CS 280 Computer Vision
Spring 2018 Yang Lecture 26

1 Lecture

To create an immersive experience in 3D space, we need a lot of components to work together. The main challenge in getting computer vision to work for AR/VR is *motion-to-photon latency.* As we move, the camera in an AR system has to track our movement with a certain frequency, and we'll need a consistent augmentation with respect to the world.

Ideally, end-to-end/round-trip latency should be no more than 12 ms (i.e. at least 90 fps). And this is for the entire loop – including perception and display. (This problem is not solved!)

3D Geometry Refresher

We will assume that objects are rigid bodies (not articulated or deformable). All points move in the same way; distances within the object are fixed. We can easily change between the object and world coordinate systems based on the rotation and translation between the two. *A note: be careful about left-handed versus right-handed coordinate systems.*

In computer vision, we look at images, i.e. 2D projections of a 3D world. We lose information:

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = \underbrace{\begin{bmatrix}f_x & 0 & c_x\\0 & f_y & c_y\\0 & 0 & 1\end{bmatrix}}_{\text{intrinsics}}\underbrace{\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_1\\r_{21} & r_{22} & r_{23} & t_2\\r_{31} & r_{32} & r_{33} & t_3\end{bmatrix}}_{\text{extrinsics}}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

where (u, v) are the projected coordinates, (X, Y, Z) are the 3D world coordinates, (f_x, f_y) are focal lengths and describe scale, and (c_x, c_y) are the origin coordinates (usually at the image center).

By the pinhole model, any point along a 3D ray can generate a given image point. If we only have a single image, we lose depth information. In SfM algorithms, we add more views and compute 3D point locations from the aggregated data. (The SfM process gives us back the depth information.)

SfM requires point correspondences in images. How do we find correspondences? We need similarities. If we look at a point from *this* angle and *that* angle, it should have some kind of invariance. The “most invariant” points are corner points. For bonus points, we can also use SIFT descriptors.

AR/VR

In the past, we have communicated over 2D mediums: a screen, a piece of paper, etc. The attraction of VR is that it gives us a way to communicate in 3D: our canvas is now not a screen, but a space. It is a 3D world, everything around us. Think *Ready Player One.*

The first VR prototype, Sensorama, was developed in the 1950s. It allowed users to drive a motorcycle in an immersive fashion – with a vibrating chair and handles, visual input, and even a smell generator.

In 1968, Sutherland created the Sword of Damocles – the first wearable AR prototype. (In AR, we see not only the augmented information, but the world as well.)

Most of the major problems constraining AR/VR are targeted, ideally, to be solved around 2020. It's time to start thinking about a post-AR/VR world.

Challenges

There are a few main issues that make it difficult to achieve 12 ms latency. (We're on a timer: every 12 ms, we need to make a decision. Otherwise we'll lag, and our rendering will drift.)

Near-Eye Displays

When anything gives us information, it starts with the display.

- In the previous generation, most displays were **LCDs** (*pros*: cheap, bright, no burn-in effect; *cons*: render-to-display latency in excess of 5 ms, low pixel density, low contrast, narrow viewing angle).
- Today, most smartphones use **AMOLED** displays (*pros*: higher contrast, latency less than 1 ms; *cons*: burn-in effect, deterioration in quality over time, screen-door effect).
- Currently, the solution for a good visual experience is the **micro OLED** display (*pros*: extremely low sub-1 μs latency, high pixel density, no screen-door effect; *cons*: expensive, burn-in effect).

LCD technology is now a bottleneck and is bad for wearables.

OLED is fast because illumination of pixels depends on current, not voltage. We don't need to wait for crystals to align, as we do with LCDs.

Micro OLED builds (encapsulates) everything on a single silicon wafer.

Accommodation and Vergence

When an eye sees an object in 3D, it does two things: (1) it converges on the point by rotating (*vergence*), and (2) once it converges, the pupil adjusts to focus on that point, meaning anything not on that depth will be blurry (*accommodation*). If we want to create something virtual, we also have to satisfy these two cues: *blur* (the rendering machine had better display the correct blurriness) and *vergence* (we have two images, and when we show an object in space, we had better show it at the right distance in the images).

However, the objects are not in real 3D space. They are being displayed on a screen. Thus no matter now we change the vergence, the accommodation is always set to the screen. Our individual eyes always tell us that the focal plane (the clearest place) is at the screen. (If we're talking about a real object in space, the vergence and the accommodation converge at the same point, so we don't have this problem.)

This leads to discomfort – the accommodation-vergence conflict.

Solutions: near-eye light field displays, visual hallucination.

Depth Cameras

We can now measure depth from cameras (though not perfectly). This can avoid our having to do SfM.

OpenARK

Berkeley has a long-standing tradition of making open-source contributions to the community. We are currently developing **OpenARK**, an open-source solution to many AR problems.