

Menlo Park CA
1:31:08
m4v

Bob Bolles

An interview conducted by
Peter Asaro
with
Selma Šabanović

Novemeber 18 2010

Q1: Why don't we start? You can introduce yourself, tell us where you were born and where you grew up.

Bob Bolles: Okay, I'm Bob Bolles. I was born in Baltimore, Maryland in 1945. I moved to Florida, Gainesville, Florida, when I was two and a half, so I really grew up in Gainesville, Florida. My father had been a professional flutist. He'd gone to Juilliard and played professionally in New York City for ten years, and then decided when he was going to have family, that wasn't going to work, that playing at night meant there was going to be no family life. So he went back to school and got a doctorate in education and then became a professor of music at the University of Florida.

Q1: Where did you do your undergraduate studies?

Bob Bolles: So undergraduate studies at Yale. Let's see, okay, yeah.

Q1: What did you study there?

Bob Bolles: So I studied in mathematics. At that time – let me go back a little bit. In high school, there were two things that were real important about high school for me. One was, sort of surprising, the University of Florida had a rule that said only one of a couple could work at the university as a professor. So there were a lot of teachers in my high school whose spouses were at the university. As a result, I think we had just superb teachers in high school. The second thing was, by the time my senior year, I'd already taken all the science classes in high school, and so they set up a program where we could go be an intern over at the University of Florida. I was an intern for a professor in mathematics, Ralph Selfridge, and I showed up the first day and he said, "I'd like you to write a computer program to try out this idea about how to solve high water polynomial equations, find the roots of higher equations." And he said, "There's a programming course starting next week." So I took a course, in 1966, to learn to program Fortran II and that sort of launched me off in computers and things like that. When I got to Yale, a couple of things happened. One is, they didn't have a computer science major, so I majored in mathematics. I also got a job, sort of through the university, to work ten hours a week to earn a little money, supplement things, where I wrote programs for the educational testing company – not company, what is it? Educational testing institute at Yale. So I wrote programs there and then I took a number of computer science classes, a couple from Professor Rosen who was quite an amazing person in testing, in stretching us.

My recollection was, every time he'd hand us a homework assignment, I'd look at it and say, "I don't have a clue how to do this." Then after working, we were allowed to work together, we'd work, talk, talk to him. We'd figure out how to do it. We'd get it written, get in gone and

we'd say, "Anything he gives us now, we've got it knocked. Whatever he gives us, we know how to do it." Sure enough, next one he'd give us one, we'd say, "I don't have any idea." So that was really nice. In high school, I'd had a couple of teachers that were like that, that would generate bonus questions. I don't think they were even for credit, but they were just extra hard problems that they would give us and I would spend most of my time doing those. I did all right in the other things too, but those were things that were really motivational.

Q1: How did you first become interested in robotics?

Bob Bolles: So let's see, it's a long story here. So I went from Yale to the University of Pennsylvania and there again, they didn't directly have computer science. They had – let's see – it was a master's in electrical engineering and I ended up having to leave – I was in a PhD program. Because it was in Vietnam time, I was going to have to join the military. My little draft board in Gainesville, Florida, was going to take me. I was physically fit and that was all they required. So I ended up leaving, after finishing a master's degree there, and going into the navy. In the navy, somebody had figured out that computer science was a big upcoming thing and they pulled six of us – most of us had master's degrees, one or two had PhDs – and sent us to the naval postgraduate school in Monterrey. So I was there for almost four years, teaching computer science to virtually everybody that was getting a master's degree there. The navy had this foresight that they wanted everybody to learn about this new thing.

One of the people that was there was Garry Kildall, who later became famous for writing CPM, which was the first operating system for microprocessors. There's a famous story about him not accepting a bid and Bill Gates winning who was going to write the operating system, and stuff like that. He was a great guy and I learned a lot from him. I would sit in on his compiler classes, operating system classes and so forth. When I finished my navy duty, I went to Stanford to finish, or actually, to earn a PhD at that point. I got assigned to Professor Feldman, who was doing language work, which I had been interested in, but he was also doing languages for robots and so that's really how I got into robots, was that Professor Feldman, Jerry Feldman, was developing description languages for robots. Then one of the people working with him was a research associate, Lou Paul, and he had finished there, I don't know, probably a year or two before I got there, but he was like a post doc there. He had the knack to pick a problem that was hard enough to really stretch you, but not impossible. He was real good at picking things that were sort of fun and interesting. So almost right when I got there, he had chosen a problem to automatically assemble a Ford Model T water pump. He thought that would be symbolic, to have a Model T, that we could have the robot arm assemble.

So I worked on the perception part of that, because he did the manipulation, control and things like that. We used an arm then called the Stanford arm. We had a gold one and a blue one that Vic Scheinman had developed. We were using the gold one and we developed a program to pick up the parts and assemble them, screw them together and things like that. We

went on to do several things from there. The next thing we did is we went down to visit McCullough Chainsaw Engines down in LA. They gave us one and we then worked on assembling some parts of that. So Lou Paul was an excellent colleague the long whole way. Actually later, it was a big step for me in my career when he and Mike Brady organized the first international symposium on robotics research, and he invited me to come and talk about some of the work I had done. At that, I then met and talked with a lot of people. These were great meetings, because they were small enough, probably 50 people, so they were small enough, you got to listen to every single talk. You would talk to them and there was a lot of encouragement to meet and do outside activities. Every afternoon was free to go fly kites, take hikes with your other colleagues, so you knew them completely differently than if you just heard their technical talk.

Q1: Do you remember when and where that first symposium was?

Bob Bolles: That was at Bretton Woods and I think like 1981. It's going on 30 years. I've gone to those meetings virtually every time and at the third one, I realized that they were sort of rotating round and the next one would be in the United States. So I talked to Professor Bernie Roth at Stanford and said, "Hey, we should host this, you know. We've got robotic stuff here." He agreed, so we hosted the fourth one here, which was UC Santa Cruz, which was wonderful. After that, I got on the board of the group and then continued to go to the meetings and meet people and things. So that really opened up my scientific career from being more local here in the US, to more all the way around the world.

Q1: Prior to that, what kind of conferences would roboticists go to, or where would you see robotic work being done?

Bob Bolles: So conferences in general, over my lifespan, have gone from the ACM, which was everything – as a student, I went and that was the only thing that had anything to do with computer science – and then it sort of specialized to IJCAI, which is the International Joint Conference on Artificial Intelligence, and it had a part of robotics. And then that split into much more specific robotics conferences and computer vision conferences like CVPR and other, IEEE, PAMI related conferences and things. So I sort of look back, that I started with ACM and I then progressively got more and more specialized in computer vision and robotics.

Q2: The meetings, you mentioned that allowed you to expand internationally. Where did the various participants come from?

Bob Bolles: So for this particular meeting, ISRR, there are three parts of the world, United States, Europe and then Asia, including Australia and things. So Peter Cork, I met at this meeting. Ray Jarvis is also Australian. Let's see. And then a number of people in Japan I've gotten to know well, like Professor Takeo Kanade who now is at CMU. He and I happened to be working on some similar things, so we talked a lot. He invited me later to be one of the editors for the National Journal of Computer Vision, which I did for three years. I was never very good at it. Technically, I was fine, but journal editing was hard for me, because I had other pressing things to do, and if I did it tomorrow, it was just the same. But after a few weeks of pushing it off, I'd end up with a pile of papers and I'd panic and I never was very good at that. So yeah, I've met people from France and Germany that we still coordinate. I haven't actually done – I've done more discussions and things. We've done less direct, say, joint projects. I have done some joint projects with people from other countries, but not directly from that. At a CVPR meeting recently, we met. We put in a proposal for an IARPA contract recently with a group from the University of Amsterdam, and won, so there we're working with that. We're working with a group from University of Leeds on a project. Actually it's called Mind's Eye, because they've done representations for regions in time and other types of elements that we need for this reasoning, to couple it with the perception. Because perception, we're more focused with kind of starting with pixels and working our way up, but we clearly need to be able to understand occlusion and the permanence of objects and things that they represent, that we typically haven't.

Q1: What kind of vision system were you working on, on the first projects?

Bob Bolles: So the very first project, like this Model T assembly work, we had a color camera. We would locate mostly black and white things, because we simplified the world, to detect objects of certain known – previously modeled objects, in pretty much like an industrial application. At that time, there were no industrial robots. When I came to SRI, Charlie Rosen was head of the AI Center and a fellow working with him, David Nitzen, the two of them had a large NSF project called the Industrial Affiliates Program. The whole point was to explore and distribute ideas about how to use robotics and computer vision in US industry. So the affiliates program had companies like General Motors and Ford and Deck and Scott Paper and all sorts of companies. I think at the height, it probably had 30 of these companies, maybe more. We'd get together once every quarter. Every other meeting was here at SRI and the other one, so two times a year, we'd go to one of these companies and see their industrial line, see what they need, what were their problems.

It was real interesting, we would hear from the managers. They'd often say, "Our problem is this." We'd go see the line and talk to the managers. They'd say, "Our problem is that." Then we'd say, "Can you send us some copies of the good ones and the bad ones?" or whatever. They would do that and then we'd work on them for a little bit and send back ideas for them. And it also meant that it gave us ideas what they really needed. During that time, three or four of us developed a program here to recognize parts that were mostly flat. They were

viewed from above so they were two dimensional parts being viewed in two dimensional images, but still they were complicated enough. They had lots of holes and indents and things, and sometimes they needed to be inspected to make sure they had all the flanges and things on them. So we wrote a thing called the local feature focus method that would do that. One aspect of that, that was sort of fun, that I liked to continue to do, but we really haven't done as much, is you could show it apart and it would automatically pick out the distinguishing features for that part, relative to maybe upside down or some other part or a set of ten parts. So it picked out the distinctive features automatically, and then use those to distinguish them. And if it couldn't, it would say, "I can't tell part A from part B." Maybe the color was the only thing that was different. It couldn't tell that very reliably.

Q1: How many pixels were you basing the decision on?

Bob Bolles: Ah! So originally, we were 128 by 128 and the cameras cost \$10,000. Some place around here, we've got collections of cameras. The original cameras are these great big things. Well, actually, the one at Stanford is actually a box this big. Now of course, you can have them in your sunglasses in the middle of your nose.

Q1: It was just black and white?

Bob Bolles: It was black and white originally. We graduated to color, but quite a bit later. It's astounding. That dimension, now, as you know, HD is plausible. You can have really nice resolution. As a computer vision person, I always want more resolution and better dynamic range, so it handles highlights and dark things a lot better.

Q1: Contrasts. What was your actual dissertation thesis?

Bob Bolles: So it was on verification vision, and the idea was that if I had a model ahead of time, could I use that to predict what I'm going to see, so that when I go looking for it, I'm constrained enough that I can reliably detect it and use that model to verify that I actually have the right thing. Now at the time, we did it just, we'd move, stop and look at it, take a while to think about it and say yes or no. Then if it's right, I measure its position and then maybe move. Now of course, the computational power and things are such that we can do this in real time. In fact, the ARM project which we just started about two months ago is to do exactly that. Is to marry real time perception with real time control. So far, a lot of robotics, there's a lot of nice real time perception and real time arm, but not much where they really marry them together. So

this project is to trying to do that, something I would loved to have been able to do 30 years ago, but now we're able to do it.

Q1: What's the project?

Bob Bolles: This project is a DARPA program, and it's called ARM. There's ARM-S for software and ARM-H for hardware. SRI won a software contract and a hardware contract. Hardware people are developing a new hand and the idea is, you want to have a new hand that's capable but relatively inexpensive, and they have some interesting ways of using flat tendons and things for grabbing and holding and measuring. They also have some electrostatic techniques, where you can pick up a cup by just touching it to the side and turning on, and picking up the cup that way, so you don't have to be able to surround it like most grasps do. On our ARM project, we're working with Columbia University, which has done a lot of work on grasping, and Carnegie Mellon, so Matt Mason.

Q1: Who are you working with at Carnegie Mellon?

Bob Bolles: So Matt Mason and Sid Srinavasa who's at Intel, but he's on the CMU campus in an Intel building, so they work really closely and have a lot of CMU graduate students working in the lab and things. So we work with both of them.

Q1: Matt Mason's at Columbia?

Bob Bolles: No, Matt Mason's at CMU. Peter Allen is the main contact at Columbia.

Q1: When did you come to SRI?

Bob Bolles: So I finished in '76, so I came here in '76. Actually, I didn't get my degree, it wasn't conferred until '77. My wife and I, I'm from Florida and she's from Atlanta, we both sort of viewed this as a four, five year post doc, maybe. But then I liked the group so much and we were doing a lot of interesting things, that I couldn't see – at that time, there weren't equivalent things in the south. Now there are things at Georgia Tech, things in North Carolina, even Florida and other places, but at that time there weren't. I wasn't really ready to start something new. Now we certainly could, but at that time, I wanted to join a group and add to it. So I ended up staying here; I'm still here.

Q2: What kind of things did you start working on when you came to SRI?

Bob Bolles: So I worked on two different things. One was this, industrial applications. The program that I mentioned, this local feature focus method for finding things that are essentially flat, was picked up by Adept Corporation here. That's because I had known Brian Carlisle and Bruce Shimano at Stanford and they wanted to provide a vision system for their robot, for the industrial case. So they re-implemented, but they used a lot of the basic ideas. That was kind of interesting to see and fun to see people starting to really use it. The other side I was more aerial video analysis, where we started out using stereo and other things to build 3D maps. We then transitioned to trying to recognize the objects like the roads in the buildings and build 3D models of them, which still is an important problem. I still do that here now. So I have the ARM project that we're working on, Mind's Eye, which is more ground level. There's no robot involved. It's just video watching something and trying to learn and understand and <inaudible>. We do aerial video work as well.

About two or three years ago, we'd developed some technology for watching a scene from the air. So when you're in the air, of course, the camera's always moving. So things like parallax on tall buildings and trees look like things are changing, but they're not movers. So we were detecting – we had a technique for handling that. The idea is, for the military, they'd like to know, can you detect some humans or vehicles moving? They always want to do it with the widest field of view, which means you've got the fewest pixels. So you've got to be able to do it with four or five pixels on a car, or four or five pixels on a person. We had developed a basic technique and refined it. Then at the last phase of that, we worked as a subcontractor to Raytheon, where we actually ported it to the hardware on the Predator. So it runs now on the UAV and it's being tested. It's kind of in hiatus. It's not directly been deployed. I think they eventually want to get it, but they've had other priorities. Hopefully, that's going to come back around. The idea there is that one application would be, say a squad picks up a cell phone off some ridge over there. They'd actually visually see them. The humans can do it better than we can, but even on a 640 by 480, they still focus at the middle. They don't notice things around the edge and they get tired after ten minutes. And we can do it all night. So the computers, they're still better at it than we are in the short term, for a few minutes, but we can cover massive areas and do it. We can say, "What about this?" and they can look at it and say, "No, that's just something, you know. Ah, okay, let's watch that." So it's a cuing mechanism.

Q1: What are some of the biggest technical challenges you've had in that kind of system?

Bob Bolles: In that particular system, the biggest challenge was, they had a particular computing box that had a low speed power PC in it. We had to port our stuff to run on that particular board. We couldn't add much to it because there were heating constraints too. Even though there was room for more things, they couldn't really handle it. So real world problems pop up when you're porting it to an existing device. Let's see, what else? One of the advantages of that, we'd done a number of projects where you analyze the data that had been sent down by telemetry. It's often compressed in some way, so the data we get down on the ground is never as

good. There are break ups and other things. So we've always wanted to work, actually up on board. So there was one of the opportunities we could do that. We've ported the same sort of software to run on the ground for the little teeny things like Raven B, where the hand launched UAVs. Sarnoff which is a company that SRI owns, and is going to be merged into SRI here January 1st, they've developed a chip to do a lot of the computer vision algorithms. So the hope is, they'd be able to make the whole processing unit small enough to run up on board the Raven B and again, save the bandwidth and get higher quality images.

Q1: What were some of the other robots you started working on in the early days at SRI?

Bob Bolles: We had a Unimate robot, made by Joe Engelberger, which started the Unimation company, which he'd used for a long time. It was mainly a large pick and place. It could pick up 100 pounds and move it. We worked on some perception algorithms for them to detect big parts. I think there was a lot of concern there for human factors. If people could pick up the engine block, but they would ruin their back. So the idea is, could you do it in a way that would save them – they could basically do the more technical things and have the big weights being used by the Unimate arm.

What other robots have I used? SRI had developed the shaky robot, which I did not work on. It was pretty much finished by the time I came. It was developed at the end of the '60s and shown through the early '70s. We've worked on some other mobile robots. Three or four years ago, we had a program from Darpa called the learning applied to ground robots, LAGR, and the goal there was essentially a race. They would put you in a place and give you a GPS coordinate and say, "Go there." There could be fences and hedges and paths and tall weeds and all sorts of things, and the idea was, could you learn what the properties of these are, so that you could know that it's safe to go across this, and not over this deep sand or water or whatever. One of the things that we developed for that, that turned out to be a really important key, was a technique for locating the robot.

The way the race worked is they would let you run this course three or four times and then throw out the worst one. Whoever had the lowest cumulative score wins. So if you could help map something in the first one and could remember it and find your way back there, then you'd know that this is a good way, or this is a blocked way. But you have to be able to do that pretty precisely. So if there's a fence and you're a meter off, and you think you're on one side of the fence and you're not, then your map doesn't do you much good. So we developed – we, Modey Agerwald who works here, developed a visual odometry system that used the stereo cameras on board the robot to visually watch and measure how fast it's moving and detect where it's going, to do essentially SLAM, simultaneous localization and mapping. But he used the other sensors as well. So he had wheel odometry, but it would get confused every now and again. It would slip in sand going up a hill, or if things were wet, it would slip. We had IMU,

which would drift. It was basically good over short periods of time, but over long periods of time, it would drift.

We had GPS, which worked, except when we were inside buildings or under really heavy trees. And we had visual odometry, and it would work most of the time unless you're looking at a plain wall with no texture on it that it couldn't see any features. So that mix of those four sensors made it so our team could locate itself quite precisely and reliably over this few hundred meter course. So that meant we really could reuse the data we'd gotten in the previous runs, so we could get better and better. That was critical. I think there's a lesson there. Everybody has talked about robots that have redundancy, multiple ways to do things, but the expenses and the size and the complexity has been such that not many people have really done it. So I think we're now at the place where we're going to be able to have robots that have multiple ways to do things and they cross check each other. They can sort of be self aware. Probably one of the biggest troubles on all the robots is that they break. When they break, sometimes you don't know whether it's hardware, electronics, software or something. We don't usually use the industrial robots. We're usually using an experimental one. Like with ARM, we're running Barrett Arms, which are the LAM arms and they've been used for a number of things, but they break. Frequently we can jam the fingers and do things.

You'd like a system that's self aware enough to say, "My left finger is jammed," or, "The software to control the neck didn't boot up right." There ought to be something that basically automatically diagnoses the thing, because otherwise, in the past, you'd often come in and say, "I've got the whole day to work on this thing," and then something's not working. You spend the whole day just debugging what it is, and you just felt like, "Well, gee, the gods didn't want me to work today. I'll come back. We'll try to debug it today and come back tomorrow." Again, I think there we're getting close. Like the WAM arms now have temperature sensors on all the joins, so they have a direct way to measure things. They can measure the location, of course, the positions of things. I think looking at things, to look at the hand and say, "Wait a minute, your fingers are not all open." You commanded them to be open but they're not and your position device, something's messed up, because it says it's open but it's not. That kind of cross checking, and I think this one where they were using basically four sensors to cross check each other and figure out what the fusion should be is probably the way we're going to be going.

Q1: What kind of psychological theories or visual perception theories were influential on your work in robot vision?

Bob Bolles: So David Marr, way back when, had theories on perception that influenced all of us, I think, in terms of how to do the pixels, to symbolic information. There are a couple of fellows here at SRI actually, when I came, Marty Tennenbaum and Harry Barrow, who came out with a little different theory called intrinsic images, where the idea was that the world really is

3D. To understand it, you have to understand the surface normals and the textures and the colors and the lighting and all of those things rolled together. Sometimes you may be in a controlled lighting situation, but not know the texture. Other times, you know exactly what the texture is, but you don't know where it is. So their point was to show this general interrelationship of all these aspects, and sometimes you might know a couple of these that would narrow it down and help you locate the others. Biology in general is pretty humbling, to see how effective – throughout the whole career, you would often show a demo to somebody like my mother or my father, and they would say, “It's turned around, obviously. It's upside down.” Just because everybody does it so easily.

Q1: Did you take much inspiration from JJ Gibson, ecological perception, the sort of embodied kinds of vision?

Bob Bolles: Yes. Harry and Marty Tennenbaum took it quite seriously, so sort of by osmosis, we were the group here. Our group has gone from ten up to maybe 15 or 20. It's now at about eight or ten, so it floats around. People in the group have different feelings. Some are much more computational. They don't care whether it's biologically inspired. I like it to be biologically inspired, but I'm a pragmatist, so I'm quite happy to do it in a way that they don't. For example, early on, after we did this two dimensional recognition, we started doing – there was a group there that had developed a LIDAR sensor, one of the very first. It was very slow, took an hour to scan the whole scene or something, but it was a range sensor, not biologically inspired, unless you think of acoustical things, like dolphins, bats, things like that. But yeah, so in that sense, it could be, but it wasn't in the human inspired sort of sense. But I was quite happy to use 3D range data however I got it. If it was from stereo, that's fine. That's biologically inspired, or trinocularly. It's a little bit stretching it, but there are multifaceted or multi eyed things. So I was quite happy to work on range data. My feeling was pretty simple. The objects are 3D, you want 3D data. Otherwise you're starting one D behind already. So you really wanted to have it. Nowadays, I had predicted that range sensors – this was 30 years ago – were going to take five, ten years, and we're just now seeing practical lasers and stereo and flash LIDARs and things like that, that are becoming practical. They all still have a few quirks and stuff. Back then, sometimes at a conference, I would be viewed as a heretic for using laser data. It's not fair to them. I didn't care. I had a problem to solve. I was quite happy to do it from engineering.

Q1: Did you use a lot of sonars, range finders of other kinds?

Bob Bolles: A little. I personally haven't. Our group had used those some, but they're pretty poor. So when planar laser scanners like SIC and even now the Hokuyu and some of those, they are so much better that sonar was kind of a poor stepping stone in that direction.

Q1: What about the use of visual flow? Do you need real time data to do that?

Bob Bolles: Yes. Harlan Baker and I, maybe 20 something years ago, took a sequence of maybe 100 or so images where you had to take one image at a time, move it. So that was video 20 years ago, right? It took forever to do it. But we developed something called the epipolar plane image analysis, which essentially showed the paths of the features in the scene, that there were patterns to them that you could capitalize on them and locate where they were in 3D. So it's like structure from motion, but we took it – we didn't start with a general case. We started with the case where we were moving in a line. In a line, we could show all the features were moving in lines too. And if the features were in line, you could find lines pretty easily. From that, you could then compute where it was in 3D. So motion, that was the first thing we really had done in structure for motion. Since then, we've done a number of things. The visual odometry is essentially doing that. For the robot, since they had stereo cameras, we were using stereo, so we have a monocular version as well. So you can take a camera and move it around and get a picture of the world.

Q1: What were the evolutionary steps in between that you built on to get to this visual odometry?

Bob Bolles: Let's see. So the things that were most important were finding local features, tracking them and then doing the mathematics. There had been a lot of local feature kinds of things, and as you know, local features have gotten better and better. Probably the most popular ones now are SIFT, which have a description about them in their local area. But they often take too much time to computer even now. So here at SRI, we've developed CenSurE features, which are similar to that, but much faster to compute. It's not intellectually the most important thing, but it's something you need to be able to detect and then track them over time. Once you have these threads of tracks, then you need to do the mathematics to compute where the camera went and what the structure of the world is. When you have stereo for it, then you can compute 3D locations of all these features directly, and now tracking how you're moving through 3D features is pretty easy. So we did that first. In fact, that's what I mentioned on the LAGR robot. Going from there to single, monocular cameras, the mathematics was a little more complicated. There's a group at Sarnoff and some other places that showed that you could do this in a nice way.

We had – jumping way back – another fellow and I here at SRI, had worked on the problem of computing where the camera is relative to some scene. So we developed a new closed form solution for, if we knew objects in the scene and we could project them, we could compute directly the six degrees of freedom of the pose of this. Now the photogrameters had done that for years, but they required a human to point out and say, 'That's the corner of the building, that's the top of the wall, that's the other corner of the building, here's the door.' So

the humans were involved in doing the feature association and they didn't make many mistakes. I mean, a few. Sometimes it's hard to tell exactly where the point is, but basically they didn't have many mistakes. But when we were doing it automatically, trying to figure out where this camera was, the computer matching problem would generate a lot of mistakes. So we'd have a lot of good points and a lot of mistakes. So we wanted to compute where this was.

So we came up with a technique called RANSAC, for random sample consensus, that would sort through this bag of feature matches that you have and pick out the ones that were – that could be interpreted in a way to compute a coherent position for the camera. We wrote a paper back in early '80s that described it. It was actually ACM magazine at that point. We were the most proud about our closed form solutions for these, because they were complicated mathematics. But the thing that really stuck was the RANSAC method and now RANSAC's used for all sorts of computer vision things all over the world. We've become famous for that, not our nifty, cool mathematics. So you never know, see what's going on. I guess a few years ago, there was a special workshop just on RANSAC. They were celebrating the 25 years of RANSAC. That was fun. I went and presented a paper and things. I haven't kept up with all the variations that people have done to improve it. In different situations, there are ways to be even better at it.

Q1: What are some of the big applications today that use the RANSAC?

Bob Bolles: So almost any fitting thing that's done. So for example, if you have range data off this table, and you know that sometimes there'll be points like on the pen or on the cup or something like that, but a large percentage of them are on the table, then people use the RANSAC method. They'll take a handful of these points, fit a plane, find out how many other – so if you had a point up here and it fitted a plane, you wouldn't have many other coherent ones. But if you get points on the table, we get a whole bunch of coherent ones. You say, "That's bad. That's not on it. This isn't, that isn't," and so forth. So fitting of anything, ellipses, lines, geometric parts, it's sort of a standard tool now that they use, knowing that there are what we called gross errors. There are a lot of gross errors mixed in with the really good data. This is a standard technique that's easy to understand and I think probably easy to implement, so it was picked up.

Q1: What are some of the other mobile robotics or arms that you might have worked on in the '80s or '90s?

Bob Bolles: It might have been late '90s, there was a group here at SRI that had a project to do with Darpa to explore an unknown building and map it, and then find people in it, and then sort of guard it, watch it. So the approach that they used – and I was only on the side of this project, but a little bit involved – where it was called the centabots, because they literally used 100 robots

to explore. This floor here, there's actually the third floor of the building next door that was not being used, so they used that as one of their main things. For the actual test, they went to some abandoned warehouse in Texas and they hid some things and people in this. They said, "One, two, three, go," and they sent the robots in. So some of the robots had these laser scanners that they could map and do really well. Others just had sonar, so they were going around by themselves, and they had sort of primitive monocular cameras, so they could detect somebody moving. They knew where they were, and they set up communication chains.

The idea was, basically, to send robots into a building before you send people, so you could go in and say, "There's a person there, a person there, and this one's moving. That one's going there." So we still have probably 20 of those robots around. We were involved with a robot with Kurt Konolige who was here, called Flaky. It was hereditary, with Shaky being the original one in the late '60s. Flaky, you could talk to it and it would follow you and do some initial things, initial tasks. The idea was to say, "Flaky, go bring me a bagel." On Wednesdays, they sell bagels here. The idea was for it to be able to navigate itself, avoiding the people and things, and go get a bagel and come back. The speech recognition worked fine, except that when anything went wrong, your pitch and excitement would – you had taught it, say, "Stop." But when you really needed it, you said, "Stop!" It said, "Excuse me? Say it again." That wasn't very helpful. They've worked on that. That was one of those clear examples. Alan Alda came for Scientific American Frontiers or something and filmed during the day. They had Flaky following him around and sending things to people and stuff like that. They wanted a little clip of Flaky actually pushing him against the wall, which we didn't want to do. I think they eventually took a still of it. Flaky was about garbage can size. It's in the front room, up where we got nametags. I clearly come down on the side of helpful robots, constructive, keeping people from hurting their backs, not the monster versions that attack people and push them against the wall. We were not so pleased about that. I understand why they want to do it. It's not our real choice.

Q1: Where has your funding come from over the years?

Bob Bolles: Most of my funding, starting at Stanford and here, has been through DARPA, Defense Advanced Research Project Agency, largely because they've had substantial resources to explore some of these things. Now the funding has transitioned dramatically over the last 30, 40 years, from robotics and computer science is real important. This is not quite true, but basically, do good stuff and explore some interesting, novel things, to now, they're much more focused on a lot of them. Although, like the ARM project, the idea is, a lot of the current arms that are being used in the military, like for bomb disposal, what they really can do is pick up the bomb, by teleoperation. They don't do it automatically, and they can't do things that are very delicate, like separate two wires and cut the red one.

That's the sort of thing the goal for ARM is, to be able to do things in a more delicate way, one. Two, to be able to adapt to whatever it is. If the wires get stuck and you're trying to separate them, then you need to be able to do something a little differently. Or if the part's not exactly the right size or it has different weight or something, you need to be able to do it. Ultimately, this redundancy, they'd like you to be able to reach inside like a gym bag and feel around, without normal visual perception and say, "Let's see, there's my baseball, or there's my whatever it is." So it's pretty fundamental. As I say, the thing that's exciting to me is this marriage of real time perception and real time control that in this manipulation world – manipulation's pretty hard. Again, people do it very easily. It makes it look easy and our attempts are not at that level, anywhere close.

Q1: One of the biggest challenges of trying to realize machine manipulation?

Bob Bolles: One is this adjusting for the position. There are always calibration errors and things like that. One way to do it is to literally watch your hand reach around something so you don't hit it. You watch it go down and grasp it. Once you've done it, you feel it. Your fingers are in the right place. You can see it come up with you. It's not still lying there. So it's this multi checking that we'd like to be able – our goal is to develop a generic way to do that kind of multi-checking. So you tell me about a part. It may not be exactly the same thing, or it may be full of water or not, so it weighs different amounts, that we need to be able to adjust and say, "Ah, it's a little heavier." And to be able to automatically do that. In industrial applications, things are very tightly constrained, so the weights of things are known and the lighting's known and all that sort of stuff. Here we're trying to do away with all those constraints.

Q1: What's the difficulty of coupling the vision there? Is it the bandwidth, the amount of information?

Bob Bolles: The difficulty is, all the different components have quirks, meaning computer vision can't handle some very smooth surfaces like this. It's much harder to recognize than this, which is kind of counterintuitive. You'd think that the easier objects, like nice smooth blocks would be the easiest ones. Well, they're not. They have the least amount of features on them so they're actually, from a perception point of view, hard to measure and detect. That was an interesting comment on the original robotics work done at MIT and Stanford and other places, where they would literally worked in a blocks world, thinking it simplified things. But in fact, it made it harder. If you had the world with lots of nice textures, you could actually see it more easily. Similarly, probably, for touching things. If it's really slippery, it's hard to hold onto those. If you've got a softer object that's got more texture, you can pick it up and manipulate it more easily. So one is dealing with the artifacts and knowing what the uncertainties are and how to deal with them.

So for one of the sensors, we have a flash LIDAR, the sensor is much better at locating things left and right than it is along the beam. The range data itself is not that wonderful. To grab things, then, you ought to take that into account. You ought to know that you can grab it this way better than trying to grab it that way. Then there's a number of things in the actual communication between the devices. Bandwidth's not quite right, but it's the representation of all the information and artifacts that I'm getting out of this sensor, maybe merging with another sensor, and trying to deal with things in real time. As I mentioned, one of our main goals is to try to come up with a generic way to represent the tasks and the object so that we can automatically insert these checking steps. Because we could do it now, but there's so many things that you would like to check, that you don't want to have to list them all, all the time. You'd like the compiler, in a sense, to compile them in and so when you pick up something, you verify it's there, you check that your hand's correctly positioned, that your fingers touch it, the weight's right, that you didn't collide with things. All of that ought to be essentially built right in for any object here. There are special cases, if you're trying to pick up a piano or something, but for a lot of things, you'd like a generic way that just gets applied.

Q1: In terms of that kind of software development, do you see movement towards more unified kinds of robotic operating systems and scripted languages?

Bob Bolles: Yes. For the ARM project, they provided everything to us in ROS. We'd done a little bit but not that much. It's nice, because open source, it's well supported. Brian Gerkey was actually here at SRI for a while. He's now over at Willow Garage. It's nice, because it provides a lot of the infrastructures that you would need. It seems to have taken over completely from the Microsoft robotics part, which was being treated rather differently. So ROS seems to be taking over the world. I haven't checked around the world to make sure that's the case, but locally, certainly, it seems to be doing well.

Q2: Why do you think it was treated differently than the Microsoft system?

Bob Bolles: Well, I think part of it was that they wanted to provide an open source community support and they really did it. That is, they would react in a day or an hour. If you sent in a comment saying, "I'm having trouble with such and such," either Brian or somebody else in the community would react quickly. That kind of support you can't beat. Microsoft wasn't doing that kind of thing. Microsoft also had much more, I would say – and I'm not the fellow next door. Regis knows a whole lot more about these things than I do. I understand that it provided a number of things that were good, but in a more fixed way. ROS opens it up and you can define your own robots easily, so that's good.

Q2: What kinds of changes have you seen in SRI through the years?

Bob Bolles: I guess the easiest thing to say there is that we see waves of things. So one wave is the fundamental and applied versions. It seems to go in stages. There's a reaction that you say, "Hey, you've done a lot of fundamental things. What can you do for me on this particular thing?" You do that for a while and then you realize, "Wait a minute. There's some fundamental things I still don't know." So now we do ARM and Mind's Eye and stuff like that. Then there's another, "Okay, you're doing well there. Let's see the applications." That's happened here. In the AI Center itself, which is about 80 people now. When I was first here, it was maybe 35 or something like that. There's also been a wave of how big projects we're working on. Are we getting money for five person, 10 person projects, or are we getting money for one person or half a person projects? We see sometimes funding has gone through waves like that.

We've also seen things, some associated with this fundamental versus applied, is how much engineering obviously is really required. How critical is it to make it so somebody else can use your code? The atmosphere here hasn't changed that much in terms of ten years ago, we had to start wearing badges. People didn't like that. But from the government's accountability thing, they needed to make sure. We do, do some classified. A number of us have clearances so we do things that are occasionally classified. We have the capabilities to do it. So some of the time we've done that, like in mapping and things. We've done a project in the open and then we use our clearances to take the real data and run them in special computers and see how they work and then improve and enhance and so forth, so see where they can be really applied. That's been sort of nice. We don't normally do a completely black program. There are parts of SRI that do that, but in our group here, we've not done that.

Bob Bolles: – so every time we get together, we –

[audio ends abruptly]

Q: So, who have been your most important collaborators or people that you've had a lot of intellectual exchange with over the years?

Bob Bolles: Right, so, I mentioned before, Lou Paul at Stanford was probably the most influential one. I'd say his knack for picking out good next projects to work on was really inspirational. Tom Binford, who was another professor at Stanford, or he was a research associate. He had a very definite aesthetics on the quality of computer vision, and you were talking about Gibson and things like that. Those things he knew well, and he wanted all of us to consider seriously. Here at SRI, I mentioned Harry Barrow and Marty Tenenbaum also generated their idea of how computer vision could be viewed and was certainly important. I've had – Marty Fischler, the fellow who I worked with – he was actually my boss for a good while, but the two of us worked for 10 or 15, 20 years together. He just recently retired. He and I

developed RANSAC, but we did a lot of other things. He had a lot of really ingenious ways of looking at a problem, and then I started out doing a lot of the implementation and then sort of worked up to so we could compare notes and improve ideas. So, that worked out really well. Let's see, people outside of this – outside of SRI and Stanford, I've had – there was some people working on a robot when we worked on Autonomous Land Vehicle for Martin Marietta outside of Denver. We were doing a stereo project and two other groups were doing a stereo. So, we worked closely with them. One with Keith Nishihara, locally here in Palo Alto, and JPL.

And so we were developing stereo to be run in real time to detect obstacles in front of this robot. Now the Autonomous Land Vehicle looked like a bread truck, because this was back in the middle eighties, I guess. I don't remember exactly the timing, but the computers were big, and the – and then they needed big cooling, and the sensors were big. The laser scanner was like this big. The cameras, again, were sort of big. So, we were doing stereo on this and laser scanner analysis, but for stereo, we worked with JPL and Keith Nishihara here. JPL team went on to do work that went into the rovers that are on – both Opportunity and Spirit on Mars. So, we've stayed in touch with them. We still do similar things. We worked on robots through General Dynamics Robot Systems in Maryland. We were part of the Robotics Consortium, and we focused there on obstacle detection and classification. “What is it? It's a big lump, but is it something I'm interested in? Is it just tall grass or is it brick wall?” And, also people detection from the moving robot.

So, the army rightfully said, “Hey, you guys are able now to navigate. Well, you can go around this course. It's 10 kilometers long, and you can keep doing it well, and things – the weather changes and range and all that sort of stuff. You're doing that well, but what about safety? What if we've got people around? How are you going to detect them? And, if they're walking around, that's one thing, but if they're lying down. There are mechanics in the garage. How are you going to detect these things?” So, they looked at infrared and motion and geometry and changes and things like that. So, we worked on that project with them, as well.

Q: Were you involved with the DARPA challenge?

Bob Bolles: I was not. I went down to both of them, the – well, I guess there were three runs. So, I usually went down ahead of time, and we talked to the people and – because we know a lot of the people doing the work. We didn't originally, because in the first couple – in the desert, because they – the rule was if they – if DARPA was funding you, then you couldn't participate, because they felt that was unfair, right. For the Urban Challenge, they changed that so you could do it, but we didn't quite get our act together soon enough, and so I just enjoyed watching some Sebastian Thrun and Stanford people. We went down and cheered them on, and things like that. I mean, along that line, as you know, not too long ago Google announced that Sebastian's been working on that, and they have a fleet of six or ten cars that they've logged, whether it's 160,000

they said or some huge amount. So, my wife is counting on the robotics navigation systems to get good enough in the next 15 years so that she won't have to drive. So, whenever we meet some of these people, she says, "Hey! Are you on track to get this done? I'm counting on this by the time I'm, you know, whatever, but I don't want to have to drive." Because, right now her mom is 87 or something, and it's dangerous. She still drives.

Q: I have a new excuse for not having a license.

<laughter>

Bob Bolles: Right.

Q: So, apart from the self-driving cars, what do you see as the other big applications for robotics and robotic perception?

Bob Bolles: So, we were talking about waves and things. So, in the seventies, there was – when robots and computers and things started getting built, real ones, like Shakey and stuff, there was a huge media push. "It's coming. It's going to take over all your jobs. It's going to be great in your home and it's going to do all these things," but we couldn't deliver. It was just too hard to do it. All these easy things – quote "easy" for humans that they've learned are really hard to do from a robot. So, then there were – after this big hype, there was kind of a low period, and now in the last 10 or 15 years, robots have kind of infiltrated everything, and if you look around in medical things, now they're – like endoscopy, the da Vinci robot, which actually is a company that SRI started, spun off. We still have da Vinci – next generation da Vinci things in the basement – in the floor below us. And so, in medical things, they can do it better. That's more tele-operated, but turns out things like hip replacement – Russ Taylor, one of the fellows I knew at Stanford is now at Johns Hopkins, and there the robots can do better at hip placement because they – than a human, because they can make the joints smoother, which they – means they heal faster, more reliably, less infection, and all that sort of stuff.

So, in medicine, in cars now there are automatic cruise control that'll slow you down if you get – and warn you about a lot of things. So, it's sneaking in there. It's sneaking in education. There's robots now – Lego robots in elementary school, First Robotics in high school. So, if – everywhere you look, there are robots, and it hasn't been this huge media hype, fortunately, <laughs> but they've sort of come back around. I think some of the things that are going happen – there's obviously the elderly care, which needs help. I think probably initially it's going to be some monitoring and communications. There are now several companies like Willow Garage, and, oh geez, Anybots and things like that that have little robots. So, you can

essentially send a robot down and you can now watch the meeting or attend a meeting, and you're – just your robot's in the meeting, but you're actually watching it.

So, I think that kind of communication for elderly is going to come in first. Then there are going to be things where it can monitor the – did they fall down in some way? Do they need help? And later it'll start doing things where it can go get things, check on medicines, and later even than that, probably be cleaning the house and doing dishes. I mean, that's what the cartoons all showed, right, was doing dishes and cleaning the house, but that's probably the third wave of – even for the elderly. Of course, manufacturing robots have infiltrated that and have been very active there for a long time. Let's see, what else? Perception is probably going to take off here, too. As you know, everybody's cell phones now have cameras, and those cameras are dramatically better than those 128 x 128 cameras that we started with. So, I think there's going to be more and more applications of things where you can use your camera to do something other than just take a picture and post it on Facebook.

I remember, a few years ago there was a group in Japan that you could take a picture of a barcode, and it would tell you what the product is, who makes it, what it costs, where he could find it cheapest on the web and all sorts of things. That's a little niche thing, but I think there are going to be more and more things that – and probably 3D, too. I think the 3D movies – there's been waves of that, too, but I think the capability now to do real-time stereo and – the world is 3D. Actually it's 4D, because they're moving. You really want something that, just like I was saying, 2D and you have 2D image, that's okay; 3D and 3D data, that's okay. For motion and 3D world, you really need something that deals in that inherently. You don't want to have to deduce it all, because you make mistakes. That's my bias, but...

Q: So, where do you see your own work going in the near and somewhat further future?

Bob Bolles: Mm-hm. Let's see, I'm hoping, like in – both Mind's Eye and ARM are projects that I'm excited about. ARM has already said, "I like this idea of mixing and really getting control and perception together." And by perception, I mean touch and weight, forces and things like that. Not just visual. And I think that's – now's the right time to do that. We've got devices that are reliable enough to really run experiments and do it, and there are a lot of control issues in how do you – again people do it very easily. They can insert their hand in something pretty narrow and avoid obstacles and stuff, but that's still not thoroughly known, and how – where I might – so, for example, one little – not so little problem but is "How should I place my hand knowing that I want to see it, as well as grab the object?" Alright, that's, again, a simply stated problem, and we can do it in a lot of easy cases, but in general, how do you do it, knowing that they've got six florescent lights here. What's that going to do to my image? And if I'm using infrared laser spots, what does that do and so forth. Though you really want to be able to do the comp – basically use full access to computer graphics to predict what it's going to look like and

what you can see, what you can't see and what's occluded, and how you might get feedback to do the next little delicate thing I need to do. How can I see that the wires are separated? So, when I cut them, I don't get them both. I only get one, because I not only have to get access for my fingers, but I have to get access to the perception and know the – taking into all the quirks, the artifacts.

I think that's – we're stuck with that. I don't think we're ever going to get away from devices that have strengths and weaknesses. That's just sort of the basic thing. So, I – so, this marriage of control and perception is one. The other is a deeper reasoning. Perception, a long, long time ago talked about top-down with models and things, and we've done it in industrial cases where we have very specific models, but we've done less with generic a priori information that we know is going. We really need to have a better marriage of perception and this reasoning. As I mentioned, there are not many computers that know that if I put some – like the perception of an object, I put it behind there, and any two-year-old knows it's there. They're not surprised when it pops back out again. Computer vision, they're still surprised. They still don't really reason at that level, and so I think – so there the ontology and representation people, the reasoning people, the theorem-proving side of the house and things, they're – they've pretty much been in stovepipe of their own. They haven't really dealt with real problems.

There are exceptions, of course, but in general they've – this whole thing is separate and perception's been separate. I think that's got to get together, too. So, you know a container. You know what that means. There are tops and it holds certain things, and it holds some kinds of things better than others, and the way you get in – if you the view the house as a container, you can get in through the door, you can get in through the windows, you can get in – if you're air, you can get in through this, that and the other. That's yet to come. Mind's Eye is the first little step in that direction, but I think's that's – it's going to make – when we can do that in a pretty solid way, I think we're going to be much better off. It'll also help this checking part, because then they'll be able to say – the common sense saying, “I did this. This should happen.” One part that we have in ARM that we would love to do, but we are so busy doing other things we're not going to do it, is we have microphones on the robot. So, if we drop something, we can say, “Hey, I just heard something hit.” We have the capability to do that, but we don't have that integrated yet. That sort of thing, people all the time – if there is a certain kind of noise, you look. You're well-trained, because that's what you do, and the robot ought to be able to do the same thing. It ought to be able to deal with all those things in a coherent way.

Q: Do you have any concerns about some of the potential military applications of your work or as these technologies may go to other kinds of militaries who are developing robotic systems?

Bob Bolles: So far, I've focused mostly on the detection and – me, personally, and I've been pretty comfortable with that. Or, I'm helping – at least I view myself as helping the U.S.

hopefully save lives by finding out more information about what's out there. So, don't send somebody into a house unless you know what's in there, and there are things, again, from a biological term, that turns out there are radars and stuff where you can look through walls so you can tell that and go on in. I would just as happy to do that sort of thing. Am I worried – clearly there's a potential that somebody could use it for more nefarious, more dangerous things. It's possible. I haven't – I don't see – I think it's still a pretty long way away before there's a serious danger. I guess the closest thing might be detecting and tracking somebody and shooting, actually having autonomous shooting in that sense. The U.S., the rules on engagement that that doesn't happen. There's still a person in the loop, but you could imagine something whether it would work. Still a ways away, but...

Q: I just had one final one, I think, that related to your previous discussion about bringing together this reasoning perception of these different disciplines or different groups that were interested in different kinds of aspects of robots, like control and representation or these various things. So, what do you think were the reasons why this didn't come together earlier? Were they mainly technical or were there particular kinds of social reasons for that or how the universities or something were organized?

Bob Bolles: So, I think what happened was that they started out together, and so they said, "Geez, we need a system that does top-down, bottom-up and all this sort of reasoning and representation and everything," but then they realized, "Wait a minute. This is much harder than we thought." There's – Minsky had a little summer project, and they were going to do perception and they were going to finish it that summer, and then do the rest of it. And I think that was true of all of these different areas that when you got into them, you realized, "Wait a minute. Time, I thought was pretty easy. It's linear," but James Allen came out with his representation of temporal intervals and things, and that took a lot of thought, and it's been adopted by other things. But then there's regions and areas and all sorts of things that other people have added to it, and it's taken years. So, I think what happened when you start out together and, you're, "Woo, this is really hard. Here's an interesting way to solve this problem." Then, there's a little group that forms around that, and then here's another interesting way to solve the time problem, and the people get interested in that. So, I think there's a natural specialization.

So, it takes some special insight to say, "Hey, it's time to put these together." Now, in putting them together, it's often hard, because there's completely different languages, like the representation I was talking about. Ontologies, you go and ask them, "Well, how do you represent your ontologies?" Well, they have OWL and nine different kinds of languages on top of each other, and I'm thinking, "All I wanted to know is this thing a kind of plant or is it a –" <laughs> And they do the same thing with vision. They say, "You mean, you can't tell that's a cup? It's on its side, but you still ought to be able to tell it's a cup." So, there's communication, and you have to set up an agreement on it's okay to ask dumb questions and find out what's

capable and what isn't, and that really is a barrier. It's true with us in control, meaning – us meaning perception, because some things I think they can surely do, but they can't. It's hard to do it. They can maybe do it, but it's special purpose or something. It's not a general way to do it. Similarly with us – when they look at us. So, when I say the time is ripe, I think these communities have matured pretty well in the representation, control and perception, and I think there's – the benefits from putting them together are going to be substantial, not just a little increment. In the past, I think, “Well, geez, just a – I might get a little bit from it, but I have to learn this whole new language and stuff, and I don't – that's too much trouble.” So, I think the maturity level is important. So, you can get to something that you really could use, and when you bring in a little group that's helping you do it, then, and they're – you have to have this mutual agreement. As I say, you have to have a negotiation that says, “Okay, I don't understand what you're – I don't understand what's hard.”

Q: I'm also curious as to whether this part of the reason why these things are now having to be put together is also this focus on real world applications in a lot of different ways, where it's not easy to say like, “Okay, I won't worry about that one. I'll fix it with by making sure that the environment is very structured or something.”

Bob Bolles: Right. So, they handle it another way where they constrain it to do away with it. One of my pet peeves with – this is sort of related to what we're talking about. At DARPA when they do a project, they often want to – they want to make sure it's self-contained. So, that means they can get the data for it, that they can set up some milestones and see progress and say, “Hey, we're doing it better, more precisely, faster, whatever,” but because of that, they often don't want to give you everything that you might have available. So, for example, if you're sending a robot out ahead of me to scout around, the soldiers all have good maps. They have – depending on where they are and how recently it's been done and things, but they might have lidar scans nowadays. They might have good can – satellite and aerial images. You've got good topographic, and so my thought is, “If I'm sending a robot out there, I ought to get all that, too.” And sometimes they've been able to do that, but often they say, “Nah, it's too much trouble for me to get all that information. You just start here.” Where, that seems wrongheaded-thinking to me. If a soldier has that when they're going out there, the robot ought to have it, and the robot can enhance it and improve it and send it back. And the soldiers, when they come along a little later, can use it.

So, and I – that's changing now again partly because some of these communities, again, their representations of all this information is getting better. It's more prevalent that you have these different kinds of maps and where the roads and buildings and heights and things are, but it's – people have had – perception has often had to do a lot more than I think they should have. So, if you start with a model, it's a whole lot easier to say, “Oh, there's that building,” then trying to construct one from scratch. And so I – some of the practical applications, you really do come in with a lot of information, and you're not adding any new constraints, you're just using

information that's available and could simplify things in a – both for robustness and speed capabilities.

Q: So, apart from RANSAC, what do you look back on and see as your biggest accomplishment or the thing with the biggest impact?

Bob Bolles: Well, the industrial part recognition, probably, the 2D and 3D things. They were LFF and the 3D we called 3DPO, which was fun. Let's see, one of the things I view (it's not as well known) is the epipolar-plane image analysis, because that was sort of the first structure for motion where we showed that we can convert things into linear. Let's say, if you can detect lines, you can then detect how far things are in the world, which is important. One of the – it's still not a solved problem to take an arbitrary world with sensors and build a good 3D model. There are lots of applications. We had a company maybe ten years ago come and say, "Geez, we spend a lot of money with picking up sofas that we've delivered to somebody's house because when they get it there, it doesn't look the way they want it. So, if you could just go scan their house for us, and we could stick this in there virtually, then they could say, 'Hey, I like that. It would be better if it was a little longer or it's a little shorter or a little – or if you didn't have that purple pillow on it.'" It's still not quite possible. With laser scans you can do pretty well, but – and it's even getting where you're going to probably be able to take your iPhone and do this and get a model of it, but it's still not quite there yet. So, there's plenty for us to do in the future.

Let's see, things – the motion detection from a moving object like a robot or a plane or something like that, that's – we were certainly not the only ones to be doing that, but that's an important step, because then you don't have to worry about physically having your camera fixed. You can zoom and pan and fly it around and bounce over things and stuff. That was a big step. Hopefully, we'll be able to brag about some things with ARM and Mind's Eye, but haven't gotten there yet.

Q: We'll come back to that later. <laughs>

Bob Bolles: Right.

Q: I think we're done with all the questions we had, but if you have anything you'd like to add or anything you've missed.

Bob Bolles: I guess, one – it's not earth shaking, except that we found that pairs of people work the best. So, on a project – we often have projects that might be five or eight-person projects, but still pairs seem to be the best, and part of the reason, it seems, that it's hard to keep a third person up to speed. So, if there are two of you, you can go knock on his door and get excited, and you

guys talk about it, and you change your idea this way. If there's a third person over here, they're behind already, and then so somebody has to go get them up to speed on that area, and if they go off, then you're behind, right, and then – so, pairs are easier than threes and pairs are better than ones, because one – people tend, including me, tend to go off sometimes. So, I need corrections to come back and say, "Wait a minute. That's a cool idea, but the real problem is over here."

So, from a working point of view, we've tried to encourage mixing and matching, and one side effect of that that we need to do better on is to have people that don't just work on one project at a time, because by working as a pair with some other thing, it broadens you and gives you new ideas, but often things – there's correlation between things that come back that kind of give you a new way to look at it. So, having a person work on two, but probably not more than two – once it gets to three or four, then there's change in gears so frequently during the week, that they – it's hard to make progress. So, twos and not just focusing on one thing at a time, partly because if you get blocked on one, you can't figure out how to solve it, go work on the other one and see what's going on. Or, if the person you're working with here is off, you can work on something else.

Q: I think that's good advice for any kind of work, really.

<laughter>

Bob Bolles: Well, except that when I was in school and even in college, you had to do your own work. We're not allowed to talk anybody. Sometimes they'd do joint projects, but it was almost always do your own work, but the world isn't that way, at least this world isn't that way. You really – having somebody else to work with and bounce ideas on and check your sanity is dramatically better. I think that's changed. I think our children had more opportunities to do joint things in a constructive way than we ever did.