EPISODE 452

[INTRODUCTION]

[0:00:00.8] JM: Augmented reality applications are slowly making their way into the world of the consumer. Pokemon Go created the magical experience of seeing Pokemons superimposed upon the real world. IKEA's mobile app lets you see how a couch would fit into your living room, which has a significant improvement on the furniture buying process.

Augmented reality applications can have even more dramatic impact on industrial enterprises. Have you ever set up a factory? You might need to build a conveyor belt, you might need to put together the parts of a giant machine that extrudes steel, you might need to fix a silicon wafer fabrication machine. It takes an expert to set up these heavy complicated machines. Scope AR is a company that builds augmented reality tools. One of the Scope AR products allows users to tell a presence with each other to collaborate on the construction and maintenance of heavy machinery.

Imagine I'm setting up a factory and I have a complicated piece of machinery, let's say a conveyor belt in front of me. I've never constructed a conveyor belt before. I put on a Hololens and I set up a VoIP call with an expert who has experience with that piece of machinery, and they point out what I need to do by superimposing 3D arrows and text and other instructions on my field of vision. They can share my experience and help guide me through the process.

This is such a flexible tool. You can imagine applications for augmented reality assistance being useful in medicine and construction and education and lots of other fields. Scott Montgomerie is the Scope AR, and in today's episode we talk about the state of AR. How the AR tools from Apple and Google compare and how the similarity between the tools used for mapping the world in AR relate to the tools used to map the world by autonomous cars to actually a very similar technology.

Scott was a great guest, and I hope to have him back in the future. We've done some other great shows about augmented reality and virtual reality applications, as well as the nature of reality. You can find these old episodes by downloading the free Software Engineering Daily app for iOS or for Android.

In these podcast apps, you have access to all of the episodes of Software Engineering Daily, whereas in your normal podcast player you only have access to the most recent 100. With these apps, we're going to build a new way to consume content about software engineering. Right now, it's only the podcasts, but over time we're going to expand it to other forms of content.

They're open-sourced at github.com/softwareengineeringdaily. If you're looking for an open source project to get involved with, we would love to get your help. Shout to today's open source featured contributor, Edgar Pino. He is working in a real-time chat application for Software Engineering Daily, so that we can have chat rooms for people to discuss the episodes easily. It's a really innovative piece of work and I'm happy to have Edgar contributing to the open source community.

With that, let's get on with this episode.

[SPONSOR MESSAGE]

[0:03:22.8] JM: You are programming a new service for your users, or you are hacking on a side project. Whatever you're building, you need to send e-mail. For sending e-mail, developers use SendGrid. Send Grid is the API for e-mail trusted by developers.

Send transactional e-mails through the SendGrid API. Build marketing campaigns with a beautiful interface for crafting the perfect e-mail. SendGrid is trusted by Uber, Airbnb and Spotify. But anyone can start for free and send 40,000 e-mails in their first month. After the first month, you can send 100 e-mails per day for free.

Just go to sendgrid.com/sedaily to get started. Your e-mail is important. Make sure it gets delivered properly with SendGrid, a leading e-mail platform. Get started with 40,000 e-mails your first month at sendgrid.com/sedaily. That's sendgrid.com/sedaily.

[INTERVIEW]

[0:04:37.0] JM: Scott Montgomerie is the CEO of Scope AR. Scott, welcome to Software Engineering Daily.

[0:04:42.1] SM: Thanks. Great to be here.

[0:04:43.7] JM: I want to talk about everything in AR that we can get to in the next 40 to 60 minutes. But let's start with the state of AR and what you're working on. What are the low-hanging fruit applications where AR is actually useful today?

[0:05:04.5] SM: It's interesting. The AR landscape has gone a dramatic shift in the past six months since Apple came out with ARKit. That's brought the word into the common lexicon now. Everybody knows what augmented reality is.

I still think that we're a ways away from finding the killer use case in AR. We've been focused on AR for over five years, and gradually grinding out the typical use cases that are really providing value. I think we're almost everybody sees the initial value. Augmented reality is in the enterprise.

Part of the reason for that is that augmented reality, it's still nascent with hardware, especially wearables. It's clunky. There's lot of limitations to it. But if you solve the right problems and saving a lot of money with enterprises, they're more than willing to overlook those limitations to get their workers to use this nascent technology. They're definitely leading the way.

In particular, I think learning augmented reality on top of remote assistance applications is really interesting. Part of the great aspect of augmented reality is that it allows you to communicate better more visually in a 3D space. It's just a lot more natural to interact in three dimensions.

In a remote assistance use case, you've got technicians and experts collaborating over video. If you're just using FaceTime, you're still telling a guy what to do. It's like, "Hey, look at that thing on the left or the right or whatever. No, not that left. The other left. No, not that left."

If you've got an augmented reality experience, you're able to add the three dimensional annotations on top of reality and it goes a long way towards improving communications and collaboration.

The second use case I think that everybody is exploring is work constructions. Being able to show you more in-depth illustrations and communicating how do you perform things. In a very basic way, you know with Google Glass, you've got text-based instructions that are just overlaid on top of your glasses.

The things that w do more commonly are actual 3D models overlaid on top of equipment, showing you how to do things. Augmented reality really allows you to communicate intention a lot better, and leaves a lot less room for error and miscommunication.

[0:07:19.8] JM: Yeah, you've describe the Scope AR product at this point. So if I'm – let's say I am running a factory, a company that does factories and maybe I am making – my factory makes speakers, for example. Within this factory that makes speakers, there is all these complex pieces of machinery. There is steel extruders or big complicated saws or things like that.

If I'm setting up a new factory, let's say I got one factory and it's going great and I decide I'm going to expand to another factory on the other side of the country, and the person who is setting up this new factory, they've got to set up all these complicated machinery and they got to understand how to use it. The people at the original factory, they know how to use that technology, but unfortunately they're across the country and they are not present with me setting up the machines in the factory.

This is where Scope AR can be useful, because I can get on a remote telepresence call with somebody at the factory that's already been set up, and they can walk me through the setup of these complex machinery while I'm wearing, for example, a Hololens. They can see what I'm seeing and can place little arrows and instructions and helpful visual guides that will be overlaid across my Hololens and will guide me through the process. Am I giving an appropriate use case of Scope AR?

[0:08:56.3] SM: Absolutely. That's exactly what we do. But there's a whole bunch of other enterprise use cases. They're similar as well, that we don't really focus on things like where I was part picking.

Let's say you're an Amazon warehouse and you are walking up to an aisle with a 100 feet stack of boxes, how do you know which part to go pull and do that as efficiently as possible? Well, if you use augmented reality to be able to highlight the box and guide you how to get there, that's a heck of a lot better efficiency than paper and trying to do it the old fashion way correlating an ID number.

Or they used to use QR codes and scan those, but that's also a little bit inefficient. If you can actually see that three dimensions where to put it, a heck a lot more efficient. Then in terms of packaging, you can get an algorithm optimized how to package boxes, and then it shows you how to do it and then you go ahead and do it. So you can save a lot on space and again more efficiency. It's all about that user interface lettering and allowing computers to help you do things in the real world and real task. That's really where we're seeing almost all the value in AR.

[0:10:07.3] JM: Let's through some more practicalities and then we'll get into the engineering. Are there serious workplace safety issues that happened because people don't have clear sets of instructions for how to use heavy machinery?

[0:10:23.0] SM: Yeah, definitely. We haven't seen a lot of benefit improving safety with heavy machinery so much. We've certainly seen a dramatic – I think the reason for that is that those industries, because they're so safety conscious they're a little bit averse in new technologies. They have the procedures, they're established, they've done it a certain way, because they've learned a lot of mistakes along the way.

They don't really want to change right now. In the same vein, we have seen a dramatic decrease in error rate in a whole lot of different use cases. So in terms of manufacturing and assembly instructions, and then in terms of maintenance use cases. Because you have these visual instructions that are guiding workers how to do things, the author of those instructions can communicate his intent of how to perform a procedure much more clearly than with just text-based instructions or verbally.

If we think of a really basic example, like IKEA for example, everybody screws up IKEA furniture. When I talk about the company, I would say three-quarters the time somebody is like, "Oh, man. I wish I had this for IKEA," because it's just a common thing that reach people up.

[0:11:30.7] JM: That's why they bought TaskRabbit, right?

[0:11:33.3] SM: Exactly. That's right. Exactly. But if you were able to see that furniture being built in front of you step by step in three dimensions, you would never put something on the wrong way or use the wrong screw or anything like that.

The whole reason we screw up for insurers, because the instructions are poorly communicated and we misinterpret them. If you take that to a much more complicated procedure and a much more high-stakes procedure like manufacturing or heavy equipment maintenance, it just makes a lot of sense and you can see why the error rate has significantly decreased. We're seeing in some cases, we're decreasing error rates down to exactly zero. Yeah, it's pretty impressive.

[0:12:09.9] JM: What about medicine? My dad is a doctor, I talk to him a lot about telemedicine. Are there applications today for AR in the telemedical industry?

[0:12:21.3] SM: Yeah, absolutely. We're in discussions with a couple of companies doing interesting use cases. I haven't got super far yet, but they're certainly excited. For us personally, we've stayed away from medical just because there's a lot of low-hanging fruit in heavy industry and we got a target, you know a certain niche.

But I think there's a lot of opportunity in telemedicine. I think one of the success stories in medicine that I've heard of is Augmedix. They're just using Google Glass as a recording mechanism. They're essentially using as a head-mounted video camera to record a session between a patient and a doctor.

That dramatically decreases the amount of data input that doctors have to go through, because doctors spend a significant amount of their time writing up charts and essentially doing basic data entry. They can't upload that, because they're the ones that understand the problem and have a communication with the patient. But if you can record that using Google Glass, then that's interesting.

It's not exactly a great use of augmented reality, but for head-mounted displays it's interesting. One of the use cases that we've – we actually have seen is training in terms of medical equipment, which is kind of an offshoot of heavy industry just applied to the medical field.

Hospital equipment in particular is very complicated and training is expensive and costly if you do it wrong. But again, because of the increased communication, if you can show an augmented reality how a particular device is supposed to be use, then there's an enmity benefit.

[0:13:38.9] JM: If I'm a Amazon warehouse worker and my day is spent walking around the warehouse and picking items off the shelves and putting them into boxes, the goal is obviously to get a robot doing this because this is not work that we should have humans do. This is highly automatable, and we're trying to figure out how to train machines to do this.

You could imagine somebody walking around with a headset that has a camera on it and recording their view of the world and the things that they pick, and you can imagine also giving them AR hint about where they should be going and what they should be picking from.

In an ideal world, maybe you would be able to use the data like – the hints that we're giving to the humans to train eventually machines and maybe you want to give humans the same kind of hints that you eventually will give machines, and then the machines can just learn to do the picking that way. Is that a fruitful avenue for us to pursue, or am I thinking down the wrong avenues of the future?

[0:14:50.1] SM: No. I think that's perfect. You're exactly right. Right now, robots need to be trained in how to perform tasks, and that by – A lot of industrial robots are trained by watching humans actually perform that task with computer vision.

Part picking is no different. Ideally, you're absolutely right, you got this head wearing a headset tracking where you are, probably mapping out the 3D mesh of the facility, understanding in 3D where the parts were going. That would entirely be possible. There might be more efficient ways to do it, but I think your line of thinking is right. Like I said, that's certainly one of the most interesting use cases of augmented reality.

[0:15:28.8] JM: You said we have seen industrial applications where machines have been successfully trained by watching a human?

[0:15:36.7] SM: Yeah. I mean, a lot of the robots used for assembly need to be trained somehow. Yeah, if you got a human hand that is able to assemble something, that machine can also see how to put it together. I'm not an expert in robotics, but I'm pretty sure I've seen this in some of the automotive factories I visited.

[SPONSOR MESSAGE]

[0:16:03.5] JM: At Software Engineering Daily, we need to keep our metrics reliable. If a botnet started listening to all of our episodes and we have nothing to stop it, our statistics would be corrupted. We would have no way to know whether a listen came from a bot or a real user. That's why we use Incapsula, to stop attackers and improve performance.

When a listener makes a request to play an episode of Software Engineering Daily, Incapsula checks that request before it reaches our servers and it filters the bot traffic preventing it from ever reaching us. Botnets and DDoS attacks are not just a threat to podcasts, they can impact your application too. Incapsula can protect API servers and micro-services from responding to unwanted requests.

To try Incapsula for yourself, go to Incapsula.com/2017podcasts and get a free enterprise trial of Incapsula. Incapsula's API gives you control over the security and performance of your application and that's true whether you have a complex micro-services architecture or a Wordpress site, like Software Engineering Daily.

Incapsula has a global network of over 30 data centers that optimize routing and cashier content. The same network of data centers are filtering your content for attackers and they're operating as a CDN and they're speeding up your application, but doing all of these for you and you can try it today for free by going to incapsula.com/2017podcasts and you can get that free enterprise trial of Incapsula. That's Incapsula.com/2017podcasts. Check it out. Thanks again, Incapsula.

[INTERVIEW CONTINUED]

[0:17:52.0] JM: That's pretty cool. Okay, well I'll need to interview somebody about automotive stuff. Let's get in to engineering. Scope AR was started – you just said five years ago, but I thought it was seven years ago. I think it's something like that, or –

[0:18:06.4] SM: Yeah. It's a little bit of a gray area. Officially we started in 2011. So yeah, it would be six years ago.

[0:18:15.7] JM: Okay. Sure.

[0:18:16.2] SM: I guess, unofficially it started even further than that. The way we had started was I had developed some computer vision technology for another purpose. I was actually trying to build a game for the Apple TV. This didn't really pan out, mostly because the Apple TV never came out with an app store.

Once the WWC came and went and there was no app store, I was like, "Crap, what the heck do I do with this computer vision technology I spent so long perfecting?" I realized that it could actually be applied to augmented reality.

We started looking at interesting use cases. We started looking at marketing and advertising use cases. But of course, this was 2011. Still, we're not really seeing advertising in marketing use cases take off. But certainly before AR was even then in a lexicon, people are not willing to pay for this crazy new advertising platform in AR.

But we had a customer that came to us and said, "Hey, can we use this for training?" We did a really quick proof of concept with them. It went really well. Then they said, "Cool. We want to show this at a tradeshow." This was a giant mining tradeshow in Las Vegas. It was a big company, so we had pretty much the best spot on the floor in this giant tradeshow.

It was a bit of novelty at the time. We're supposed to show it a few times a day over the three-day show, and we ended up showing it over a hundred times. Every time we showed it, people were coming up to us being like, "Oh, my God. It's the coolest thing I've ever seen." Which was really impressive, because the machines we were sitting beside, were like this six-storey giant mining trucks that you find in the oil sands in Northern Canada With us, with our little geeky augmented reality thing people were saying this was the coolest in the show. That was pretty cool.

The stories we heard from people about how they could see this in a field were really why we started doing what we do. One gentleman in particular came up and said, "This exactly piece of equipment you guys are showing, I've been maintaining this my entire career for 35 years, and just now I learned how to do it properly. I have trained hundreds of other guys to do it the wrong way. I have probably shorten the lifespan of this thing the way I do it. I probably cost my company tens of millions of dollars by doing it the wrong way. I need this technology. How do I get it now? Like sell to me right now."

Just hearing that kind of story we're like, "Wow, apparently we found something here. We should probably pursue this." That's really how we got kick-started.

[0:20:30.7] JM: Sure. It's interesting how many successful products come out of ill-fated games.

[0:20:37.0] SM: it's true. It's true.

[0:20:41.8] JM: You stumbled onto this augmented reality, instead of use cases. Did you have any background in computer vision? I mean, what did you need to know about computer vision six years ago to start an augmented reality platform?

[0:20:55.7] SM: Yeah. No formal training in computer vision. I actually did my degree in bioinformatics, which was a combination of computer science and genetics. Essentially, I was just writing algorithms and machine learning algorithms to analyze the human genome and do protein folding.

I've always been a guy that just if I have a problem, I figure out a way to solve it. Basically it just started playing around with open CV and started building things around open CV and doing augmented reality that way. Yeah, no formal training. Just grinding and figuring out how to solve the problem.

[0:21:30.5] JM: Give me a little bit of the story of what happened after that conference where you were talking to somebody in a mining company. I'm sure you came home from that conference with adrenaline pumping through your veins just thinking, "Wow, this is actually a huge set of opportunities. We could build augmented reality stuff to train people to do all kinds of stuff." Where did you go once you had that adrenaline pumping through your veins?

[0:21:55.7] SM: Yeah. Clearly it was very early. We'd stumble upon something, but we had absolutely no idea how to turn into a viable business. We started essentially consulting. We got contracts from guys like Boeing and Toyota and NASA, building out initial proof of concepts with them.

Eventually we realized that the – although the proof concepts we're building came down to a common set of use cases and something that we could build into a platform. This was a noble platform and it was going to solve a whole lot of the problems that were going to happen at some point when this became viable; scalability, maintainability, ease and speed of authoring these types of instructions.

After those, first of – few initial use cases and a whole bunch of others, we started building a platform to solve those and that was really the genesis of our products.

[0:22:45.2] JM: I want to just point out, again this is actually another interesting way that people explore ideas while hedging their bets is like. One, how do you explore a new space well build a game? Two, how do you explore a space once you found some kind of product market fit, but you're not exactly sure what the company looks like? You consult and you do consulting until you the understand the idea space a little bit better.

Once you understand the idea space and the enterprise customer process a little bit better, you can start to encounter, "Okay, what are the canonical use cases? How do we build an abstract platform for it?" Then you start to get to the real engineering problems that are going to give you the most leverage once you solve them. What were those hard engineering problems that you eventually decided to focus on?

[0:23:38.4] SM: Yeah. It was really about listening to the customer and just understanding their initial problems. I guess, the first big project we had was in many, many steps. It was about a 150 steps with a really, really complicated piece of hardware, which meant it was thousands or I think distinct parts.

So we built our software on top of Unity, which actually – I mean, this was 2012 I guess. Back then, it was one of the hardest decisions I had to make, which was choosing a three-dimensional rendering platform.

There was a whole bunch out there. There was you know obviously unreal. A few of the lesser known game platforms, there was like open source software and stuff like that. Finally, we settled on Unity. Mostly because when we first – actually let me back up. When we first built that proof of concept for that tradeshow, the customer wanted basically the impossible. They wanted augmented reality glasses and they wanted to have a reliable, good-looking demo to show at this tradeshow.

Really, the first start was they gave us a whole bunch of money and they said, "Hey, go buy a bunch of augmented reality glasses. Like everything on the market. If nothing is suitable, then buy some."

We did buy every pair out there. There weren't many and we had building some. That was incredibly challenging. That was the first and last piece of hardware I ever built. But essentially, what we ended up doing was we partnered with Epson, who had just launched a pair of glasses. They didn't have a camera on them, so they weren't really augmented reality-enabled.

We ended up hot-gluing a webcam on top of these glasses, running the webcam through USB, to a laptop. Running the computer vision calculations and the rendering on a laptop, opening it to VGA, converting to VGA to component and then hacking the operating system on the glasses to accept external component input and then displaying that VU in the glasses.

This is one of the first pair of AR glasses; total Frankenstein solution, but it worked as proof of concept and clearly kickstarted the company. But the real reason we chose Unity as a platform back then was because it had a great plugin architecture that allowed you to plug in various AR toolkits.

By this time, we had kind of moved on from my homegrown computer vision solution. I realized at that point that there were companies that were actually working on real AR toolkits. They're doing it much better than I could. I couldn't spend two million dollars in R&D trying to build an AR tracking solution that wouldn't be nearly as good as something; some of the commercial guys out there that are really focusing on it. Who actually had legitimate academic backgrounds in computer vision, and not just my hacky problem solving.

[0:26:12.3] JM: Even six years ago, people were building AR platforms.

[0:26:15.6] SM: Yeah. I mean, there weren't a lot. At the time, if I remember correctly there were three. Before it was owned by Qualcomm who is actually an acquisition. It was Metaio who got acquired by Apple and is essentially the basis for ARKit.

[0:26:28.9] JM: Wow. Okay.

[0:26:30.0] SM: Then there was another company out of LA, I can't remember now. Anyways, so –

[0:26:35.0] JM: Metaio? So Apple bought them and then developed it further. What did Metaio do, or when was that acquisition?

[0:26:44.7] SM: Yeah, Metaio had been around for a long time. I believe they started in 2006 or something like that. Although I could be totally wrong. They were based out of Munich and really initially were focused on the automotive industry.

They though the killer use case for augmented reality was basically allowing customers to customized the look and feel of their cars. If you could recognize the shape of a car and then change the paints and the grill and customize it, that would be really a compelling use case.

They focus on a lot of those things. The focus on a lot. They did a lot of different things. They were the same as us, they did a lot of consulting, but they also had this toolkit they released. They also had a very early version of the work instruction platform that we have that allowed you to kind of create a procedural animation-type thing.

Yeah, Apple acquired them in 2014, maybe 2015, I think. It was because they had the best computer vision out there, the best augmented reality algorithms. They were also working on some interesting things with bringing the algorithms in the Silicon.

The way you typically speed up calculations is first you write it in a high-level language like C Sharp, then you can optimize it further by bringing the C, then you can optimize it further by doing a lot of assembly for your really – the parts of the code that need to be really fast. Then if you can't get that fast with assembly, then you build them into silicon, where you actually have specific hardware that can speed it up.

That's how you get really speed of a graphic card. It's specialized hardware for doing those specific calculations. I think Apple's strategy all along was to take these calculations and bring it into the hardware to really speed it up. That's how AR is so great today.

[0:28:21.1] JM: Wow. Okay. It's incredible. This is what I hear about the Apple acquisitions. Like Apple acquires a company then you don't hear anything about the company for the next 11 years. Then Apple brings forth a fully developed augmented reality platform based on the technology they acquired 11 years ago, or 10 years ago, or 8 years ago I remember it was.

[0:28:42.7] SM: Definitely.

[0:28:43.2] JM: you said they were acquired in 2015, I think.

[0:28:44.8] SM: Something like that. I'd have to do the research. Either 2014 or 2015.

[0:28:47.9] JM: All right. So maybe like two to three years to bring it out of silence, to bring it to market. But that's so cool that they put it into the chip. I mean, this is what you see with the machine learning stuff more recently where everybody is trying to press the machine learning calculations into a special kind of chip, because why wouldn't you? You got matrix calculations that happen all the time in a machine learning model, why not make these more efficient by throwing them in the chip?

But AR. Help the listeners understand I guess a little bit what Scope AR does, because you were building stuff long before ARKit made it to market. Okay, I think here's the question, ARKit came out recently. You were building your own software up until then. Actually ARKit and ARCore came out recently.

These are Apple's augmented reality platform and Android, that Google's augmented reality platform respectively. Have you had to refactor the software that you built over time in order to run it on ARKit or ARCore? Because obviously, you would like to take advantage of the on-device speed up – I'm sorry, on-ship speed up that ARKit has, for example?

[0:30:08.6] SM: Yeah. Just to clarify what we do. If I look at the augmented reality market, I see three distinct areas and they're blurring into two. But there is the computer vision calculations that essentially take the measurements from the camera and other sensors like the IMU and combine them to get a pose.

A pose is essentially your orientation in 3D space and your position in 3D space. By getting that pose, then you can augment your reality by placing a 3D object in the camera relative to some surface or something.

In order to do that, you need really fast calculations to keep your pose updated, so that you can change the position of that – those 3D models relative to the camera image. Or and a pair of glasses relative to your vision. That's one of the role of computer vision in augmented reality.

The second piece of it is the hardware. Whether it's phones or tablets or smart glasses, they need to be good enough hardware to be able to do those calculations fast enough. But the display need to be responsive enough to update the image relative to where you're looking, so that you have a good experience.

If you don't have a fast enough processor or refresh, then you get a concept that we call swimming, which is you can move your device, but your camera doesn't update as quickly, or the position of the image doesn't update as quickly. So it lags. That means that the connection between your superimposition of this object over reality loses its connection to reality. So you get kind of an uncanny valley type of effect, where it just doesn't become real anymore.

Where we really focus is the third piece, which is content creation. As soon as you have really good hardware and really good computer vision and you can create rudimentary experiences by placing things in your reality, now you need to make that useful.

Much like, back in the early days of computer we might have image programs to create basic images. Then somebody was great and wanted to string a whole bunch of those images together in a video or in a PowerPoint presentation to illustrate something. So you needed additional software to make that easy to use.

What we do at Scope AR really is focused on that content piece. Our work line product is like PowerPoint for augmented reality. It makes it easy to bring in 3D models from your manufacturing processes and then build out a workflow to illustrate a procedure. It's like a visio-flowchart type thing.

For each step in the flowchart, you can associate text video, audio and video and text speech just like you would in a normal procedure. But then you can also associate these 3D models and superimpose them on top of real equipment to show how to do things. Then the second piece we do is the remote assistance piece. This is content creation created in real-time between an expert and a technician to collaborate and really solve a problem.

Getting back to the original question, going back to the history of the company, so when we first started it there were a whole bunch of different computer vision toolkits that were available, or I guess not a whole bunch but two or three. More have definitely popped up recently.

We realized that the market was going to shift really quickly. These toolkits were coming out frequently, hardware was being updated frequently. Now we're seeing a new device or a new set of computer vision technologies coming out every few months.

We realized that it was really necessary to be agnostic, so that we could really quickly take advantage of the latest and greatest technologies. We really built a layer that abstracted all of the hardware and computer vision capabilities into the platform.

What that meant was that when AR came out, it took us two or three days to actually adopt the ARKit platform, because it had all the same components and APIs that all the rest of the platforms did.

You need essentially to get your camera image, you need to get the pose, you need to get – there's a few other things you need to get, but it's a pretty simple abstraction. We are essentially able to plug and play. We'll do the same thing ARCore, we did the same thing with Hololens. If there's another toolkit that comes along, then we'll be able to adapt it pretty quickly as well. It's really just good software engineering, thinking ahead to be able to abstract it so you could easily plug and play.

[SPONSOR MESSAGE]

Go to softwareengineeringdaily.com/raygun and try it out for yourself.

[INTERVIEW CONTINUED]

[0:35:41.1] JM: Got it. I've seen a little bit. This is I've toyed around with ARKit that if you want to build, let's say a – if you want to build the Instagram for augmented reality where instead of putting a photo filter on a photo, or putting a filter over a video, you want to have augmented reality assets come into that video, and then you can imagine the bright future where we're all wearing augmented reality glasses or contact lenses and we walk past a restaurant and a little pop-up comes up over our glasses that says, "Hey, 50% off a meal at McDonald's today if you come in."

That's a pretty appealing advertising platform, but it's actually really hard to do for a number of reasons. One of which is it's – there's not like an off-the-shelf way to build augmented reality assets. You've got to build these ad hoc 3D models, you've got to figure out how to get the 3D model into your app, you've got to figure out how to serve it to users.

Then that's not even talking about how we're going to integrate sound and whatever else with those 3D models. You're talking about building a workflow manager for doing all of that stuff. It sounds pretty useful and it sounds pretty flexible, and it sounds like you also were – you have been interested in the whole augmented reality as an advertising tool yourself for a while, so I'm sure you have had these kinds of thoughts go through your head.

[0:37:19.7] SM: Yeah, absolutely. There is definitely a lot of moving pieces to it. Some of which exist now and some of which don't. For example, if you are going to do this augmented reality advertising platform where you're looking at McDonald's and wanted to show an ad, there's a lot of pieces that go into that.

The first piece, and this is where we're at and it's pretty basic, is that you need to know – let's say if you got a pair of AR glasses, you need to understand where those glasses are looking, where you are in space both at a sub-millimeter level, but also – sub-millimeter relative to the ground and to the store. But also, you need your GPS coordinates to recognize which McDonald's you're looking at and how far away it is.

We can do that now, but the next thing is it would have to recognize at a pretty accurate level and not necessarily millimeter, but where that McDonald's is in real space so that you could overlay that ad on a McDonald's, then place that content in there and stuff like that.

We're a little bit far away from that. Computer vision isn't great at recognizing objects yet. It's going to come, especially with the new machine learning features coming big into Apple's products and a few others, but also cloud-based machine learning platforms.

You can do some object recognition rudimentarily, but then moving that into an augmented reality context and really understanding what it is and overlaying on top of that, that's really challenging and not really feasible right now.

One of the most common features we get asked is if I'm looking at a piece of machinery, can it recognize that I've removed a nut or removed a bolt or put a wire in the right place? The answer is no. We're barely at the point where we can actually track you properly and know where your phone is in real space, at an accurate enough level to show animations on top of that.

But recognizing really small changes in a scene is really challenging. That's going to require a lot of machine learning to understand that that is a nut in fact and that it has been moved from previous frames. It's just a pure content problem that that nut has been moved off of a bolt. So to recognize that and then to program it to recognize that, that's another challenge entirely. There's a whole space of problems that are yet to be solved and areas of businesses that are going to be solved at some point.

[0:39:40.2] JM: In the area of self-driving cars, you hear the debate between people who are saying we need LIDAR, which is the lasers I think – lasers that help you understand where you are in space and maybe they help with mapping. I need to do some shows on cars, but that versus computer vision, so you can –

There's different ways of doing this simultaneous localization and mapping where you can just do computer vision, where you just have cameras around your car and you're bringing in the world an understanding from the computer vision the depth and where you are in space and where other objects are.

You can also do lasers, you can also do I think radar or bouncing sound around places to understand what's going on. It sounds like there is similar debates in the AR world, because for example, Apple has a new phone coming out, I think that has lasers in it that are related to ARKit. Can you help me understand this conversation?

[0:40:45.7] SM: Yeah, absolutely. It's all about the amount of information that you can perceive about the worlds, and having senses to understand what's around you. Similar to humans, we have many senses, we have touch and hearing and I say to be able to understand where we are.

Then that's all tied into a centralized system in our ears, where the only reason we could stay standing up is because there is a tight integration between our ear and our mandibular vestibular system with our eyesight, so that's why we get seasick. When you start to play with those senses in VR, for example, a lot of people get seasick, because what they're seeing doesn't line up with what they're feeling and how they're moving.

Similarly in the computer vision on the autonomous driving world, you need to merge a whole bunch of sensors and derive information out of all that raw data. There's a whole debate around processing power and how you get that in real-time, because of course that's necessary to make decisions for autonomous driving. Then it's necessary in augmented reality context to be able to figure out where you are in space as I mentioned and manipulate your pose to update your 3D models.

One of the reasons why ARKit was so amazing is because it was able to do really good augmented reality experiences with only an RGB camera. An RGB camera is just a plain old camera, the same thing you'd find in SLR camera or a webcam that you've had that technology for decades.

We get to a red, green and blue set of pixels. ARKit essentially combines that camera, the information coming out of it with the IMU, so accelerometer and your gyroscope to get that pose estimation at a very fast rate and very accurate.

I've been in the augmented reality space for a long time, and all previous toolkits that are out there that try to do this without fiidiucial markers, they could do it, but A, they had very high CPU usage, which made it really heavy on battery consumption and not viable for a lot of applications. But B, if you moved your phone too fast, or if you moved it out of the area of initialization.

Even the best ones on the best hardware, you'd get like 45 degrees of freedom. By that, I mean if you're looking at an object, a coffee cup for example and if you rotated it 45 degrees around that coffee cup, it would usually lose, or at least move, drift the estimate. That's because of the accumulation of errors.

Because you're trying to merge the sensor information from your IMU with your camera, sometimes they don't match up and so calculation errors start to accumulate over a period of time, especially if you're moving your phone quickly, then you completely lose it.

What Apple has been able to achieve is a really great experience where it essentially never loses the tracking. That's really remarkable. Previously, if you wanted that level of detail, you needed to do something like Google Tango, which incorporated a depth camera.

This was another device that it was essentially a laser, so it fired off an infrared laser in front of you and then reflected that laser back into a sensor on the phone, kind of like a flashlight. It measured the time difference between each pixels. What that meant was you actually got RGBD, so D stand for depth.

You got a really accurate depth measurement. Using that information, you could plug that into the tracking algorithm and get that level of tracking accuracy. That also gives you some really other – interesting other things.

With RGB, essentially you get a – it's called a point cloud. If you can track various points in the camera image, if you get a slight difference between – if you move your phone even a few millimeters, points that are further away are going to move less in your camera image than if they're closer to you.

By measuring the difference between those points, you can get a measurement of the three-dimensional distance. By doing that, then you get a point cloud. But it's a very sparse point cloud. You might get about a thousand points in any given camera image at best and on average under a hundred. But with a depth camera, now you have a very rich point cloud. In fact, every pixel in your camera is now a point in 3D space.

You can do really cool things, like building really complicated and detailed meshes. You can go around and scan a room, you can scan a chair and objects and you can recognize objects. Yeah, it's differential between the two of them.

[0:45:13.0] JM: Okay. Let's go a little deeper there. So any frame that I am seeing in the world is basically a bitmap, right? Every little pixel on my screen is representing an RGB, but in the world where we have a laser in our camera, there is also a depth associated with each of those pixels. Am I understanding the mapping correctly?

[0:45:46.7] SM: Correct. Yeah.

[0:45:47.6] JM: Okay. That gives us a more accurate view of the world, especially from the point of view of the – I guess it gives our computers more accurate view of the world. In order to generate the depth that is associated with each pixel on a frame, does the laser that's getting reflected back, how hard is it to map the feedback of the laser accurately with the image that's coming in through the other camera? Are there multiple cameras, or is everything coming in through the same camera all at once? Do you know much about that, or does that get too much into physics and stuff?

[0:46:34.2] SM: No, absolutely. There are two cameras. One is the plain RGB camera, and then the second one is the depth camera. The way the depth camera works is it shines its laser on a particular pixel and then waits for the reflection of that laser beam. It's got a unique time stamp or a signal on it, so it recognizes that that reflection is that pixel.

[0:46:54.4] JM: Oh, my God.

[0:46:55.2] SM: Based on the timing of how long it's taken to reflect off of that surface, you know how far away it is.

[0:47:02.1] JM: Oh, my God.

[0:47:02.2] SM: We're talking extremely, extremely accurate measurement, and especially if you're doing that many, many times a second for every pixel in your frame. You were talking like – I'm not what the scale is, but I'm guessing either nanoseconds or femtoseconds. But you're able to get a fairly accurate depth map. We've done a whole bunch of testing on this.

You can get sometimes half a centimeter in accuracy and the only reason it's not that accurate is because depth cameras are really power intensive. Lasers aren't cheap. You're shining this laser out, and if you shine that laser out many times a second, the CPU frequency needs to go up and yada, yada, yada. Yeah, that's essentially how a depth camera works, is it's capturing that time of light.

[0:47:44.8] JM: It's no surprise to me that Apple is pouring money into this, pouring resources and pouring strategic efforts into ARKit, because this is an area where their manufacturing expertise can keep the costs reasonable while still getting super futuristic hardware features. You can imagine this as being a durable competitive advantage against Google as Google tries to move up market in the smartphone world. Now is ARCore doing something similar to ARKit?

[0:48:24.5] SM: Yes. It's yeah, almost a direct copy. There is reasons for this, and I think it plays into more of the strategy that we're seeing in a Google. Just to get down on the weeds a little bit more, what you need for the amazing experience that ARKit has provided is tight integration between the hardware units.

Your IMU is doing measurements many times a second and then your camera is doing measurements many times a second. You need to fuse those two sensors many times a second to get your calculations.

As I mentioned, the reason why previous efforts of this had failed was because of accumulation of error. What's really required is a centralized time ship, where when you get a measurement from the IMU, you get the time stamp on it incredibly accurate and you can line that up with the timestamp on the camera incredibly accurately.

By doing that, you can minimize a lot of the air, because if you have kind of – you can imagine if you have offsets between the measurements, your IMU – the measurement from your IMU was done a split second after the measurement from the camera and that you can't account for the very small amount of movement and so your calculation is going to be wrong, and that error will accumulate over time.

Apple has been able to do this, because they control the entire hardware stack. They've had this integrated time ship from day one essentially. With Google, because they've got such a fragmented ecosystem and hardware, and because Android manufacturers can only really differentiate themselves on a few key pieces now, one of which being the camera, they really don't have this tight integration where –

The time ship might be on the – at the SOC for one ship, which is distinct from your camera unit. Then you can't really do the proper calculations and your error accumulates. With Tango, what they did was they mandated a certain bill of materials, which was you needed this depth camera this specification, this camera this specification, this ship this specification.

Manufacturers like Acer and Lenovo were able to manufacture these and deliver the experience that Google required. With ARCore if you noticed, ARCore is only going to be supported on a couple of phones; IE, the Google pixel and the Samsung S8.

My guess is that's because they got caught with their pants down when Apple did ARKit. They really quickly went to Samsung and HCC who manufacture the pixel and was like, "Okay, we need tight integration between these things in order to produce ARCore and keep up with Apple."

Very shortly after, they acquired HTC which indicates to me that they really want to take control of the hardware top to bottom. I think there's a few other reasons for it, but I think that was probably a large reason for it.

They're a little bit behind in the VR game against Facebook with Oculus, Daydream – they've done a great job with Daydream, but I think they wanted to bring the hardware for the vibe in-house and really be able to drive that.

But then in the augmented reality game, if they were able to tightly integrate the manufacturing with HTC with their pixel line, that will just help them in the future. I think they are trying to take more control.

It's really interesting, because historically Apply tried to control the hardware back in the early Mac days, and that strategy failed miserably. The strategy that Microsoft took, which was – Intel which was the piecemeal – you could build computers yourself and blah, blah, blah. That really worked out in that space.

But what we've seen in the smartphone space is that modular componentization really hasn't worked out. Google's Project Aida, or whatever it was to try to model the smartphone and failed miserably. Now they're following Apple's lead and building really tight hardware integration top to bottom.

[0:52:07.0] JM: You got to love the Apple haters, who have been talking about how Apple doesn't innovate and Apple holds on to too much cash and they just accumulate cash. Well, sometimes you got to hold a lot of cash, because you're going to have to do some crazy investments in hardware advancements and like holding onto the platform, which is like – Apple is in a business where they had to stockpile cash. I think now we're going to start to see why that is.

[0:52:36.2] I completely agree. I think they're spending on this space and R&D is going to be tremendous, and we're going to see some really cool things come out of what they're doing.

[0:52:46.0] JM: Yeah. Okay. I know our time is getting somewhat short and there's so much more I wanted to ask you about. But let's quickly go over Hololens in this world where Apple is suddenly leading the way with its hardware integration and Google is struggling to catch up by acquiring HTC and trying to integrate its hardware all of a sudden. Where does Microsoft sit with the Hololens?

[0:53:11.0] SM: Yeah. Interestingly, Microsoft led the way in this. The Hololens came out way before ARKit and ARCore, although if you go back Apple was obviously doing acquisitions in the AR space many years ago. I really like the whole lens. I think it's a fantastic device.

They issued essentially what ARKit did with really robust tracking in a head-worn device. There's a lot of problems with it. The field of view is small. It's shocking when people first put it on. The whole user interface and how you interact with the operating system is not great. Gesture interaction is not great.

They've made a voice control through Cortana, so that's great. Overall, for a first try at a true augmented reality device, I would argue that Google Glass was not a true augmented reality device. This is the first real viable AR device with really robust tracking, which is really what you need for a good augmented reality experience and they've just nailed it.

I'm really excited to see what comes in the next version now that they've learned that they put it in market and they've learned what the core competencies of the device are and the killer use cases, as well as the deficiencies. I think the next one is going to be fantastic.

[0:54:24.0] JM: Can you contrast it? How does Microsoft fair as a hardware company these days?

[0:54:28.9] SM: Well, it's interesting. They gave up on all their phones. But I think they're really making a big bet in the AR space with Hololens. Then the mixed reality platform they're building, where they've got a whole bunch of partners for virtual reality.

Yeah, I think they're doing everything right. I think they're keeping up with – Actually, I mean Facebook didn't do so well with Oculus. Oculus's sales haven't been great. I think they really overpaid for the platform. HTC came out of nowhere and really dominated that for a long time. But Microsoft, I think is doing a really great job at really identifying that niche and I'm really impressed, to be honest with you.

[0:55:07.7] JM: have you seen Magic Leap?

[0:55:08.7] SM: No. I've tried a bunch of times, but yeah haven't seen it yet.

[0:55:15.3] JM: Have you heard anything about it?

[0:55:16.1] SM: Not that I can say.

[0:55:19.7] JM: Okay. All right, we'll have to get coffee some time.

[0:55:23.7] SM: Definitely.

[0:55:24.3] JM: Maybe you'll tell me. Maybe you'll tell me off the air. Okay, well I guess we're nearing the end of our time. This was great that we definitely had to have you on again in the future hopefully. Okay, why didn't Google Glass work?

[0:55:37.1] SM: I think Google Glass was released prematurely and they put it out there – I think it got out of control, honestly. I think they put it out there to see the public's reaction. Somebody decided that a mass release and it was going to be socially acceptable and this was great.

The public backlash was just – we all know happened. But I really don't think they thought through the use cases. It was really not ready for primetime. I got mine, and I was really excited about it for about the first week and then I put it away and then it hasn't seen the light of days since that first week, which is crazy for a $1,500 device. That was a giant waste of money. I think most people felt that way.

I was at Google I/O the year they launched it and virtually everybody had theirs on as a badge of honor. Just a year later, I saw like one or two guys with it on. That fad really faded pretty quickly. It was because you couldn't do anything with it.

You could maybe take a screenshot, or you could send an e-mail really quickly or a text message, but it was nothing you couldn't do with your smartphone, and almost arguably easier. I thought the killer use case would be when I go cycling, I would be able to take screenshots and keep up with my text messages, but it didn't work.

While I'm cycling, the voice control didn't work because it was too loud. I couldn't take my hands off my handlebars to play with rudimentary gestures. That killer use case didn't work. I really couldn't find the utility for it. I'm not sure and clearly people couldn't.

Like anything, there are killer use cases for it and I think they found a nice niche in the enterprise again. That's delivering really rudimentary tech instructions in your field of view to workers. That solves the problem of workers not having to carry around papers or tablets that can get broken or dirty and they're tough to handle with work gloves on. Now you can just talk to your device and it shows you rudimentary things.

Again, coming back to the enterprise, that's really where the use cases are first being adopted with this admittedly clunky technology. But if it saves time and money, businesses will adopt it.

[0:57:41.7] JM: All right, Scott. Well it's been great having you on the show. Next time you have a product launch or you give a conference talk that you want to get some broader appeal to, let me know. I'll have you back on the show. It's great talking to you.

[0:57:52.2] SM: That would be great. Pretty nice talking, you too Jeff.

[END OF INTERVIEW]

[0:57:57.4] JM: Simplify continuous delivery with GoCD, the on-premise, open-source, continuous delivery tool by ThoughtWorks. With GoCD, you can easily model complex deployment workflows using pipelines and visualize them end to end with the value stream map. You get complete visibility into and control over your company's deployments.

At gocd.org/sedaily, find out how to bring continuous delivery to your teams. Say goodbye to deployment panic and hello to consistent predictable deliveries. Visit gocd.org/sedaily to learn more about GoCD. Commercial support and enterprise add-ons, including disaster recovery are available. Thanks to GoCD for being a continued sponsor of Software Engineering Daily.

[END]