JOHN LEONARD: OK, thanks. Thanks for the opportunity to talk. So hi, everyone. It's a great pleasure to talk here at MBL. I've been coming to the Woods Hole Oceanographic Institution for many years as my first thing over here at MBL. And so I'm going to try to cover three different topics, which is probably a little ambitious on time. But there's so much I'd love to say to you.

I want to talk about self-driving cars. And use it as a context to think about questions of representation for localization and mapping, and maybe connect it into some of the brain questions that you folks are interested in, and time permitting, at the end mention a little bit of work we've done on object-based mapping in my lab.

So my background-- I grew up in Philadelphia. Went to UPenn for engineering. But then went to Oxford to do my PhD at a very exciting time when their computer vision and robotics group was just being formed at Oxford under Michael Brady. And then I came back to MIT and started working with underwater vehicles. And that's when I got involved with Woods Hole Oceanographic Institution. And I was very fortunate to join the AI lab back around 2002, which became part of CSAIL. And really, I've been able to work with really amazing colleagues and amazing robots in a challenging set of environments.

So autonomous underwater vehicles provide a very unique challenge because we have very poor communications to them. Typically, we use acoustic modems that might give you 96 bytes if you're lucky every 10 seconds to a few kilometers range. And so we also need to think about the sort of constraints of running in real time onboard a vehicle.

And so the sort of work that my lab's done-- the more we investigate more fundamental questions about robot perception, navigation, and mapping, we also are involved in building systems. So this is a project I did for the Office of Naval Research some years ago using small vehicles that would reacquire mine-like targets on the bottom for the Navy. And so this is an example of a more applied system where we had a very small resource-constrained platform. And the sort of work we did is a robot built a map as it performed its mission, and then

matched the map against the prior map to do terminal guidance to a target.

Another big system I was involved with, as Russ mentioned, was the Urban Challenge. And I'll say a bit about that in the context of self-driving cars. So let's see. So who's heard any of the recent statements from Elon Musk from Tesla?

So he said self-driving cars are solved- he said. And a particular thing that he said that just made my-- I don't know, maybe steam came out of my head-- was that he compared autonomous cars with elevators that used to require operators but are now self-service. So imagine you getting in a car, pressing a button, and arriving at MIT in Cambridge 80 miles away, navigating through the Boston downtown highways and intersections.

And maybe that will happen. But I think it's going to take a lot longer than folks are saying. And some of that comes from just fundamental questions and intelligence and robotics. So in a nutshell, when Musk says that self-driving is solved I think he's wrong, as much as I admire what Tesla and SpaceX have done.

And so to talk about that, I think we need to be very honest as a field about our failures as well as our successes, and try to balance what you hear in the media with the reality of where I think we are. And so I wanted to quote verbatim what Russ said about the robotics challenge, about a project that was so exhausting and just all-consuming and so stressful, yet so rewarding.

So we did this in 2006 and 2007-- my wonderful colleagues, Seth Teller, John Howe, Amelia Fratoli-- amazing students and postdocs. We had a very large team. And we tried to push the limit on what was possible with perception and real-time motion planning.

So our vehicle built a local map as it traveled from its perceptual data, using data from laser scanners and cameras. And we didn't want to blindly follow GPS. We wanted the car to make its own decisions because GPS navigation was part of the original quest with the challenge.

And so Seth Teller and his student, Albert Wang, developed a vision-based perceptual system where the car tried to detect from curbs and lane markings in very challenging vision conditions. For example, looking into the sun, which you'll see in a second-- really challenging situation for trying to perceive the world.

And so our vehicle-- at the time, we went a little crazy on the computation. We had 10 blades, each with four cores-- 40 cores-- which may not seem a lot now, but we needed 3.5 kilowatts

just to power the computer at full tilt. We fully loaded the computer with a randomized motion planner, with all these perception algorithms. We had a Velodyne laser scanner on the roof. And about 12 other laser scanners, 5 cameras, 15 radars, and we really pushed the envelope on algorithms.

And so when faced with a choice in a DARPA challenge, if you want to win at all costs you might simplify, or try to read the rules carefully, or guess the rule simplifications. But that would have meant just sort of turning off the work of our PhD students, and we didn't want to do that.

So at the end of the day, all credit to the teams that did well. Carnegie Mellon-- first, $2 million, Stanford-- second, $1 million, Virginia Tech-- third, half a million dollars, MIT-- fourth, and nothing for fourth place. But it was quite an amazing experience. And in the spirit of advertising our failures I think I have time to show this. This used to be painful for me to watch. But now I've gotten over it. This is our--

[VIDEO PLAYBACK]

- Let's check in once again with the boss.

JOHN LEONARD: Even though we finished the race, we had a few incidents so DARPA stopped things and let us continue.

- --across the line.

JOHN LEONARD: Carnegie-Mellon, who won the race. Why did that stop? Let's see.

- --at the end of mission two behind Virginia Tech. Virginia Tech got a little issue. [INAUDIBLE] Here's--

JOHN LEONARD: We were trying to pass Cornell for a few minutes.

- Looks like they're stopped. And it looks like they're-- that the 79 is trying to pass and has passed the chase vehicle for Skynet, the 26 vehicle. Wow. And now he's done it. And Talos is going to pass. Very aggressive. And, whoa. Ohh. We had our first collision. Crash in turn one. Oh boy. That is, you know, that's a bold maneuver.

[END PLAYBACK]

JOHN LEONARD: So what actually happened? So it turned out Cornell were having problems with their

actuators. They were sort of stopping and starting and stopping and starting. And we had some problems. It turned out we had about five bugs. They had about five bugs that interacted. And here's a computer's eye-- sort of, brain of the robot's view.

Now back in '07, we weren't using a lot of vision for object detection and classification. So with the laser scanner-- the Cornell vehicle's there. It has a license plate. It has tail lights. It has a big number 26. It's on the middle of a road. We should know that's a car. Stay away from it. But to the laser scanner it's just a blob of laser scanner data. And even when we pull around the side of the car we weren't clever enough with our algorithms to fill in the fact that it's a car.

And you have the problem when it starts moving of the aperture problem-- that as you're moving, and it's moving, it's very hard to tell and deduce the true motion. Now, another thing that happened was we had a threshold. And so in our 150,000 lines of code our wonderfully gifted student, who's now a tenured professor at Michigan, Ed Olson, had a threshold of 3 meters per second. So anything moving faster than 3 meters per second could be a car. Anything less than 3 meters per second couldn't be a car.

Now that might seem kind of silly. But it turns out that slowly moving obstacles are much harder to detect and classify than fast moving obstacles. That's one reason that city driving or driving, say, in a shopping mall parking lot is actually in many ways more challenging than driving on the highway. And so despite our best efforts to stop at the last minute, we steered into the car and had this little minor fender bender.

But one thing that we did is we made all our data available open source. And we actually wrote a journal article on this incident and a few others. And so if you'd asked me then in 2007, I would have said we're a long way from turning your car loose on the streets of Boston with absolutely no user input.

And the real challenge is our uncertainty and robustness and developing robust systems that really work. But for our system, some of the algorithm progress we made-- I mentioned the lane tracking. Albert Wang, who's now, I think, working at Google, developed-- was given very sparse-- I'd say about 10% of the recent graduates or more are working at Google these days.

AUDIENCE: Albert's at [INAUDIBLE].

JOHN LEONARD: Oh. OK. And then here is a video for the qualifying event to get into the final race. We had to navigate-- whoops, I can't press the mouse. That's going to stop. So we had to navigate along

a curved road with very sparse waypoints.

And so, in real time the computer has to make decisions about what it sees. Where is the road? Where am I? Are there obstacles? And there are no parked cars in this situation, but other stretches had parked cars.

And our car-- in a nutshell, if our robot became confused about where the road was it would stop. It would have to wait and get its courage up, like lowering its thresholds as it was stuck. But we were the only team to our knowledge to qualify without actually adding waypoints. So it turns out the other top teams, they just went in with a Google satellite image and just added a breadcrumb trail for the robot to follow, simplifying the perception.

So this was back in '07. Now let's fast forward to 2015. And right now-- so of course, we have the Google self-driving car which has just been an amazing project. And you've all probably seen these videos, each with millions of hits on YouTube. The earlier one of taking a blind person for a ride to Taco Bell, this was driving-- that was 2012, city streets in 2014, spring 2015. And then the new Google car, which won't have a steering wheel in its final instantiation, won't have pedals. It will just have a stop button. And that's your analogy to the elevator.

And so I think that the Google car is an amazing research project that might one day transform mobility. But I do think, with all sincerity-- so I rode in the Google car last summer. I was blown away. I felt like I was on the beach at Kitty Hawk. It's like this just really profound technology that could in the long term have a very big impact. And I have amazing respect for that team-- Chris Urmson, Mike Montemerlo, et cetera.

But I think in the media and in others, the technology has been a bit overhyped, and it's poorly misunderstood. And a lot of it goes down to how the car localizes itself, how it uses prior maps, and how they simplify the task of driving. And so even though people like Musk have said driving is a solved problem, I think we have to be aware that just because it works for Google, doesn't mean it'll work for everybody else.

So critical differences between Google and, say, everyone else. And this is with all respect to all players. I'm not trying to criticize. It's more just trying to balance the debate. The Google car localizes on the left with a prior map, where they map the lighter intensity off of the ground surface. And they will annotate the map by hand-- adding pedestrian crossings, adding stoplights. They'll drive a car around many, many times, and then do a SLAM process to optimize the map.

But if the world changes, they're going to have to adapt to that. Now, they've shown the ability to do response to construction, bicyclists with hand signals. When I was in the car we crossed the railroad tracks. That just blew me away. I mean, it's pretty impressive capability but more a vision-based approach that just follows the lane markings. If the lane markings are good, everything's fine.

In fact, Tesla either just have released-- or are about to release-- their autopilot software, which is an advanced lane keeping system. And Elon Musk, a few weeks ago, posted on Twitter that there's one last corner case for us to fix.

And apparently he-- on part of his commute in the Los Angeles area there is well defined lane markings. And part of it is a concrete road with weeds and skid marks and so forth. And he said publicly that the system works well if the lane markings are well-defined. But for more challenging vision conditions like looking into the sun it doesn't work as well.

And so the critical difference is if you're going to use the LiDAR with prior maps, you can do very precise localization down to less than 10 centimeters accuracy. And the way I think about it is robot navigation is about three things-- where do you want the robot to be? Where does the robot think it is? And where really is the robot? And when the robot thinks it's somewhere, but it's really somewhere different, that's really bad. That happens.

We've lost underwater vehicles and had very nervous searches to find them-- luckily-- when the robot made a mistake. And so with the Google approach they really nail this "where am I" problem-- the localization problem. But it means having an expensive LiDar. It means having accurate maps. It means maintaining them.

One critical distinction is between level four and level three. These are definitions of autonomy from the US government-- from NTSA. A level four car is what Google are trying to do now, which is really, you just-- you could go to sleep. The car has a 100% control. You couldn't intervene if you wanted to. You just press a button. Go to sleep. Wake up at your destination.

Musk has said that he thinks within five years you can go to sleep in your car, which to me I just-- five decades would impress me, to be honest. But level three is when the car is going to do most of the job, but you have to take over if something goes wrong.

And for example Delphi drove 99% of the way across the US in spring of this year, which is

pretty impressive. But 50 miles had to be driven by people-- getting on and off of highways and city streets. And so there's something about human nature, and the way humans interact with autonomous systems, that it's actually kind of hard for a person to pay attention.

Imagine if 99% of the time the car does it perfectly. But 1% of the time it's about to make a mistake, and you have to be alert to take over. And research experience from aviation has shown that humans are actually bad at that.

And another issue is-- and this is-- I mean, Mountainview is pretty complicated-- lots of cyclists, pedestrians, I mentioned the railroad crossings, construction. But in California they've had this historic drought. And most of the testing has been done with no rain, for example, and no snow. And if you think about Boston and Boston roads, there are some pretty challenging situations.

And so for myself, when I first-- a couple of years ago I said I didn't expect a taxi in Manhattan in my lifetime-- a fully autonomous taxi-- to go anywhere in Manhattan. And I got criticized online for saying that. So I put a dash cam on my car, and actually had my son record cell phone footage.

The upper left is making a left turn near my house in Newton, Mass. And if you look to the right, there's cars as far as the eye can see. And if you look to the left, there's cars coming at pretty high rate of speed, with a mailbox, and a tree.

And this is a really challenging behavior for a human, because it requires making a decision in real time. We want very high reliability in terms of detecting the cars coming from the left. But the way that I pulled out is to wave at a person in another car. And those sort of nods and waves-- they're some of the most challenging forms of human-computer interaction. So imagine vision algorithms that could detect a person nodding at you from the other direction.

Or here's another situation. This is going through Coolidge Corner in Brookline. And I'll show a longer version of this in a second. But the light's green. And see here-- this police officer? So despite the green light, the police officer just raises their hand, and that means the signal to stop. And so interacting with crossing guards and people-- very challenging, as well as changes to the road surface and, of course, adverse weather. And so here's a longer sequence for that police officer.

First of all, you'll see flashing lights on the left-- which may be flashing lights, you should pull

over. Here you should just drive past them. It's just the cop left his lights on when he parked his car. But the light's red. And this police officer is waving me through a red light, which I think is a really advanced behavior. So imagine a car that's-- imagine the logic for OK, stop at red lights unless there's a police officer waving you through it, and how you get that reliable.

And now we're going to pull up to the next intersection, and this police officer is going to stop at a green light. And so despite all the recent progress in vision, things like image labeling, ImageNet-- most of those systems are trained with vast archives of images from the internet where there's no context. And they're so challenging for even humans to classify. So that if you had some data sets, like the Caltech pedestrian data set, if you got 78% performance, that's really good. But we need 99.9999% or better performance before we're going to turn cars loose in the wild in these challenging situations.

Now going back more to localization and mapping. Here I collected data for about three or four weeks of my commuting. This is crossing the Mass. Ave. Bridge going from Boston into Cambridge. And the lighting is a little tricky. But tell me what's different between the top and the bottom video.

And notice, by the way, how close we come to this truck. The slightest angular error in your position estimate, really bad things could happen. But the top-- this is a long weekend. This is Veterans Day weekend. They repaved the Mass. Ave. Bridge. So on the bottom, the lane lines are gone. And so if you had an appearance-based localization algorithm like Google's, you would need to remap the bridge before you drove on it. But the lines aren't there yet. And how well is it going to work? And so, this is just a really tricky situation.

And, of course, there's weather. Now, snow is difficult for things like traction and control. But for perception, if you look at how the Google car actually works-- if you're going to localize yourself based on precisely knowing the car's position down to centimeters so that you can predict what you should see, then if you can't see the road surface you're not going to be able to localize. And so this is just a reminder of the sorts of maps that Google uses. So I think to make it to really challenging weather and very complex environments, we need a higher level understanding of the world. I think more a semantic or object-based understanding of the world.

And then, of course, there's difficulties in perception. And so what do you see in this picture? The sun? There's a green light there. I realize the lighting is really harsh, and maybe you

could do polarization or something better.

But does anyone see the traffic cop standing there? You can just make out his legs. There's a policeman there who gave me this little wave, even though I was sort of blinded by the sun. And he walked out and put his back to me and was waving pedestrians across, even though the light was green. So a purely vision-based system is going to just need dramatic leaps in visual performance.

So to wrap up the self-driving car part, I think the big questions going forward-- technical challenges, maintaining the maps, dealing with adverse weather, interacting with people-- both inside and outside of the car-- and then getting truly robust computer vision algorithms. We want to get in a totally different place on the ROC curves, or the precision recall curves, where approaching perfect detection with no false alarms. And that's a really hard thing to do.

So I've worked my whole life on the robot mapping and localization problem. And for this audience I wanted to just ask you a little question. Does anyone know what the 2014 Nobel Prize in medicine or physiology was for? Anybody?

AUDIENCE: [INAUDIBLE]

AUDIENCE: Grid cells.

JOHN LEONARD: Grid cells. Grid cells and place cells. And so this has been called SLAM in the brain. Now, you might argue. And we might be very far from knowing. But I think it's just really exciting to-- so for myself, I'll explain.

I've had what's called an ONR MURI grant-- multidisciplinary university research initiative grant-- with Mike Hasselmo and his colleagues at Boston University. And these are a couple of Mike's videos. And so, I think Matt Wilson spoke to your group. And the notion that in the entorhinal cortex that there is this sort of position information that's very metrical, and it seems to be at the heart of memory formation, to me is very powerful and very important.

And so, we have this underlying question of representation. How do we represent the world? And I believe location is just absolutely vital to building memories and to developing advanced reasoning in the world. And the fact that grid cells exist-- to me-- and they have this role in memory formation is just this really exciting concept.

And so, in robotics we call the problem of how a robot builds a map and uses that map to

navigate, SLAM-- simultaneous localization and mapping. This is for a PR2 robot being driven around the second floor of our building, not far from Patrick's office if you recognize any of that. And this is using stereo vision.

My PhD student, Hordur Johannsson, who graduated a couple of years ago, created a system to do real time SLAM and try to address how to get temporally scalable representations. And one thing you'll see as the robot goes around occasionally is loop closing, where the robot might come back and have like, an error and then correct that error.

So this is the part of the SLAM problem that in some ways is well understood in robotics, which is how you detect features from images, track them over time, and try to bootstrap up, building a representation and using that to locate your estimation.

And I've worked on this my whole career. And as a grad student at Oxford, I had very primitive sensors. So for a historical SLAM talk I recently digitized an old video and some old pictures. This was in the basement of the engineering building at Oxford. This is just the localization part of how you have a map, and you generate predictions-- in this case for sonar measurements.

And at the time there we had-- I'm sitting at a SUN workstation. To my left is something called a data cube, which for about $100,000 could just barely do like real time frame grabbing and then edge detection out. And so vision just wasn't ready.

And the exciting thing now in our field is vision is ready-- that we're really using vision in a substantial way. But I think a lot about prediction. If you know your position, you can predict what you should see and create a feedback loop. And that's sort of what we're trying to do.

And so SLAM is a wonderful problem, I believe, for addressing a whole great set of questions, because there are these different axes of difficulty that interact with one another. And one is representation. How do we represent the world? And I think that question-- we still have a ton of things to think about.

Another is inference. We want to do real time inference about what's where in the world and how we combine it all together. And finally, there's a systems in autonomy access, where we want to build systems, and deploy them, and have them operate robustly and reliably in the world.

So in SLAM, here's an example of how we pose this as an inference problem. This is from the classic Victoria Park data set from Sydney University. A robot drives around, in this case, a park with some trees. There are landmarks shown in green. The robot's positioner drifts over time. We have dead reckoning error. That's shown in blue. And we estimate the trajectory of the robot in red, and the position of the landmarks from relative measurement.

So as you take relative measurements, and you move through the world, how do you put that all together? And so we, cast this as an inference problem where we have the robot poses, the odometric inputs, landmarks-- you can do it with or without landmarks-- and measurements.

And an interesting thing-- so we have this inference problem on a belief network. The key thing about SLAM is it's building up over time. So you start with nothing and the problem's growing ever larger.

And, let's see, if I had to say-- 25 years of thinking about this up through 2012, the most important thing I learned is that maintaining sparsity in the underlying representation is critical. And, in fact, for biological systems I wonder if there is evidence of sparsity. Because sparsity is the key to doing efficient inference when you pose this problem. And so many algorithms have basically boiled down to maintaining sparsity and the underlying representations.

So just briefly, the most important thing I learned since then in the last few years-- I'm really excited by building dense representations. So this is work in collaboration with some folks in Ireland-- Tom Whelan, John McDonald-- building on KinectFusion from Richard Newcombe and Andrew Davison-- how you can use a GPU to build a volumetric representation, and build rich, dense models, and estimate your motion as you go through the world. So this is something we call continuous or spatially extended KinectFusion.

This little video here from three years ago is going on in an apartment in Ireland. And I'll show you the end result. Just hand-carrying a sensor through the world-- and you can see the quality of the reconstructions you can build, say, in the bathroom, the sink, the tub, the stairs, to have really rich 3D models that we can build and then enable the more advanced interactions that Russ showed. That's fantastic.

And I mentioned loop closing-- something we did a couple of years ago was adding loop closing to these dense representations. So this is-- again, in CSAIL-- this is walking around the Stata Center with about eight minutes of data going up and down stairs. If you watch the two blue chairs near Randy Davis's office, you can see how they get locked into place as you

correct the error.

So this is taking mesh deformation techniques from graphics and combining it. So the underlying pose graph representation is like a foundation or a skeleton on which you build the rich representation. OK. So this is the end resulting map. And there's been some really exciting work just this year from Whelan and from Newcombe in this space of doing deformable objects, and then really scalable algorithms where you can sort of paint the world.

So the final thing I want to talk about in my last few minutes is our latest work of using object-based representations. And for this audience, I think if you go back to David Marr, who I feel is unappreciated in the historical sense of how I feel, that vision is the process of discovering from images what is present in the world and where it is. And to me, the what and where are coupled. And maybe that's been lost a bit. And I think that's one way in which robotics can help, I think, with vision and brain sciences.

I think we need to develop object-based understanding of the world. So instead of just having representations that are a massive amount of points or purely appearance, where we can start to build higher level and symbolic understanding of the world. And so I want to build rich representations that leverage knowledge of your location to better understand where objects are and knowledge about objects to better understand your location.

And just as a step in that direction, my student, Sudeep Pallai, who was one of Seth's students, has an RSS paper where we looked at coupling using SLAM to get better object recognition by effectively-- so here's an example of an input data stream from Peter Fox's group. There's just some objects on the table. I realize it's a relatively uncluttered scene. But this has been a benchmark for RGBD perception.

And so, if you combine data as you move from the world using a SLAM system to do 3D reconstruction on the scene, and then using the reconstructed points to help improve the prediction process for object recognition, it leads to a more scalable system for recognizing objects. And it comes back to this notion to me that a big part of perception is prediction-- the ability to predict what you see from a given location. And so what we're doing is we're leveraging off techniques and object detection, featuring coding and the newer SLAM algorithms, and particularly the semi-dense orb SLAM technique from Zaragoza, Spain.

And so I'm just going to jump to the end here. The key concept is that by combining SLAM with object detection we get much better performance and object recognition. So on the left shows

our system. On the right is a classical approach just looking at individual frames. And you can see, for example, here, the red cup that's been misclassified would get substantially better performance by using location to cue the object detection techniques.

All right. So I'm going to wrap up. And just a little bit of biological inspiration from our BU collaborators, Eichenbaum has looked at the what and the where pathways in the entorhinal cortex. And there's this duality between location-based and object-based representations in the brain. And I think that's very important.

OK. So my dream is persistent autonomy and lifelong map learning and making things robust. And just for this group I made a-- I just want to pose some questions on the biological side, and I'll stop here. So some questions-- do biological representations support multiple location hypotheses? Even though we think we know where we are, robots are faced with multimodal situations all the time. And I wonder if there is any evidence for multiple hypotheses in the underlying representations in the brain, even if they don't rise to the conscious level, and how experiences build over time.

And the question-- what are the grid cells really doing? Are they a form of path integration? Or there obviously, to me, seems to be some correction. And my crazy hypothesis as a non-brain brain scientist is, do grid cells serve as an indexing mechanism that effectively facilitates search-- so a location index search so that you can have these pointers to what and where information get coupled together.