

Why Virtual Reality Is Hard (and where it might be going)

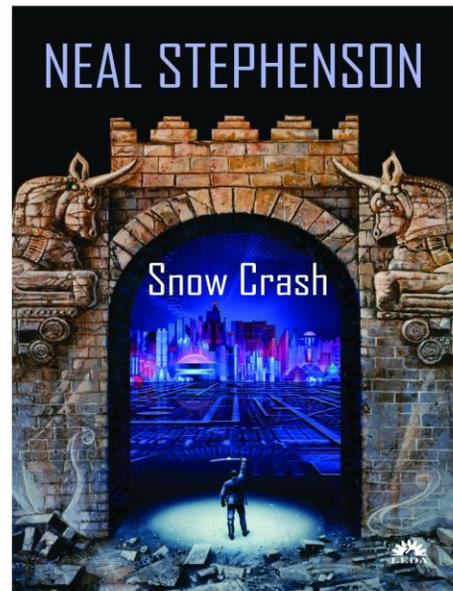
Michael Abrash

Developer, Valve Software

Good afternoon. I'm Michael Abrash, and I'm part of the group working on virtual reality at Valve. Today I'm going to share as much of what we've learned as I can cram into 25 minutes; I'm going to go fast and cover a lot of ground, so fasten your seat belts!

Flashback: 1996

- The *Metaverse* is coming!



17 years ago, I gave a talk at GDC about the technology John Carmack and I had developed for *Quake*.

That was the most fun I ever had giving a talk, because for me *Quake* was SF made real, literally.

You see, around 1994, I read Neal Stephenson's *Snow Crash*, and instantly realized a lot of the Metaverse was doable then – and I badly wanted to be part of making it happen.

The best way I could see to do that was to join Id Software to work with John on *Quake*, so I did, and what we created there actually lived up to the dream *Snow Crash* had put into my head.

While it didn't quite lead to the Metaverse – at least it hasn't yet – it did lead to a huge community built around realtime networked 3D gaming, which is pretty close.

Helping to bring a whole new type of entertainment and social interaction into existence was an amazing experience, and it was all hugely exciting – but it's easy to forget that *Quake* actually looked like this:



And it took 15 years to get to this



2013: Virtual reality is coming!



With that in mind, let's come back in the present, where one of the key missing pieces of the Metaverse – virtual reality – looks like it might be on the verge of a breakthrough.

Before we get started, I'd like to define a few terms:

Virtual reality, or VR, is when you're immersed in a purely virtual world.

Playing Team Fortress 2 in an Oculus Rift would be VR.

Augmented reality, or AR, is when real reality, or RR, is enhanced with virtual images that appear to coexist with the real world.

Playing a game on a virtual chessboard that's sitting on a real tabletop while wearing a see-through head mounted display would be AR.

Most of what I'll say about VR today applies to AR as well.

The key commonality between VR and AR is that virtual images appear to exist in the same frame of reference as the real world.

So when you move your head, virtual images have to change correspondingly in order to appear to remain in the right

place.

This tight association between the virtual world and the real world is how VR and AR differ from wearable information devices such as Google Glass. It's far harder to keep the two worlds visually aligned than it is to just display heads-up information.

Why is it different this time?

- Flat-panel displays
- Batteries and power management
- Mobile CPUs and GPUs
- Wireless
- Cameras
- Gyroscopes, accelerometers, and compasses

Of course, we've heard that VR was on the verge of a breakthrough before; why should we believe it this time?

There's no knowing for sure at this point, but it looks like this time really may be different, due to a convergence of technologies, including a lot of stuff that was developed for mobile but is useful for VR too.

Why is it different this time?

- Projectors
- Waveguides
- Computer vision
- Tracking hardware
- Content: 3D games

There have also been tremendous advances in head-mountable display technology, including projectors and waveguides, as well as in computer vision and in hardware that's useful for tracking.

And finally, for the first time there's compelling content in the form of lots of 3D games that can be ported to VR, as well as a thriving indie game community that will jump in and figure out what's unique and fun about VR.

Not just around the corner



Also, this time VR isn't just around the corner.

It's here now, at least in the form of the Rift development kit, and the Rift is on a path to shipping to consumers in quantity.

The Rift has a good shot at breaking through, for several reasons:

It has a wide field of view – that is, the display covers a large portion of the eyes' viewing area;

It's lightweight and ergonomic;

It's affordable;

And there's potentially lots of content in the form of ported 3D games.

Most important, gaming on the Rift is highly immersive. I remember how blown away I was the first time a rocket trail went past me in Quake – it's like that, but on steroids, when a rocket goes past in TF2 in VR.

So the Rift has huge potential, and is very exciting.

But...

The Rift is only a start

- Room for improvement across the board
- It will take many years to fully refine VR
- AR is even harder

It's still very early days for the Rift – this is only the first development kit – and for VR in general, and just as was the case with *Quake*, there's lots of room for improvement in all areas. Some of those areas are familiar ones, such as:

Field of view

Mobility

Input

And resolution. The math for resolution together with wide fields of view is brutal; divide 1K by 1K resolution – about the best an affordable head mounted display is likely to be able to do in the next year – into a 100-degree FOV and you get a display with less than 1/50th the pixel density of a phone at normal viewing distance

Other areas for improvement are unique to VR and not at all familiar, and we'll see some of those over the remainder of this talk.

The bottom line is that as with 3D, it will take years, if not decades, to fully refine VR.

AR is even harder and will take longer to make great.

The main point of this talk is to get you to believe this and to understand why it's true, so you can make rational plans for VR game development now and in the future.

There's no way I can give you a proper understanding in a 25-minute talk of the complexity and depth of the issues associated with making virtual images seem real to the human perceptual system, but I can give you a sense of the breadth of those issues, along with a deeper look at one specific problem, and that's what I'm going to do today.

How hard can it be?

- What's so hard about mounting a screen in a visor and displaying images?

It seems so simple – isn't VR just a matter of putting a display in a visor and showing images on it?

That actually turns out to be hard all by itself.

But solving it just gets you to...

The really hard parts

- Tracking
- Latency
- Producing perceptions indistinguishable from reality

The really hard parts.

There are three really hard problems that if solved would make VR far more convincing

Tracking

Latency

And stimulating the human perceptual system to produce results indistinguishable from the real world

All three are different aspects of the core problem, which is the interaction between the display and the human perceptual system

You may well wonder whether head mounted displays are really so different from monitors.

The answer is yes – and more so than you'd imagine, for two key reasons.

Images must seem fixed in space



The first reason is that, as I mentioned earlier, in order for virtual images to seem real, they have to appear to remain in the correct position relative to the real world at all times. That means, for example, that as the head turns in this slide, the virtual view has to change correspondingly to show the correct image for the new head position, just as would happen with a real-world view.

This is obviously required in AR, where virtual images coexist with the real world, but it's required in VR as well, even though you can't see the real world, because you have a good sense of your orientation, position, and movements even without the help of vision, thanks to hardware and software in your head that's effectively a gyroscope, accelerometer, and sensor fusion filter.

VR displays move differently

- They move relative to both RR and eyes
- They move with your head
- Your head can move *very* fast
- Your eyes can counter-rotate just as fast

The second reason is that, unlike monitors, VR displays move rapidly relative to both the real world and the eyes.

In particular, they move with your head.

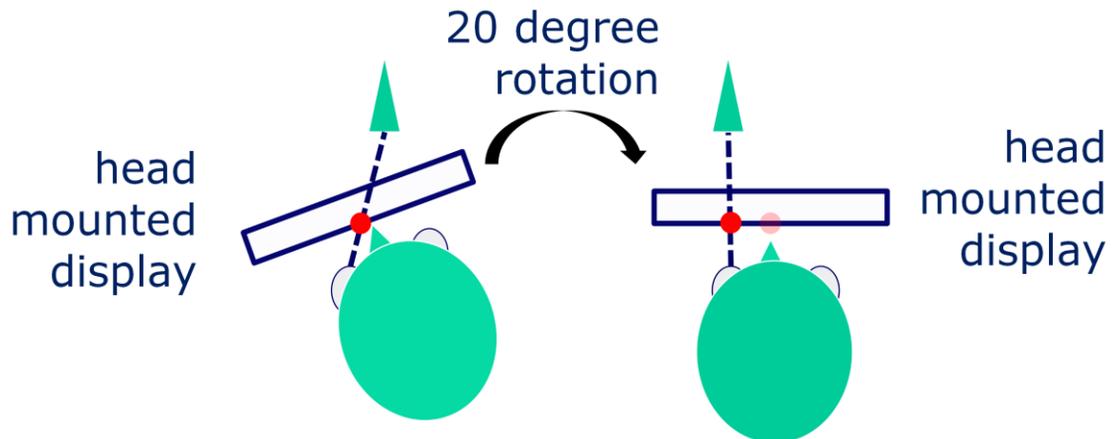
Your head can move *very* fast – ten times as fast as your eyes can pursue moving objects in the real world when your head isn't moving.

Your eyes can accurately counter-rotate just as fast.

That means that if you fixate on something while you turn your head, your eyes remain fixed with respect to the real world, but move very quickly relative to the display – and they can see clearly the whole time.

It's important to understand this, because it produces a set of artifacts that are unique to head mounted displays.

Rapid relative motion



This slide illustrates the relative motion between an eye and the display. Here the eye counter-rotates to remain fixated on a tree in the real world, while the head, and consequently the head mounted display, rotates 20 degrees, something that can easily happen in a hundred milliseconds or less. The red dot shows the pixel on the display that maps to the tree's perceived location in the real world, and thus the orientation of the eye, before and after rotation, with the original position shown with a faint red dot in the right-hand diagram. You can see that the red dot shifts a considerable distance during the rotation. While it's true that the effect is exaggerated here because the tree is about six inches away, it's also true that the eyes and the display can move a long way relative to one another in a short period of time.

This rapid relative motion makes it very challenging to keep virtual images in fixed positions relative to the real world, and matters are complicated by the fact that displays only update once a frame.

That results in a set of major issues for VR that don't exist in the real world, and that are barely noticeable at worst on monitors, because there are no useful cases in which your

eyes move very quickly relative to a monitor while still being able to see clearly.

Much of the rest of this talk will be dedicated to exploring some of the implications of the relationships between a head-mounted display, the eyes, and the real world.

The really hard parts

- Tracking
- Latency
- Producing perceptions indistinguishable from reality

The first implication has to do with tracking.

Tracking

- Pose: head position and orientation
- Images must always be in exactly the right place relative to RR and the head
- Perceptual system is tuned for anomalies

By tracking, I mean the determination of head position and orientation in the real world, known as pose.

Images have to be in exactly the right place relative to both the head and the real world every frame in order to seem real; otherwise the visual

system will detect an anomaly, no matter how brief it is, and that will destroy the illusion of reality.

The human perceptual system has evolved to be very effective at detecting such anomalies, because anomalies might be thinking about eating you, or might be tasty.

Tracking

- So tracking has to be super-accurate

In order to keep the visual system from thinking something is wrong, tracking has to be super-accurate.

How accurate?

On the order of a millimeter at 2 meters distance from the sensor.

There is currently no consumer-priced system that comes close to the required tracking accuracy and reliability across a wide enough range of motion.

How close is VR tracking now?

- Rift: inertial measurement unit (IMU)
- Pros:
 - Inexpensive
 - Lightweight
- Cons:
 - Drifts
 - Only supports rotation, not translation

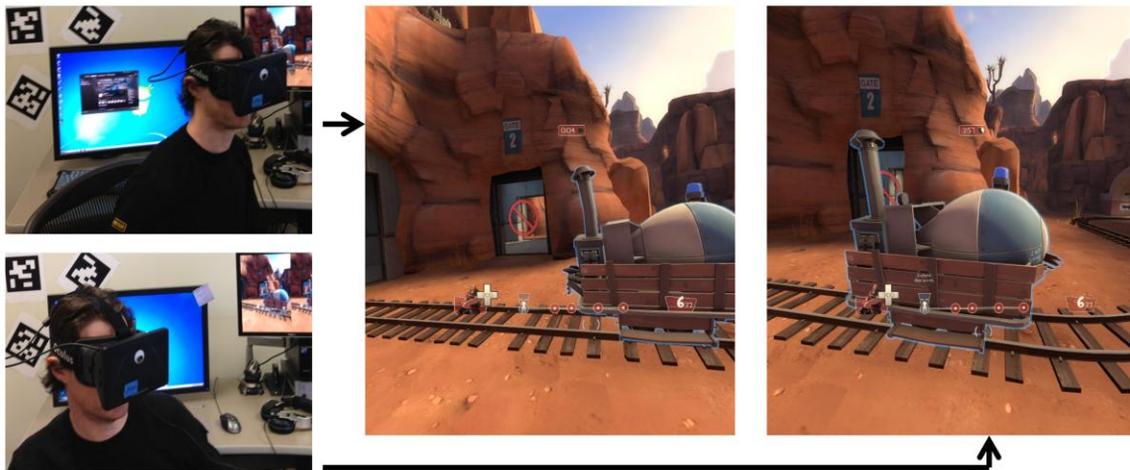
So how close is VR tracking right now to what we really want? Let's look at what the tracking in the first Rift development kit can and can't do.

The Rift uses an inertial measurement unit, or IMU, which contains a gyroscope and accelerometer.

IMU-based tracking is inexpensive and lightweight, which is good.

However, it also drifts because there's no absolute positioning, and it doesn't support translation – that is, it doesn't provide accurate reporting of head movement from side to side, up and down, and forward and back, and that's a significant lack.

Translation



Here we see translation in action. As the head moves from side to side, the virtual view changes accordingly, with nearer objects shifting by greater distances than farther objects. This produces parallax, one of the key depth cues and an important part of making virtual scenes seem real.

In the case of the Rift, translation has little effect on the virtual scene. I say "little" because there is a head and neck model that attempts to reproduce the translation of your head as it rotates, but it has no way to know about translation resulting from any other head movement.

For those of you clever enough to notice that my illustration of translation actually features a Rift, I admit that the translation in these screenshots didn't come from the head movement – it was simulated by strafing from the keyboard. But that is how translation would look if the Rift did support it.

IMU tracking works for games that don't require anything but head rotation – FPSes, for example.

But even in FPSes, the lack of translation means you can't peek around corners or duck down.

In general, the lack of parallax makes virtual images seem

less real.

Also, drift means that IMU-based VR can't stay stable with respect to the real world, and that rules out games like board games that need to stay in one place.

This is definitely a long way from where we really want to be, although bear in mind that this is only the first development kit, and Oculus continues to work on tracking.

The really hard parts

- Tracking
- **Latency**
- Producing perceptions indistinguishable from reality

Next, we move on to latency.

Latency

- Latency = delay between head motion and virtual world update reaching eyes
- Images drawn in wrong place
- Like tracking, introduces anomalies
- Varies with head motion

Latency is the delay between head motion and the corresponding virtual world update reaching the eyes.

Too much latency results in images drawn in the right place, but at the wrong time, which can create anomalies.

The magnitude of the anomaly varies with head motion; the faster the head moves, the greater the anomaly.

When the head reverses direction, the anomaly is briefly multiplied and becomes far more apparent.

Again, this destroys the "reality" part of VR.

It's also a recipe for motion sickness.

So latency needs to be super-low.

How low is super-low?

- Somewhere between 1 & 20 ms total for:
 - Tracking
 - Rendering
 - Transmitting to the display
 - Getting photons coming out of the display
 - Getting photons to stop coming out
 - Google "Carmack Abrash latency" for details

How low?

Somewhere between 1 & 20 ms total for the entire pipeline from the time head motion occurs through:

Tracking

Rendering

Transmitting to the display

Getting photons coming out of the display

And getting photons to stop coming out of the display

Since a single 60 Hz frame is 16.6 ms and latency in a typical game is 35 ms or more – often much more – it will be challenging to get to 20 ms or less.

The really hard parts

- Tracking
- Latency
- Producing perceptions indistinguishable from reality

Tracking and latency are just prerequisites.

Once you have good enough tracking and latency, you can draw in the right place at the right time; then you learn about all the other interactions of displays with the human perceptual system.

The key here is that the way displays present photons to the eyes is nothing like the real world, and it's a miracle we can see anything coherent in displayed images at all.

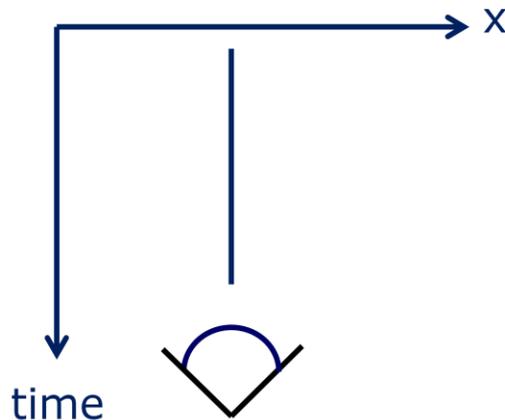
In the real world, objects emit or reflect photons continuously.

On a display, pixels emit fixed streams of photons for discrete periods of time.

Pixels are also fixed in space relative to the head.

This has major implications; let's look at a couple of them.

Space-time diagrams



For the following discussion, it will be useful to understand some very simple space-time diagrams, like the one shown here.

The horizontal axis is x position relative to the eyes, and the vertical axis is time, advancing down the slide.

You can think of these diagrams as showing how an object or an image would move horizontally across your field of view as time passes.

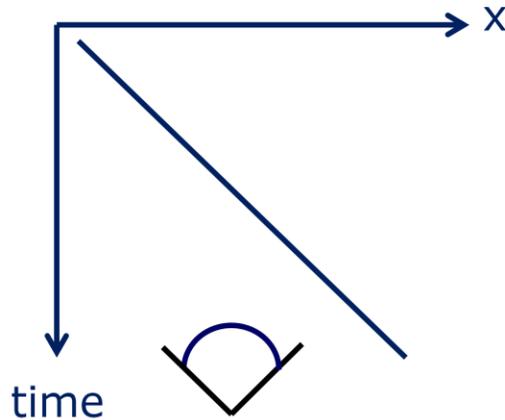
You can also find a discussion of space-time diagrams on my blog; I'll give the URL at the end of the talk.

In this diagram we have an object that is not moving relative to the eyes. The plot is a vertical line because there's no change in x position over time.

It's important to understand that the x position in these diagrams is relative to the position and orientation of the eyes, rather than the real world, because the eyes' frame of reference is what matters in terms of perception. So this diagram could be a case where both the eyes and the object are unmoving, or it could be a case where the eyes are smoothly tracking an object as it moves. In either case, the

object would remain in the same x position relative to the eyes.

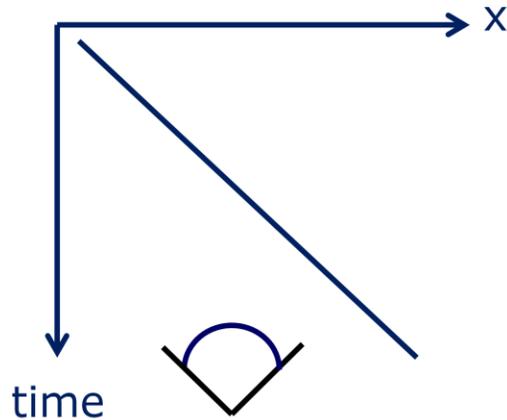
Spatial movement over time



This diagram shows a real-world object that's moving from left to right at a constant speed relative to the viewer, while the eyes remain fixated straight ahead – that is, the eyes aren't tracking the moving object. Here's an example of the sort of movement this diagram depicts.



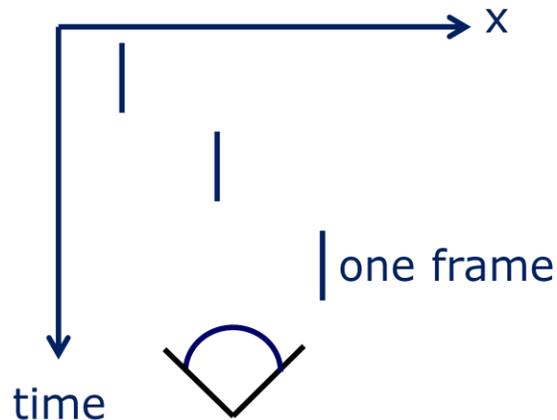
Spatial movement over time



Here's the diagram of that movement over time again; take a moment to absorb this, because I'll use several more of these diagrams.

These can be trickier to interpret than you'd expect from something so simple, especially if the eyes are moving, as we'll see later.

Pixel-based movement



Now let's look at the output of a display.

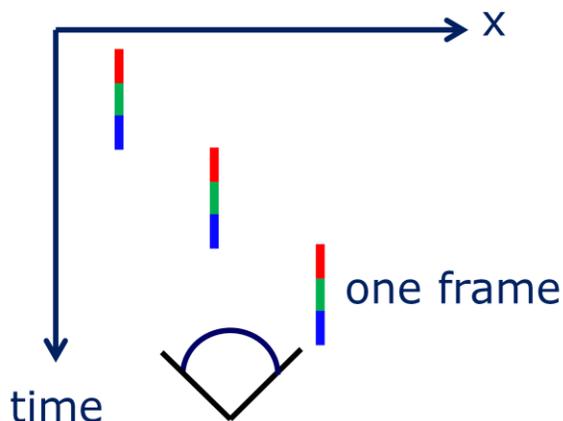
Here, a virtual point is moving across the screen while the eyes again remain fixated straight ahead. The point is illuminated at a single location on the display – the same pixel – for a full frame, because pixels only update once a frame. So instead of a smooth x displacement over time, we get stepped motion as the pixel is redrawn each frame.

Here's an example of that stepped motion.



This discrete nature of photon emission over time – temporal sampling – is key to the challenges posed by head mounted displays.

Sequential RGB display

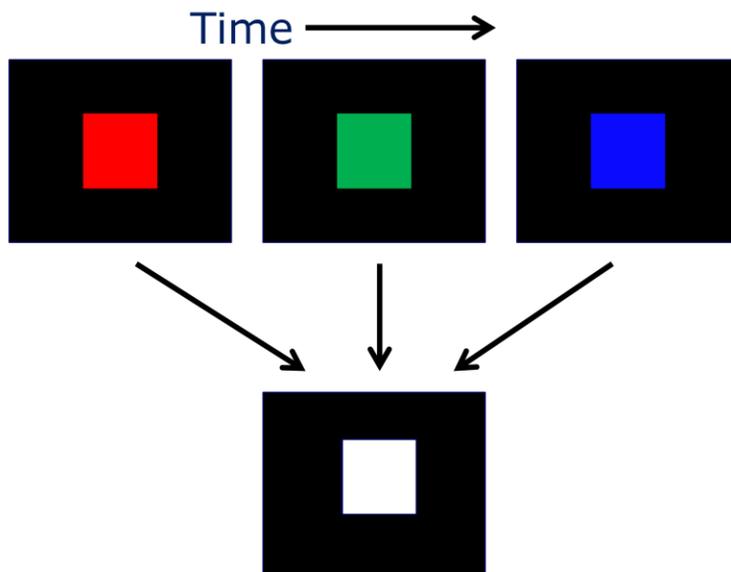


One good illustration of this is color fringing.

Color-sequential liquid crystal on silicon, or LCOS, projectors display red, green, and blue separately, one after another.

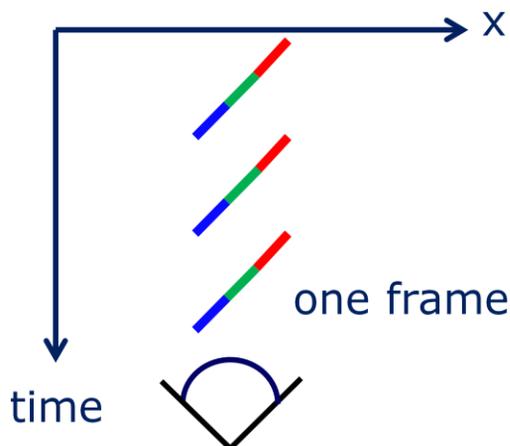
This diagram shows how the red, green, and blue components of a moving white virtual object are displayed over time, again with the eyes fixated straight ahead.

For a given pixel, each color displays for one-third of a frame; because the full cycle is displayed in 16 ms, the eyes blend the colors for that point together into a single composite color.



The result is that you see an image with the color properly blended, like this. Here the three color planes are displayed separately, one after another, and the three colored squares line up on top of each other to produce a white square. The red component doesn't actually stay illuminated while the green and blue components display, and likewise for blue and green; this is just to convey the general idea of sequential display of color components fusing to produce the final color.

Sequential RGB with eyes moving

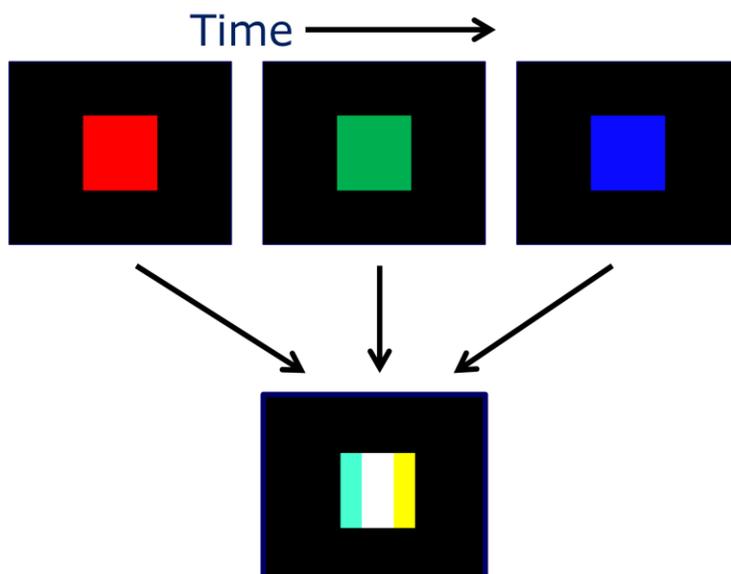


Now, what happens if your eyes are moving relative to the display, for example if you're tracking a moving virtual object from left to right?

The color components of a given pixel will each line up differently with the eyes, as you can see here, and color fringes will appear.

Remember, these diagrams are relative to the position and orientation of the eyes, not the real world.

There's actually another important implication of this diagram, which I'll talk about shortly.



Here's how the color fringing would look – color fringes appear at the left and right sides of the image, due to the movement of the eyes relative to the display between the times red, green, and blue are shown.

You might ask how visible the fringes can really be when a whole frame takes only 16.6 ms.

Well, if you turn your head at a leisurely speed, that's about 100 degrees/second, believe it or not; you can easily turn at several hundred degrees/second.

At 120 degrees/second, 1 frame is 2 degrees.

That doesn't sound like a lot, but two degrees can easily be dozens of pixels and that's very noticeable.

So VR displays need to illuminate all three color components simultaneously, or at least nearly so.

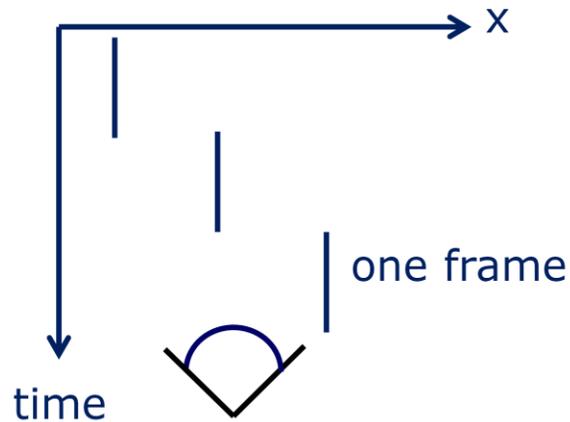
Now that we understand a bit about the temporal sampling done by displays, we come to persistence – that is, how long each pixel remains lit during a frame.

If you understand why color fringing occurs, you already know

everything you need to understand why persistence itself is a problem.

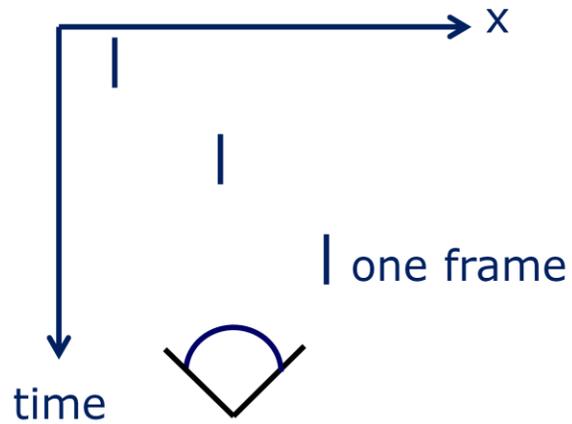
Persistence ranges between 0 ms and an entire frame time (or more!) for various display types.

Full persistence



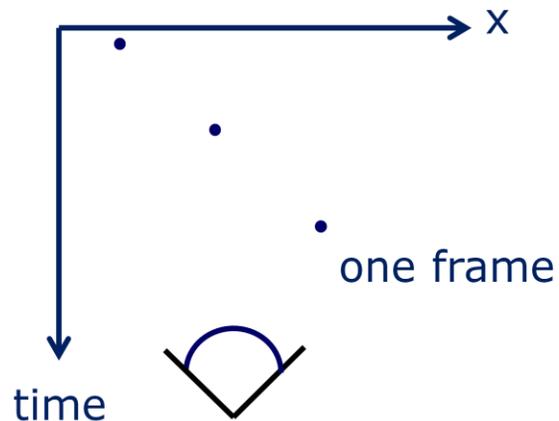
Remember this diagram? This is full persistence – the pixels remain lit throughout the frame.

Half persistence



This is half persistence, where pixels remain lit for half the frame.

Zero persistence

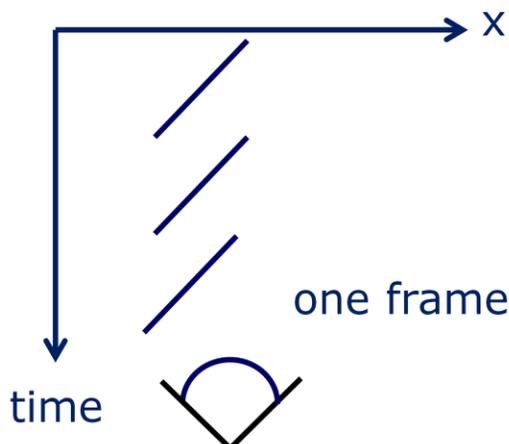


And this is zero persistence, where pixels are lit for only a tiny fraction of the frame – but with very high intensity to compensate for the short duration.

Both OLEDs and LCDs can be full persistence or less.

Scanning lasers are effectively zero persistence.

Full persistence + head rotation



Now we come to the crux of the matter.

High persistence, together with relatively low frame rate, causes what cinematographers call "judder", a mix of smearing and strobing. This diagram shows why.

Here we see the case where the eyes track a virtual object that's moving across the display. This could involve tracking a virtual object that appears to be moving through space, or it could involve turning the head – to which the display is attached – while fixating on a virtual object that isn't moving relative to the real world. The second case is particularly important for two reasons: one reason is that we tend to turn to look at new things by moving our eyes first, then fixating on the new target while the head rotates to catch up, and the second reason is that the relative speed between the display and the eyes can be an order of magnitude faster when the head rotates than when tracking a moving object without the head turning, with correspondingly larger artifacts.

Ideally, the virtual object would stay in exactly the same position relative to the eyes as the eyes move. However, the display only updates once a frame, so, as this diagram shows, with full persistence the virtual object slides away from the

correct location for the duration of a frame as the eyes move relative to the display, snaps back to the right location at the start of the next frame, and then starts to slide away again.



We can see that effect in this video, which was shot through a see-through head-mounted display with a high-speed camera and is being played back at one-tenth speed.

Here, the camera is panning across a wall that contains several markers used for optical tracking, and a virtual image is superimposed on each marker – the real-world markers are dimly visible as patterns of black-and-white squares through the virtual images.

It's easy to see that because the pixels are only updated once per displayed frame, they slide relative to the markers for a full displayed frame time (about 10 camera frames), then jump back to the correct position. Exactly the same thing happens with a full-persistence head-mounted display when you turn your head while fixating or when you track a moving virtual object.

Because this is being played back in slow motion, we can see the images clearly as they move during the course of each displayed frame. At real-world speeds, though, the pixel movement is fast enough to smear across the retina, which makes the image blurry.

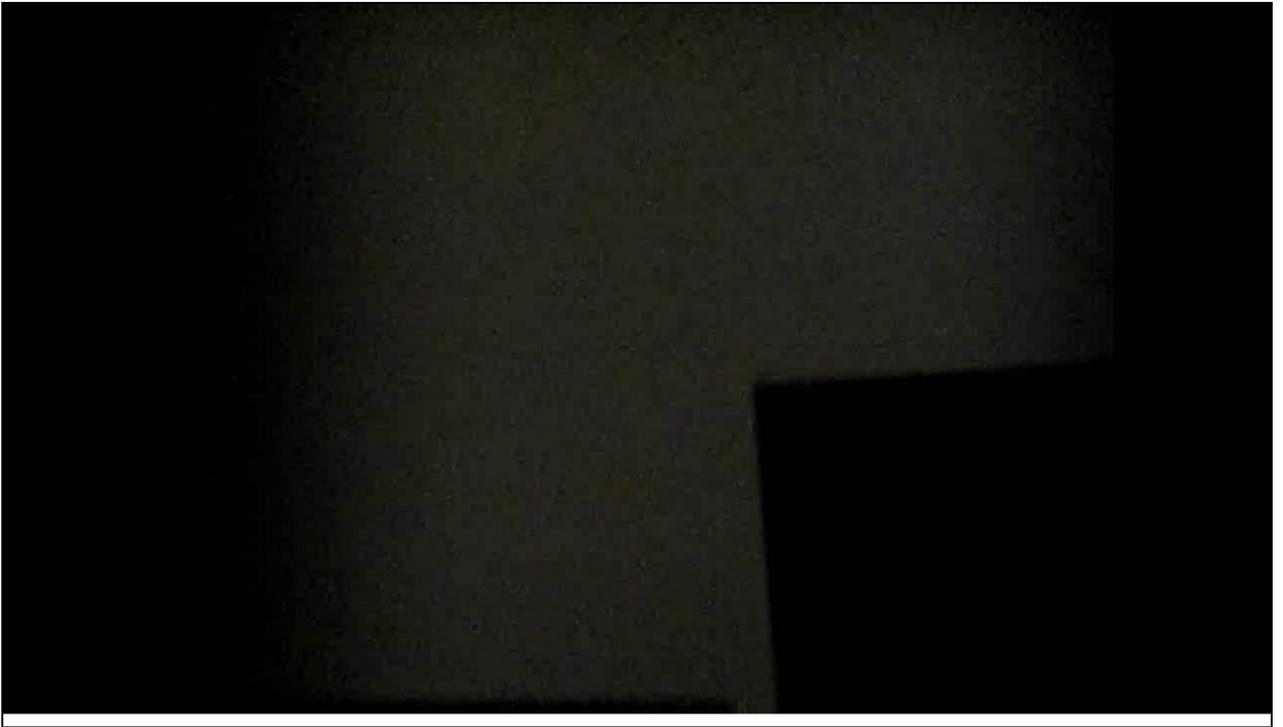


To give you an idea of what the smear part of judder looks like, here's a simulation of it. The image on the left is with the head not moving, and the image on the right is with a leisurely 120 degrees per second head turn rate. On a 60 hz full-persistence display, that results in two degrees of smearing across the retina per frame – and on a head mounted display that's 1280 pixels and 40 degrees wide, that's a full 64 pixels of smear, which as you can see reduces detail considerably.

Also, at real-world speeds the jumping back to the correct position at the start of each displayed frame makes images strobe – that is, it causes the eyes to see multiple simultaneous copies of each image, because they can't fuse detailed images that move more than about five or ten arc-minutes between frames.

The net result of the smearing and strobing is a loss of detail and smoothness, looking much like motion blur, whenever the eyes move relative to the display. You might think that wouldn't matter all that much because your eyes are moving, but again, if you fixate on an object and turn your head, you

can see perfectly clearly even though your eyes are moving rapidly relative to the display, and judder will be immediately noticeable.

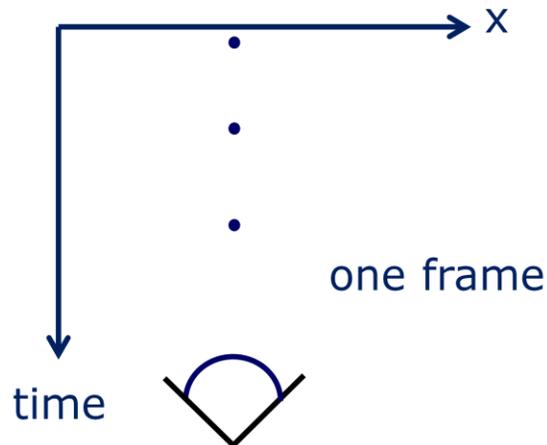


In contrast, observe how smooth the panning across text on a monitor is here.

This nicely illustrates how head mounted displays introduce a new set of unique artifacts.

The ideal way to eliminate judder is by having a high enough frame rate so that the eyes can't tell the difference from the real world; 1,000-2,000 frames per second would probably do the trick.

Zero persistence, tracked



Since that's not a feasible option, judder can also be eliminated with zero persistence, as shown in this diagram, where there's no sliding or jumping at all.

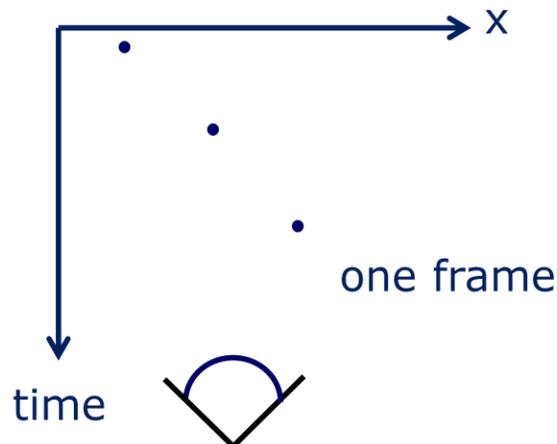
We've done the experiment of using a zero-persistence scanning laser display with really good tracking.

The result looks amazingly real; doing an A/B comparison with a full-persistence display is night and day.

So it would seem that zero persistence is the magic solution for judder – but it turns out to be just another layer of the perceptual onion.

Zero persistence works perfectly for whatever image the eyes are tracking, because that image lands in exactly the same place on the retina from frame to frame.

Zero persistence, untracked



However, as shown in this diagram, anything that is moving rapidly relative to the eyes now strobesc, because successive frames of such images fall too far apart on the retina to fuse, and there's no smear to hide that.

This is a great example of how deep the issues associated with head mounted displays go.

So at this point, judder in head mounted displays is not a solved problem.

Is judder really that bad?

- Judder doesn't make VR unusable, but...
- Great VR visual quality is going to require a lot of time and R&D

Judder isn't that bad; it does reduce visual quality and makes text all but unreadable, and is worse with higher pixel density, but the resulting virtual images are certainly good enough for gaming.

However, it's not great, either, and the point here is that, much as was the case with 3D graphics, great VR visual quality is going to require a considerable amount of time and R&D.

And believe me, there's a lot more than just color fringing and judder to figure out.

Other perceptual factors

- Sampled motion interaction with:
 - Vestibulo-ocular reflex (VOR) gain
 - Ocular following response (OFR)
 - Smooth pursuit
- Effects of raster-scan (rolling) displays

This slide shows additional ways in which the sampled nature of head-mounted displays may not produce the same interaction with the visual system that the real world does.

Other perceptual factors

- Vergence/accomodation conflicts
- Cycloverision
- Cyclovergence and shear in retinal correspondence
- Peripheral flicker
- Perception during saccades

And here are yet more perceptual factors.

So is that, finally, the end of the list of challenges? Hardly.

The really really hard problems

- Per-pixel focus
- Haptics
- Input
- How to be situationally aware of RR
- Solving VR motion sickness
- Figuring out what's uniquely fun about VR

After all that, we come to the really really hard problems.

Solving these is probably not required for VR to be successful, but it probably is required for VR to be great.

I'm not going to be able to talk about most of these today, but the single most important problem to be solved is figuring out what the compelling experiences are that can only be had in VR.

It'll be obvious in retrospect what those experiences are, but it's never obvious early on with new technology. During development, Quake had only static lighting by way of baked-in lightmaps, and John was adamant that dynamic lighting was not necessary. At GDC, however – the same GDC where I talked about Quake – Billy Zelznack gave us a demo of an engine he was building, a demo that happened to include a rocket that cast a dynamic light. When we got back to Dallas, John asked me if I wanted to take a shot at implementing dynamic lights for rockets, and when I said I had something else to finish up first, he said he bet he could do it in one hour. It actually took him only a few minutes more than that – and it was amazing how much more real the world seemed with dynamic lighting. There will be many, many such things with

VR, and they're all waiting for someone to find them.

Really hard AR-specific problems

- Wide-field-of-view see-through display
- Input
- Per-pixel opaquing
- Mobility
- Room-scale tracking
- Walk-around tracking

And here are some of the really hard AR problems, which unfortunately I don't have time to discuss either.

Possible roadmaps

- VR is likely to be straightforward



So how might this new world of gaming come to pass?

The roadmap for VR is likely to be pretty straightforward.

First, the Rift ships and is reasonably successful, and that kicks off the kind of positive spiral that occurred with 3D accelerators, with several companies competing and constantly improving the technology.

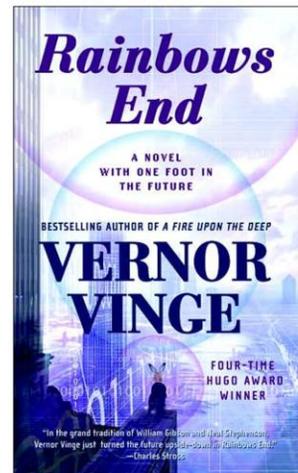
The real question is whether VR is a niche or a whole new platform – how broadly important it turns out to be.

In the long run, it's certainly potentially a new platform; just watch any "Star Trek" episode that has the Holodeck, or read *Ready Player One*.

It's just not clear how long it'll be until we can do most of that.

Possible roadmaps

- AR Roadmap #1
 - Information glasses succeed
 - Add AR as differentiation
- AR Roadmap #2
 - Console AR revolutionizes living-room gaming



Cover art: Stephan Martiniere
Published by Tor Books

The AR roadmap is less clear.

We know where we'd like to end up – with seamless, go-anywhere, always-on AR like *Rainbows End* – but that's much too far from current technology to even think about right now. We need to start by developing more tractable sorts of AR.

Possible roadmap #1 involves the success and widespread use of HUD-style head mounted displays like Google Glass, which show information, but don't do AR.

Once they're established, though, AR could become a value-added, differentiating feature - for example, allowing you to play virtual tabletop games at home, in airports, or in boring meetings.

In possible roadmap #2, living-room AR successfully ships for a console and becomes the dominant form of living-room gaming.

In either case, once AR has a toehold, it can start to evolve toward science fiction AR.

That's definitely going to take decades to fully refine.

Exciting times!

- There's a huge amount to learn, but we're already headed down that path
 - Next, Joe Ludwig will talk about some of what we've learned porting *TF2* to VR
- <http://blogs.valvesoftware.com/abrash/>
- Google "abrash black book" (free online)

I've just spent 25 minutes telling you how hard VR is – and that's certainly true. But realtime 3D was equally hard – just check out Chapter 64 in my Black Book about the lengths John went to in order to solve the potentially visible set problem, or think about how crude the early 3D accelerators were – and over time all that has worked out amazingly well.

This is the kind of opportunity that everyone in the gaming industry should dream of; if you want to do challenging work that has the potential to affect almost every game written in five or ten years, VR is a great place to be right now.

It really is like when I was working on *Quake* - a new world is emerging.

My guess is that the next few years will see more change in the gaming experience than the seventeen since I gave my talk about *Quake*. I could be wrong – but I hope I'm not, and I'm excited to find out!

Next, Joe Ludwig will talk about some of what we've learned porting *TF2* to virtual reality – and after Joe's talk, both Joe and I will do Q&A.