The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high-quality educational resources for free. To make a donation or view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu.

STEFANIE TELLEX: So today I'm going to talk about human robot collaboration. How can we make robots that can work together with people just as if they were another person and try to achieve this kind of fluid dynamic that people have when they work together? These are my human collaborators. This work is done by a lot of collaborative students and postdocs. So we're really in an exciting time in robotics because robots are becoming more and more capable and they're able to operate in one structured environment.

Russ gave a great talk about Atlas doing things like driving a car and opening doors. This is another robot that I've worked with. A robotic forklift that can drive around autonomously in warehouse environments. It can detect where pallets are, track people, pick things up, put things down. And it's designed to do this in collaboration with people who also share the environment.

There's robots that can assemble IKEA furniture. This was Ross Knepper and Daniela Rus at MIT that I worked with to do this. So they made this team of robots that can autonomously assemble tables and chairs that are produced by IKEA. And what would be nice is if people can work with these robots. Sometimes they encounter failures, and the person might be able to intervene in a way that enables the robot to recover from failure.

And kind of the dream is robots that operate in household environments. So this is my son when he was about nine months old when we shot this picture with a PR2. You'd really like to imagine a robot-- Rosie the robot from the *Jetsons* that lives in your house with you and helps you in all kinds of ways. Anything from doing the laundry to cleaning up your room, emptying the dishwasher, helping you cook. And this could have applications for people in all aspects of life. Elders, people who are disabled, or even people who are really busy and don't feel like doing all the chores in their house.

So the aim of my research program is to enable humans and robots to collaborate together on complex tasks. And I'm going to talk about the three big problems that I think we need to solve

to make this happen. So the first problem is that you need to be able to have a robot that can robustly perform actions in real-world environments. And we're seeing more and more progress in this area, but the house is kind of like this grand challenge. And John was talking about kind of all these edge cases.

So I'm going to talk about an approach that we're taking to try to increase the robustness and also the diversity of actions that a robot can take in a real-world environment by taking an instance-based approach. Next, you need robots that can carry out complex sequences of actions. So they need to be able to plan in really, really large combinatorial state-action spaces. There might be hundreds or thousands of objects in a home that a robot might need to manipulate and depending on whether the person is doing laundry or they cooking broccoli or are they making dessert. The set of objects that are relevant that are useful that the robot needs to worry about in order to help the person is wildly different. So we need new algorithms for planning in this really large state- action space.

And finally, the robot needs to be able to figure out what people want in the first place. So people communicate using language, gesture, but also just by walking around the environment and doing things that you can infer something about what their intentions are. And critically, when people communicate with other people, it's not an open loop kind of communication. It's not like you send a message and then close your eyes and hope for the best.

People, when you're talking with other people, engage in a closed loop dialogue. There's feedback going on in both directions that acts to detect and reduce errors in the communication. And this is a critical thing for robots to exploit because robots have a lot more problems than people do in terms of perceiving the environment and acting in the environment. So it's really important that we establish some kind of feedback loop between the human and the robot so that the robot can infer what the person wants and carry out helpful actions. So the three parts of the talk are going to be about each of these three things.

So this is my dad's pantry in a home. And it's kind of like John's pictures of the Google car. Most robots can't pick up most objects most of the time. It's really hard to imagine a robot doing anything with a scene like this one.

There was just the Amazon picking challenge and the team that won used a vacuum cleaner, not a gripper to pick up the objects. They literally sucked the things up in the gripper and then

turned the vacuum cleaner off to put things down. And the Amazon challenge had much, much sparser stuff on the shelves. We'd really like to be able to do things like this.

And what we're doing now I'm really going to focus on a sub-problem, which is object delivery. So from my perspective, I think a really important sort of baseline capability for a manipulator robot is to be able to pick something up and move it somewhere else. We'd obviously love a lot more things. We're also talking about buttoning shirts in the car. And you can go on with all the things you might want your robot to do.

But at least, we'd like to be able to do pick and place. Pick it up and put it down. So maybe you are in a factory delivering tools, or maybe you're in the kitchen delivering stuff like ingredients or cooking utensils.

So to do pick and place in response to natural language commands-- so let's say, hand me the knife or something-- you need to know a few things about the object. First of all, you need to be able to know what it is. If they said, hand me the ruler, you need to be able to know whether or not this object is a ruler. So some kind of label that can hook up to some kind of language model.

Second, you have to know where the object is in the world because you're going to actually move your grippers and your object and yourself through 3D space in order to find that object. So here I'm going to highlight the pixels of the object. But you have to register those pixels into some kind of coordinate system that lets you move your gripper over to that object. And then third, you have to know where on that object are you going to put your gripper.

So in the case of this ruler, it's pretty heavy, and it's this funny shape that doesn't have very good friction. So for our robot, the best place to pick it up is in the middle of the object. And there might be more than one good place, and it might depend on the gripper. And different objects might have complex things going on that change where the right place is to pick it up.

So conventional approaches to this problem fall into two general categories. The first category, the first high-level approach is what I'm going to call category-based grasping. This is the dream.

So the dream is that you walk up to your robot, you hand it an object that it's never seen before, and the robot infers all three of those things, what it is, where it is, and where to put the gripper. And there's a line of work that does this. So this is one paper from Ashutosh

Saxena, and there's a bunch of others.

The problem is that it doesn't work well enough. We are not at the accuracy rates that sort of John was alluding to that we need for driving. So in Ashutosh's paper, I think they got 70% or 80% pick success rate on their particular test set at doing category-based grasping. And I think that you're going to have to expect that to fall if you actually give it a wider array of objects in the home. And even if it doesn't fall, 80% means it's dropping things 20% of the time, and that's not so good.

The second approach is instance-based grasping. So I was talking to Eric Sudderth in my department who does machine learning, he said, instant recognition is a solved problem in computer vision. So instant recognition is I give you a training set of the slide flipper, lots of images of it, and then your job given a new picture is to draw a little box around the slide flipper. This is considered a solved problem in computer vision. There is a data set and a corpus, and the performance maxed out, and people have stopped working on it.

And a lot of the work in robotics uses this kind of approach. We were talking about you have some kind of geometric model. That's the instance-based model. These models can take a lot of different forms. They can be an image or a 3D model or whatever it is.

The problem is, where do you get that model. So if I am in my house and there is thousands of different objects, you're not going to have the 3D model most likely for the object that you want to pick up right now for the person. So there's this sort of data grab. But if you do have the model, it can be really, really accurate because you can know a lot about the object that you're trying to pick up.

So the contribution of our approach is to try to bridge these two by enabling a robot to get the accuracy of the instance-based approach by autonomously collecting its own data that it needs in order to robustly manipulate objects. So we're going to get the accuracy of instance-based approach and the generality of category at the cost of not human time, but robot time to build this model. So here's what it looks like on our Baxter. It's going to make a point cloud.

This is showing-- it's got a one pixel connect in its gripper. So you're seeing it doing a sort of raster scan. This is sped up to get a point cloud. Now it's taking images of the object. So it's got an RGB camera in its wrist. It's taking pictures of the object from lots of different perspectives.

So the data looks like this. You segment out the object from the background. You get lots and lots and lots of images. You do completely standard computer vision stuff, SIFT and kNN, to make a detector out of this data. You can get a point cloud. This is the point cloud looks like at one-centimeter resolution.

And after we do this, we're able to pick up lots of stuff. So this is showing our robot-- these two objects, localizing the object and picking things up. It's going to pick up the egg. And that's a practice EpiPen. There's a little shake to make sure it's got a good grasp.

Now this works on a lot of objects, so let's see how it does on the ruler. So the way that the system is working is it's using the point cloud to infer where to grasp the object. But we don't really have a model of physics or friction or slippage.

So it infers a grasp near the end because it fits in the gripper. It kind of looks like it's going to work. And it does fit in the gripper, but when we go and do that shake, what's going to happen is it's going to pop right out of the gripper because it's got this relatively low friction. There it goes and falls out. So that's bad, right? We don't really like it when our robots drop things.

So before training, what happens is it falls out of the gripper. In the case of the ruler, there's sort of physics going on, right? Things are slipping out, and maybe we should be doing physical reasoning. I think we should be doing physical reasoning. I won't say "maybe" about that.

But we're not doing it right now. And there's lots of reasons things could fail. So in other problematic objects, this is one of those salt shakers. It's got black handles that are great for our robot to pick up, but they're black, so they absorb the IR light, so we can't see them. So we can't figure out that we're supposed to grab there.

That round bulb looks awesome. It's transparent though, so you get all these weird reflections. So the robots-- are inference algorithms is like, oh, that bulb, that's where we should pick it up. It doesn't fit in the gripper. So it will very often slip out of the gripper.

So what our approach to solve this problem is, is we're going to let the robot practice. So we have-- I'm not going to go through the algorithm, but we have this unarmed bandit algorithm that lets us systematically decide where we should pick objects up. You can give it a prior on where you think good graphs are. And you can use whatever information you want in that prior.

And if the prior was perfect, this would be boring. It would just work the first time, and life would go on. But if the prior's wrong for any reason, the robot will be able to detect it and fix things up and learn where the most reliable places are to pick up those objects. So here's an example of what happens when we use this algorithm. We practice picking up the ruler.

I forget how many times it had. Maybe 20 on this particular object. One of the rifts on the algorithm is it decides when to stop. So we go a maximum 50 picks. But we might stop after three if all three of them work so that you can go on to the next object to train.

So here it picks up in the middle and does a nice shake. OK, so what we're doing now is scaling up this whole thing. So this is showing our robot practicing on lots and lots of different objects. A lot of them are toys. My son likes to watch this video because he likes to see the robot playing with all of his toys. And I think playing is actually-- I mean it's one of those loaded cognitive science words, but I think that's an interesting way to think about what the robots are actually doing right now.

It's doing little experiments trying to pick up these objects in different places and recording where it works and where it doesn't work. So this is sort of showing 16, 32, one in each hand objects being done in our initial evaluation. And at the end of this, basically, it works. So this is all the objects in our test set. And before learning, we were able to do with this proposal system, which uses the steps information, we get about 50% pick success rate. After learning, they go up to 75%.

And the other really cool thing is that this is a bimodal distribution. So it doesn't say 75% is what you're going to get. A lot of these objects worked eight, nine out of 10 times or 10 out of 10 times. It goes from worst to best. So the good stuff is all over there, and the hard stuff is all over there.

A lot of other objects were really hard. So that garlic press I think we picked it up one time. It's really, really heavy, so it slips out a lot. That gyro-ball thing has a lot of reflection, so we had trouble localizing it accurately. So we picked it up very few times.

I think everything from about the EpiPen over was eight out of 10 or better. So not only-- so there's a lot of objects that we can pick up, and we can know which ones we can pick up. And which ones we can't. We are right now taking an aggressively instance-based approach. And the reason that we're doing that is I think there's something magic when the robot actually

picks something up.

So where I wanted to start is let's cheat in every way we can. Let's completely make a model that's totally specific to this particular object. But the next step that we're doing is to try to scale up this whole thing and then start to think about more general models to go back to that dream of category-based recognition.

So if you look at computer vision success stories, one of the things that makes a lot of algorithms successful is data sets. And the size of those data sets is immense. A lot of the computer vision data sets, COCO DB from Microsoft, have millions of images, which are labeled with where the object is. But most of those images are taken by a human photographer on your cell phone or uploaded to Flicker. Wherever they got them from.

And you get to see each object once. Maybe you see it twice from one perspective that a human carefully chose. You don't get to play with it. You don't get to manipulate it. In robotics, there's some data sets of object instances. The largest ones have a few hundred of objects.

So computer vision people that I've talked to they laugh at it because it's just so much smaller compared to the data sets that we're working with. I think it's also so much smaller than what a human child gets to play with over the course of going from zero to two years old. I guess my son became a mobile manipulator around a year later around one and a half or so. I'm not sure exactly when.

So one of my goals is to scale up this whole thing to change this data equation to be more in our favor. So there's about 300 of these-- this is the Baxter robot-- there's about 300 of them that Rethink Robotics-- Rod Brooks-- so we were talking about Rob Brooks in the previous talk. Rod founded this company Rethink Robotics. They've sold about 300 of them to the robotics research community. That's a very high penetration rate in robotics research. Everybody has a Baxter or a friend with a Baxter.

So we're starting something which we're calling the million object challenge. And the goal is to enlist all of those Baxters, which are sitting around doing nothing a lot of the time-- to change this data equation. So what we're doing is we're going to try to get everybody to scan objects for us, so that we can get models, perceptual models, visual models, and also manipulation experiences with these objects to try to train new and better category models.

And I think even existing algorithms may work way better simply because they have better

data. But I think it also opens up the door to thinking about better models that we maybe couldn't even think about before because we just didn't have the data to play with them. So where we are right now is we've installed our stack at MIT on Daniela Rus's Baxter. That's this one.

And we went down to Yale a couple of weeks ago to Scass's lab and we have our software on their Baxter. We're going to Rethink tomorrow. They're going to give us three Baxters that we're going to play with and install there.

And I have a verbal yes from WPI. And a few other people have been like-- I pitched this at RSS. So a lot of people have said they were interested. I don't know if they'll actually translate to robot time.

And our goal is to get about 500 or 1,000 objects between these three sites. Four sites I guess if the WPI gets on board. Four sites including us.

So Rethink, Yale, MIT and us. And then do like a larger press release about the project. Advertise it, push over all of our friends with Baxters to help us scan. And then have yearly scanathons where you download the latest software and then spend a couple of days scanning objects for the glory of robotics or something. And really try to change this data equation for the better, so we can manipulate lots of things.

So that's our plan for making robots that can robustly perform actions and real-world environments. More generally, I imagine like a mobile robot walking around your house at night and scanning stuff completely autonomously. Taking these pictures, building these models, hopefully, not breaking too much of your stuff. And not only learning about your particular house and the things that are in it, but also collecting data that will enable other robots to perform better over time.

All right, so that's our attack on making robots robustly perform actions in real-world environments. So the next problem that I think is important for language understanding of human robot collaboration is making robots to carry out complex sequences of actions. So for example, this is this pantry again. There might be hundreds or thousands of objects that the robot could potentially manipulate. And it might need to do a sequence of 10 or 20 manipulations in order to solve a problem such as clean up the kitchen or put away the groceries.

For work that I had done in the past on the forklift a lot of the commands that we studied and thought about were the level of abstraction of put the pallet on the truck. But one of our annotators-- we cleared the law of data on Amazon Mechanical Turk. And one of our annotators gave us this problem that I never forgot, which was how-- it was the actual forklift operator who worked in a warehouse, and he said if you paid me extra money, I'll tell you how to pick up a dime-- a dime, like a little coin with a forklift.

Here's the instructions that he eventually without making us pay him gave us for how to solve this problem. So it was raise the forks 12 inches, line it in front of the dime, tilt it forward, drive a little bit over, you lower the fork on top of the dime, put it in reverse and travel backward, the dime kind of flips up backwards on top of the fork. Maybe you know how to drive a forklift, but you can see how that would work. And if you did know how to drive a forklift, you can follow those instructions and have it happen.

But I knew that our system if we gave it these commands, there is no way that it would work. It would completely fall apart. And the reason that it would fall apart is that we gave the robot a model of actions at a different level of abstraction than this language is using. We gave it a very high-level of abstract actions, like picking stuff up and moving it into particular locations and moving things down.

And if we gave it these low-level actions of like raising the forks 12 inches, the search steps that would be required to find a high-level thing like put the pallet on the truck would be prohibitively expensive. And I think if we want to have human-- but the thing is people don't like to stick at any fixed level of abstraction. People move up and down the tree freely. They give very high-level, mid-level, low-level commands. So I think we need new planning algorithms that support this kind of thing.

So to think about this, we decided to look at a version of the problem in simulation. The simulator that we chose is a game called Minecraft. Five minutes, OK. So it's sort of this-- this is a picture from a Minecraft world. And we're trying to figure out new planning algorithms.

So the problem here is that the agent needs to cross the trench. So he needs to make a bridge to get across the trench. So it's got some blocks that he can manipulate. And you have this combinatorial explosion of where the blocks can go. They can go anywhere. So in a naive algorithm, we'll spend a lot of time putting the blocks everywhere, which doesn't really make progress towards solving a problem. Whereas what you really need to do is focus on putting

these blocks actually in the trench in order to solve the problem.

Of course, on a different day, you might be asked to make a tower or make a castle or make a staircase, and then these might be good things to do. So you don't just throw out those actions. You want to have them both and figure out what to do based on your high-level goal. So we have some work about learning how to do this. So we have an agent that practices solving small Minecraft problems and then learns how to solve bigger problems from experience.

This is showing transferring this from small problems to big problems in a decision theoretic framework, an MDP framework. And we've just released a couple of weeks ago a mod for Minecraft, the game called BurlapCraft. BURLAP is our reinforcement learning and planning framework that James MacGlashan and Michael Littman developed in Java. So you can run BURLAP inside the Minecraft JVM. Get the state of the real Minecraft world. Make small toy problems if you want. Or let your agent go in the real thing and explore the whole space of possible Minecraft spaces if you're interested in that simulation.

OK, I'm almost out of time, so I'm not going to go too much into robots coordinating with people. But maybe I will show some of the videos about this work. The idea is that a lot of the previous work and language understanding shows people works in batch mode. So the robot does something-- the person says something, the robot thinks for a long time, and then the robot does something. Hopefully, the right thing. And as I said before, this is not how people will work.

So we're working on new models that enable the robot-- this is a graphical model that shows how it-- talking about it in the car. What happens, it incrementally interprets language and gesture updating at very high frequencies. So this is showing the belief about which objects the person wants, updating from their language and gesture in an animated kind of way, right? Like, it's updating at 14 Hertz.

So the idea is that the robot has the information. This is its own language. I would like a bowl. Both bowls go up. That one he points and then the one that he's pointing at goes up.

So the robot knows very, very quickly, every time we get a new word from [INAUDIBLE] condition, every time we get a new observation from the gesture system, we update our belief. And just a couple of weeks ago, we had our first pilot results showing that we can use this information to enable the robot to produce real-time feedback that increases the human's

accuracy at getting the robot to select the right object.

This is some quantitative results. I'll skip it. OK, so that's the three main thrusts that I'm working on in my research group. Trying to make robots that can robustly perform actions in real-world environments. Thinking about planning in a really large state action spaces that result when you have a capable and powerful robot. And then thinking about how you can make the robot coordinate with people so that they can figure out what to do in these really large state actions spaces. Thank you.