CS 498 VR

Lecture 23 - 4/25/18

go.illinois.edu/VRlect23
Review from last lecture

1. What is vection?

2. What are some symptoms when you experience vestibular-ocular mismatch?
Tracking Systems in VR

What do we want to track? (Think of rigid bodies)

- HMD
- Hands
- Eyes
Tracking Systems in VR

What do we want to track? (Think of rigid bodies)

- Head wearing HMD
- Eyes
- Palms of hands
- Fingers
- Entire body
- Interactable objects - controller, coffee cup, desk...
- Other people in the space
Tracking Systems in VR

What do we want to track? **Rigid Bodies**

For each body, estimate:

- **Rotation**: $3 \times 3$ rotation matrix, $R$  
- **Position**: $(x, y, z)$

Equivalently,

Homogeneous Transformation Matrix ($H_i$)
Tracking Systems in VR: Estimating 3D Orientation

Axis-Angle:

3-axis gyroscope measures:

For every $\Delta t$, measure $\omega_i$, and your new rotation is $\theta_i = \omega_i \cdot \Delta t + \theta_{i-1}$

Issue: Drift (or dead reckoning)
Estimating 3D Orientation: Drift Correction

Defining drift error: Divergence of the estimate from the actual value.

Correcting drift error:

- Use another sensor (or sensors) to provide a global reference.
- Gradually apply corrections
  - Fast enough to Fast enough to eliminate drift
  - Slow enough to avoid VR sickness
Estimating 3D Orientation

Separate rotational drift into **two components**:

1) Tilt error:
   a) To correct: 

   - Complementary Filters on SO(3), Mahoney, 2008
   - Head Tracking for the Oculus Rift, ICRA 2014, S. LaValle, A. Yershova, M. Katsev, M. Antonov

2) Yaw error:
   a) To correct

   *need compass*
Use “Perfect Up” Sensor to Correct “Tilt Error”

If estimated Y-axis is not aligned with measured up-vector, apply transformation to estimate to correct error.
Use “Perfect Up” Sensor to Correct “Tilt Error”

Find difference between estimated “up” and measured “up”.

Gradually apply transformation. (Also known as complementary filter)

Profit.
Using Accelerometer as “Up” Sensor

What do accelerometers measure?

If device is in free fall?

\[ \|a\| = 0 \]
\[ (0,0,0) \]
Using Accelerometer as “Up” Sensor

What do accelerometers measure?

If device is in free fall? (hint: or in outer space)
Using Accelerometer as “Up” Sensor

What do accelerometers measure?

If device is in free fall? (hint: or in outer space)

Lying still on a table?

\[
a = \begin{pmatrix} 0, 9.81, 0 \end{pmatrix}
\]

\[
a = \begin{pmatrix} -9.81, 0, 0 \end{pmatrix}
\]

\[
a = \begin{pmatrix} 9.81 \cos \theta, 0, 9.81 \sin \theta \end{pmatrix}
\]
Use Accelerometer to Correct Tilt Error

Problem:
Accelerometer measures vector sum of linear acceleration of the sensor and gravity.

Solution:
Use heuristic to detect when "not moving" and apply correction only then. 
\[ \| \mathbf{a} \| \approx 9.81 \]
Use Magnetometer to Correct Yaw Error

Similar to tilt correction:

- Calculate reference error
- Gradually apply using complementary filter

Problems:

- Vector sum of Earth's magnetic field + local ambient structure & Headset, electronics, building
Estimating Position and Orientation

The Problem:

- Allow and track parallax motion (translations)
- IMU (accelerometer + gyroscope + magnetometer) is not enough
  - Drift errors too fast and no good way to detect
- Need: high accuracy and stability

Solutions:

- Generate own EM signal
- Visibility or line-of-sight methods
Position Tracking: Visibility Methods

Camera arrangements:

On headset

In world

inside-out

outside-in
Position Tracking: Visibility Methods

Pinhole camera:

Features in an image:
Position Tracking: Visibility Methods

1) Natural
   a) Extract and maintain from natural scenes
   b) Ignore moving objects
   c) Hard computer vision problems
   d) Low reliability

2) Artificial
   a) Use known fixed markers in environment
      i) QR codes, reflective tape, LEDs, lasers
         ii) Can stay in IR spectrum
   b) Trivial computer vision
   c) Requires prior knowledge or setup of environment
Position Tracking: Inside-Out Tracking
Position Tracking: Outside-In Tracking
Position Tracking: Blob Detection

PnP Problem:

Determine rigid body transformation from identified, observed features on a rigidbody.

P1P:

DOF Analysis:

- Start with 6 unknown DOFs (for a rigid body)
- Each feature subtracts 2 DOFs
Position Tracking: Blob Detection

P2P:

DOFs left:
Position Tracking: Blob Detection

P3P:

DOFs left: zero

Solution:
Review from today

- Detail some strategies for correcting errors in measuring a body’s orientation.

3

- How many reference points are required to know the position (but not necessarily orientation) of a rigidbody from a single camera’s perspective?
  - What else can you infer from these points?
Announcements

- Check Piazza for final project deadlines, they are coming up soon!
- Reading: Chapter 9 of the book