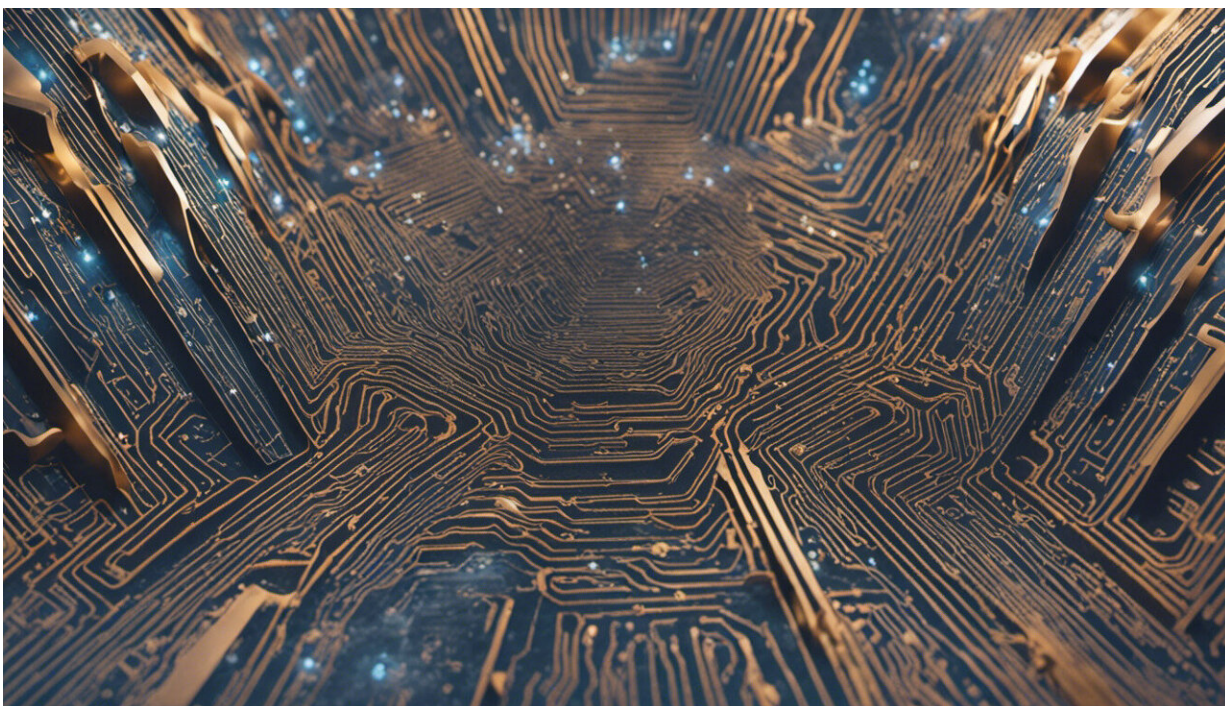


How does a computer know where you're looking?

August 31 2016, by Ann Mcnamara



Credit: AI-generated image ([disclaimer](#))

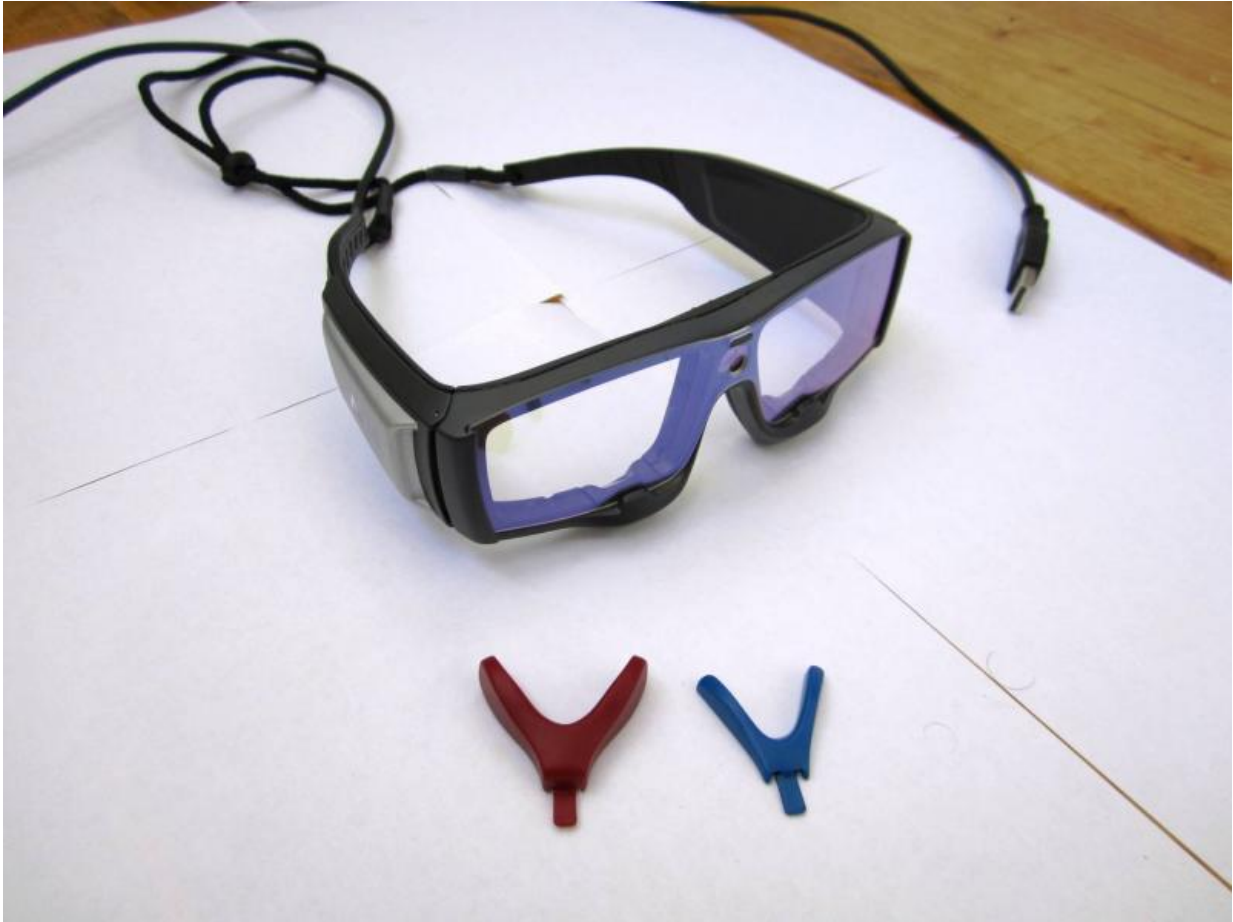
Imagine driving a car, using a heads-up display projection on the windshield to navigate through an unfamiliar city. This is augmented reality (AR); the information is used to not only guide you along a route, but also to alert you to salient information in your surroundings, such as cyclists or pedestrians. The correct placement of virtual content is not

only crucial, but perhaps a matter of life and death.

Information can't obscure other material, and should be displayed long enough for you to understand it, but not too much longer than that. Computer systems have to make these determinations in real-time, without causing any of the [information](#) to be distracting or obtrusive. We certainly don't want a warning about a cyclist about to cross in front of the car to obscure the cyclist herself!

As a researcher in AR, I spend a lot of time trying to figure out how to get the right information onto a user's screen, in just the right place, at just the right moment. I've learned that showing too much information can confuse the user, but not showing enough can render an application useless. We have to find the sweet spot in between.

A crucial element of this, it turns out, is knowing where users are looking. Only then can we deliver the information they want in a location where they can process it. Our research involves measuring where the [user is looking in the real scene](#), as a way to help decide where to place virtual content. With AR poised to infiltrate many areas of our lives – from [driving](#) to [work](#) to [recreation](#) – we'll need to solve this problem before we can rely on AR to provide support for serious or critical actions.



Eye-tracking spectacles can be relatively compact. Credit: Anatolich1, CC BY-SA

Determining where to put information

It makes sense to have information appear where the user is looking. When navigating, a user could look at a building, street or other real object to reveal the associated virtual information; the system would know to hide all other displays to avoid cluttering the visible scene.

But how do we know what someone is looking at? It turns out that the nuances of human vision allow us to examine at a person's eyes and

[calculate where they are looking](#). By pairing those data with cameras showing a person's field of view, we can determine what the person is seeing and what he or she is looking at.

[Eye-tracking systems first emerged in the 1900s](#). Originally they were mostly used to study reading patterns; some could be very intrusive for the reader. More recently, real-time eye-tracking has emerged and become more affordable, easier to operate and smaller.

Eye trackers can be attached to the [screen or integrated into wearable glasses or head-mounted displays](#). Eyes are tracked using a combination of the [cameras, projections and computer vision algorithms](#) to calculate the position of the eye and the gaze point on a monitor.

We generally look at two measures when examining eye tracking data. The first is called a [fixation](#), and is used to describe when we pause our gaze, often on an interesting location in a scene because it has caught our attention. The second is a [saccade](#), one of the [rapid eye movements](#) used to position the gaze. Basically, our eyes quickly dart from place to place taking in pieces of information about parts of a scene. Our brains then put the information from these fixations together to form a visual image in our minds.

Combining eye tracking with AR

Often AR content is anchored to a real-world object or location. For example, a virtual label containing a street name should be displayed on that street. Ideally, we would like the AR labels to appear close to the real object it is associated with. But we also need to be careful not to let multiple AR labels overlap and become unreadable. There are many approaches to managing label placement. We're exploring one option: calculating where the person is looking in the real scene and displaying AR labels only in that spot.

Say, for example, a user is interacting with a mobile application that helps him shop for low-calorie cereal in the grocery store. In the AR application, each cereal has [calorie information](#) associated with it. Rather than physically picking up each cereal box and reading the nutritional content, the user can hold up his mobile device and point it at a particular cereal box to reveal the relevant information.

But think about how crowded a store's cereal aisle is with various packages. Without some way to manage the display of AR labels, the calorie information labels for all the cereal boxes would be displayed. It would be impossible to identify the calorie content for the cereal he is interested in.

By tracking his eyes, we can determine which individual cereal box the user is looking at. Then we display the calorie information for that particular cereal. When he shifts his gaze to another box, we display the figures for the next one he considers. His screen is uncluttered, the information he wants is readily available and when he needs additional information, we can display that.

This type of development makes it an exciting time for AR research. Our ability to integrate real-world scenes with computer graphics on mobile displays is improving. This fuels the prospect of creating stunning new applications that expand our ability to interact with, learn from and be entertained by the world around us.

This article was originally published on [The Conversation](#). Read the [original article](#).

Provided by The Conversation

Citation: How does a computer know where you're looking? (2016, August 31) retrieved 23

February 2024 from <https://techxplore.com/news/2016-08-youre.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.