Positional tracking for VR/AR using Stereo Vision

Edwin AZZAM, CTO Stereolabs
Outside-in Positional Tracking

IR LED constellation tracking system – Oculus Rift

Lighthouse laser system – HTC Vive
Inside-out Positional Tracking

- Use camera on HMD to track user position in space
- Eliminates the need for external tracking hardware and allows freedom of movement
- Best for mobile VR/AR
- Many use cases outside the VR/AR industry!
Our approach to depth and motion tracking

- 2x RGB camera (rolling shutter)
- 1x RGB camera
- 1x Fisheye camera (global shutter)
- 1x IR projector
- 1x IR camera
Hard problems of stereo positional tracking

- Positional tracking with rolling shutter stereo is difficult
  - Stereo calibration issues
  - Depth reliability and computation
  - Stereo odometry computation
  - Position accuracy
  - Position jitter
  - Judder
  - Drift

Simulated jitter distribution using time series model
What is ZED SDK?

- Native C++ SDK provided with ZED camera
- Software pipeline for depth, tracking and mapping from stereo
ZED SDK pipeline modules

- Stereo Images
- Self-calibration
- Depth Estimation
- Visual Odometry
- Spatial Mapping
- Graphics Rendering

CPU

GPU
ZED SDK pipeline modules

- Pose information is output at the frame rate of the camera
- `sl::Pose` is used to store camera position, timestamp and confidence
  
  ```c++
  zed.getPosition(zed_pose, REFERENCE_FRAME_WORLD);
  ```
Mobile positional tracking demo workflow

<table>
<thead>
<tr>
<th>ZED</th>
<th>Gear VR</th>
<th>Galaxy S7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stereo Camera</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **USB 3.0**
  - Stereo images
  - 720p60

- **Wi-Fi**
  - 60Hz Pose 60Hz

- **ZED SDK**
  - Jetson TX1
  - (Auvidea J120)
Linq - Mixed-Reality demo
Most problems solved, however drift and judder still need appropriate handling
ZED SDK pipeline modules

➢ Most problems solved, however drift and judder still need appropriate handling
Mitigating drift with Spatial Memory

- Small pose errors accumulate slowly over time
- Spatial memory module learns and recognizes its surroundings to detect drift in absolute position
Drift correction strategies

- When a large error is detected, immediate drift correction will cause a jump in the absolute pose

- Need to apply corrections without the user noticing
  - Smooth error correction over multiple frames
  - Correction rate still needs to be higher than drift accumulation!

- Smoothing strategy does not work for AR
  - Misalignment between real and virtual too visible
  - Jumps preferable
Judder reasons

- Target tracking FPS not met
  - 90Hz for desktop, 60Hz for mobile, 120Hz for next-gen?

- FPS instability
  - 90fps = 11ms
  - 1ms compute variability costs 7 frames: 83fps
  - Traditional SLAM algorithms have +/- 30% variability
  - Additional factors: OS scheduling, CPU/GPU allocated resources, render vs compute...

- USB dropped frames
Mitigating judder with Pose prediction

- Pose prediction is different from timewarp
- Ensures a consistent rate of new position and orientation tracking values
Balancing vision vs rendering

- **Maxwell**: New GPU work can’t be started until compute and graphics queues are finished
- **Pascal**: Dynamic balancing fills up the remaining time with work from the other queues
- Thread and instruction level compute preemption can help
Plugins and interfaces

- Unity
- Unreal Engine (Q3 17)
- OpenCV
- ROS
- Oculus
- PCL
- MATLAB
- TensorFlow (Q3 17)
Contact

Edwin AZZAM, CTO
edwin.azzam@stereolabs.com

www.stereolabs.com
Who we are

- STEREOLABS provides **depth** and **motion sensing** technology based on stereo vision

- 30+ employees based in San Francisco and Paris

- Launched the **ZED** in 2015, the world’s first stereo camera for space and motion sensing

- Announced the **Linq** in 2017, the first mixed-reality HMD prototype with stereo vision