such as forward, forward-and-left, right, backward, etc.

Given $\mathbb{X}$ and Y, the learning problem facing the robot is to derive a policy $\pi : \bar{X} \rightarrow y$ which will allow a robot to navigate a maze successfully.

Having observed 132 people demonstrating 276 trials, the robot can bring 79,726 learning examples to bear on the problem of constructing π. Decision trees are a well-understood and established classification method for offline learning.[20] They run efficiently even on large datasets ($O(n \log n)$ to learn, $O(\log n)$ to execute, on average) and have the distinct advantage that the classification process is transparent to observers. Other learning algorithms could as easily have been chosen; our work is investigating the human component of building these training sets, rather than the induction process itself. We do, however, report experiments which suggest that the perceptual mapping effects which we are considering are robust to certain changes in how robot policies are learned (see Section 7 and Figure 9).

In building a useful and compact decision tree, the system establishes sequences of decision nodes, each of which should be maximally informative. This notion is captured in the concept of information gain IG_j (also known as the Kullback-Leibler divergence), a function of how much the entropy of the remaining dataset is reduced when one of the variables has been decided. If $\mathbb{X}$ is the set of random variables representing the system's input, $X_j \in \mathbb{X}$ is the jth input variable, and Y is the random variable associated with the output, the information gain is defined as:

$$IG_j = H(Y) - H(Y|X_j) \tag{1}$$

$H(Y)$ is the entropy before the decision is made, while $H(Y|X_j)$ is the entropy once we know the value of X_j. In information-theoretic terms, entropy is based on the surprise value of the remaining data; if the dataset becomes much more predictable once a decision has been made, then the system has moved from a condition of high entropy to low, and information gain is correspondingly large. Entropy is a function of the probabilities of variable outcomes within a set of data. If $Y = \{y_1, y_2, \ldots, y_n\}$, the set of all possible decisions, then $P(y)$ is the probability of a particular choice $y \in Y$, and the entropy of Y is defined as:

$$H(Y) = -\sum_{y \in Y} P(y) \log P(y) \tag{2}$$

In the case of discrete variables such as *bumping* or *tag_visible*, conditional entropy is defined as

$$H(Y|X_j) = -\sum_{y \in Y} P(y) \sum_{x \in X_j} P(y|x) \log P(y|x) \tag{3}$$

In the case of continuous variables such as *tag_distance*, the process is slightly more complicated. The learner must find a split point t that maximizes the information gain for a decision based on this variable:

$$\begin{aligned} H(Y|X_j, t) = \\ \max_t H(Y|X_j < t)P(X_j < t) + \\ H(Y|X_j \geq t)P(X_j \geq t) \end{aligned} \tag{4}$$

Finding the value t effectively transforms the continuous variable into a binary one: is the value greater than or less than t?. Information gain then straightforwardly becomes

$$IG_j = H(Y) - H(Y|X_j, t) \tag{5}$$

The algorithm to build the decision tree proceeds as follows.

function Build-Decision-Tree($\mathbb{X}, Y$)
  if $\exists y \in Y$s.t. $\frac{|y|}{|Y|} \geq$ Min-Majority **then**
    return new leaf node labeled y
  end if
  Choose X_j (and t if necessary) to maximize IG_j
  tree $\leftarrow$ new decision node labeled X_j
  for all values $x_j \in X_j$ **do**
    $\hat{\mathbb{X}} \leftarrow$ elements of $\mathbb{X}$ where $X_j = x_j$
    $\hat{Y} \leftarrow$ elements of Y corresponding to members of $\hat{\mathbb{X}}$
    Attach subtree Build-Decision-Tree($\hat{\mathbb{X}}, \hat{Y}$) to tree, with edge labeled x_j
  end for
  return tree
end function

The standard algorithm sets the Min-Majority variable above to 1, meaning that every decision path must end at a decision supported unambiguously by the data. With noisy data, especially when a decision tree is constructed from continuous variables such as distance and position in the visual field, it is not uncommon for different split points of the same variable to be maximally informative several times in succession, or at various points along a particular path through the tree. Such trees can grow very large indeed. Indeed, the decision tree induced from the full set of tag training data is made up of 10,957 nodes, while the tree representing the camera data likewise contains 12,998. This

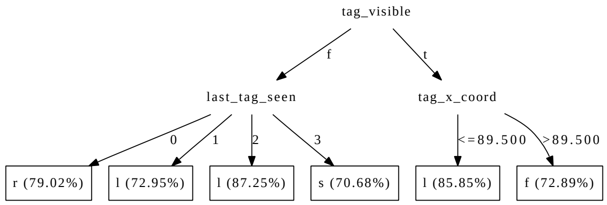

Figure 7: Minimal decision tree built from tag data

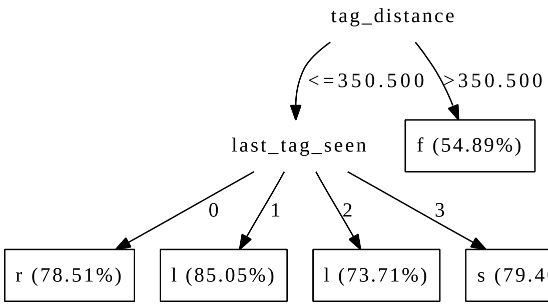

Figure 8: Minimal decision tree built from camera data

does not pose a responsiveness issue for the robot – each decision in such a tree (if reasonably balanced) requires about ten steps – but it does raise the issue of whether the model overfits its training data. To test this, we investigated the effect of constraining the decision tree size on the robot's performance.

A number of different techniques exist to enforce the building of small, generalizable trees. One simple way is to adjust the minimum majority threshold. When this threshold is reduced far enough, the trees become small enough for inspection and inclusion in a paper. A very compact decision tree built from the tag data is still quite predictive: the actions specified in the tree are correct at least 70% of the time (Figure 7). In order to produce a tree of roughly comparable size (Figure 8), the camera-trained tree must accept a much lower classification rate: 54%. However, the similarity of the two trees at this level of detail is striking. One uses tag distance while the other uses tag visibility, but the overall process is similar to what an engineer might have designed: go forward if you're far away, while turning or stopping appropriately at each marker in turn. The tag-based learner is also able to turn left if it strays too far off course, though it does not learn the symmetric course correction to the right.

7. ROBOT PERFORMANCE IN LEARNING FROM DEMONSTRATION

How large an effect does decision tree size have on the robot's ability to learn an effective maze-navigation policy? More importantly, how quickly and easily and generally does the robot learn when taught in its own sensory domain, as opposed to one more natural for a human? We now turn to the final and most important set of results presented in this paper: the robot's ability to translate teaching into action and successfully perform tasks.

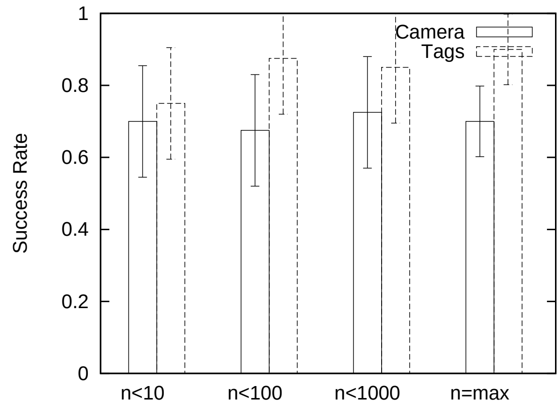

Figure 9: When forced to learn the simplest possible trees, both camera and tag data produce similar structures, and produce similar performance. As trees are allowed to grow, the tag-trained ones become marginally more effective, while the camera-trained ones do not change at all.

To answer these questions, we performed thousands of trials in the following manner. The robot was started in approximately the same position each time, facing the tag indicating the first turn about 1.3 meters away. This was the same starting position used by the human demonstrators. The robot built a decision tree out of the data it was provided, and then began to navigate. A trial was considered successful if the robot stopped in the correct goal position, facing the final tag and less than 0.4 meters away – again, the same criteria as applied to the demonstrators. If, on the other hand, the robot stopped elsewhere, or began performing repetitive sequences of actions which did not make progress toward the goal (such as spinning in place or driving into a wall), the trial was marked a failure.

To begin with, we consider the question of decision tree size and generality. Taking the entire dataset for each experimental condition, we allowed the robot to build a maximally-informative tree of a certain size. When limited to ten or fewer nodes, the learner produced the trees shown in Figures 7 and 8. The robot also built trees of ≤ 100 and ≤ 1000 nodes, as well as full, unpruned trees (Figure 9). The tree size had a minor effect on the performance of a tag-trained robot: the robot made it through the maze more often when the tree was allowed to grow to unlimited size. The tree size had no discernable effect on the camera-trained robot. These results suggest that data overfitting in this scenario is not a major concern.

The previous experiment provided the first indication that humans produced better training data for a robot when given only the sensory information accessible to the robot. However, the camera-trained robot performed nearly as well; it required two hundred trials to show a very significant difference ($p < 0.01$). A large enough quantity of data can train a competent learner even if the quality of the data is suboptimal.

The quality of the training data has a larger effect when the corpus size is small. We investigated how quickly humans could teach the maze task in each test condition by

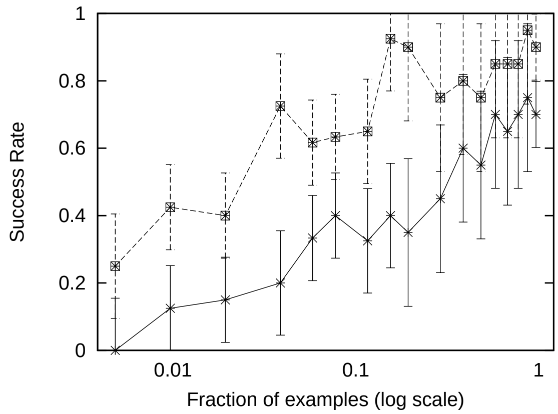

Figure 10: Robot performance as a fraction of training trials seen. Top line represents a robot trained on tag-only data; bottom line is a robot trained by people with access to video. 1320 total trials.

Figure 11: A robot picks its way through a different maze from the one it was taught to navigate.

manipulating the quantity of data a robot could use for learning. For each trial, the robot was given access to a certain percentage of the training corpus, and built its decision tree based only on that data. The data that would be made available was selected at random from the entire set, a different random selection for each individual trial. Because the largest effects were seen with small percentages of training data, we ran the majority of trials with fewer than 20% of the test trials available to the robot. For this reason, Figure 10 (as well as Figure 12, see below) is charted on a logarithmic scale.

With examples from only one or a few demonstrations, the difference in effective learning from the two datasets is striking. Having learnt from 4% of the data – four or five demonstrations, on average – a robot trained by teachers with access to video enjoyed less than a third of the success of a robot whose human demonstrators shared the robot's sensory environment. The performance premium for tag training remains significant ($p > 0.95$) and is usually

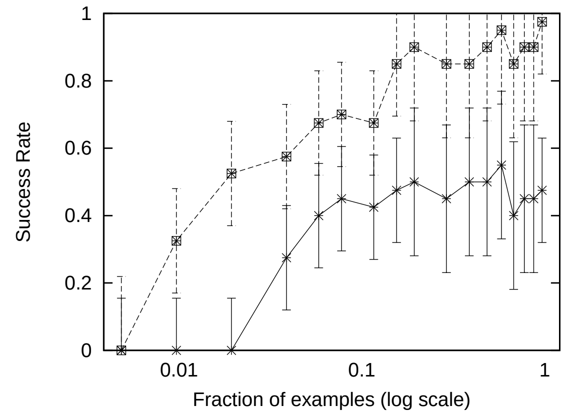

Figure 12: Robot performance in a new, unlearned environment, as a function of the fraction of training trials seen. As in Figure 10, top line represents tag training data, bottom line is video. 900 total trials. New maze environment shown in Figure 11.

very significant ($p > 0.99$) on training from 0.5% through 20% of data. Enough trials were run with 100% of the data to show significance at $p > 0.99$ at that point as well. Intermediate results from 20% to 90% are suggestive of a gradual upward trend maintaining the performance gap between the two training sets, but we did not perform enough trials at those points to obtain significance.

Finally, we investigated how well these two training paradigms generalized to similar but not identical tasks. We constructed another maze, shown in Figure 11, for the robot to attempt to navigate. Although getting through the maze required the same qualitative sequence of steps – the tags were in the same order and the turns were in the same direction as before – the maze's layout was otherwise very different. Right angles were replaced by acute and obtuse ones, and the lengths of each maze leg were changed. Once again, we allowed the robot to learn on successively larger fractions of the training data. The results are summarized in Figure 12.

In this case the difference between the two teaching environments was even greater. Robots in the camera learning condition were unable to negotiate the maze even once until they had seen four or five demonstrations' worth of examples. Except at 0.5%, at which point neither robot succeeded, the tag-trained robot significantly outperformed the other at every single measured point. In addition, the camera-trained robot mostly underperformed in relation to its success on the original maze. In contrast, the tag-trained robot's performance proved quite robust to the change in environment, performing every bit as well in the novel maze.

For the trials in both the original maze and the novel one, we also collected data on how quickly the robot was able to navigate the maze. The results (unpictured) show that the mean duration of a trial stayed more or less constant throughout most of the demonstration sample sizes. However, the smallest training sets (.5% and 1%) were slightly higher, probably owing to a few trials that succeeded in the end despite very inefficient motions (such as turning 270 degrees left rather than 90 degrees right to make a turn). In addition, the camera-trained trials were slightly though not significantly faster, (though they were significantly less likely to make it through the maze at all).

8. CONCLUSION

A robot's ability to act appropriately depends critically on its ability to make sense of the situation in which it finds itself. If humans are to train robots effectively, we must deal with the mismatch between the human and robot capacity to interpret tasks and environments. Bringing robots up to our level is a worthy goal at which researchers are making steady progress, but we have demonstrated in this paper that it is also helpful for humans to join robots at their level. Humans are adaptable and clever; they can deal with it. And our robots will learn better for the effort.

9. REFERENCES

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004.

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

[3] C. G. Atkeson and S. Schaal. Robot learning from demonstration. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, 1997.

[4] C. L. Bethel and R. R. Murphy. Use of large sample sizes and multiple evaluation methods in human-robot interaction experimentation. In *AAAI Spring 2009 Symposium: Experiment Design for Real-World Systems*, 2009.

[5] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. *Handbook of robotics*, chapter Robot programming by demonstration, pages 1371–1394. Springer, 2008.

[6] C. Breazeal, A. Brooks, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, A. Lockerd, and D. Chilongo. Tutelage and collaboration for humanoid robotics. *International Journal of Humanoid Robotics*, 1(2):315–348, June 2004.

[7] W. Burgard and D. Schulz. *Beyond webcams: an introduction to online robots*, chapter Robust visualization for online control of mobile robots, pages 241–258. MIT Press, 2002.

[8] S. Calinon and A. Aude Billard. Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the 2nd Conference on Human-Robot Interaction (HRI)*, 2007.

[9] S. Chernova and M. Veloso. Confidence-based policy learning from demonstration using gaussian mixture models. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2007.

[10] C. Crick and B. Scassellati. Controlling a robot with intention derived from motion. *Topics in Cognitive Science*, 2:114–126, 2010.

[11] K. Goldberg, H. Dreyfus, A. Goldman, O. Grau, M. Gržinić, B. Hannaford, M. Idinopulos, M. Jay, E. Kac, and M. Kusahara, editors. *The robot in the garden: telerobotics and telepistemology in the age of the Internet*. MIT Press, 2000.

[12] D. H. Grollman and O. C. Jenkins. Sparse incremental learning for interactive robot control policy estimation. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, May 2008.

[13] A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, March/April:8–12, 2009.

[14] H. Kato and M. Billinghurst. Developing ar applications with artoolkit. In *ISMAR*, 2004.

[15] J. Kober, B. Mohler, and J. Peterson. *From motor learning to interaction learning in robots*, chapter Imitation and reinforcement learning for motor primitives with perceptual coupling, pages 209–226. Springer, 2010.

[16] C. L. Nehaniv and K. Dautenhahn. *The correspondence problem*, chapter 2, pages 41–61. MIT Press, 2002.

[17] M. Nicolescu and M. J. Matarić. Natural methods for robot task learning: instructive demonstration, generalization and practice. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2003.

[18] S. Osentoski, G. Jay, C. Crick, and O. C. Jenkins. Brown ros package: reproducibility for shared experimentation and learning from demonstration. In *AAAI-10 Robot Workshop*, 2010.

[19] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. Ros: an open-source robot operating system. In *Proceedings of the Open-Source Software Workshop of the International Conference on Robotics and Automation*, 2009.

[20] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[21] P. Ruvolo, J. Whitehill, M. Virnes, and J. Movellan. Building a more effective teaching robot using apprenticeship learning. In *Proceedings of the 7th International Conference on Development and Learning (ICDL)*, 2008.

[22] C. Sammut, S. Hurst, D. Kedzier, and D. Michie. Learning to fly. In *Proceedings of the Ninth International Conference on Machine Learning*, 1992.

[23] D. Schulz, W. Burgard, D. Fox, S. Thrun, and A. B. Cremers. Web interfaces for mobile robots in public places. *IEEE Robotics and Automation Magazine*, 7:48–56, 2000.

[24] W. D. Smart and L. P. Kaelbling. Effective reinforcement learning for mobil robots. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[25] K. Taylor and J. Trevelyan. A telerobot on the world wide web. In *National Conference of the Australian Robot Association*, 1995.

[26] A. L. Thomaz and C. Breazeal. Transparency and social guided machine learning. In *Proceedings of the 5th International Conference on Development and Learning (ICDL)*, 2006.

[27] B. D. Ziebart, A. Maas, J. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2008.