Using ROS and Gazebo to Safely Validate and Verify Autonomous Systems

Joshua P. Hecker

Staff Software Engineer

ROSCon 2019

LOCKHEED MARTIN

We never forget who we're working for®

ROS-related Projects @ LMAS

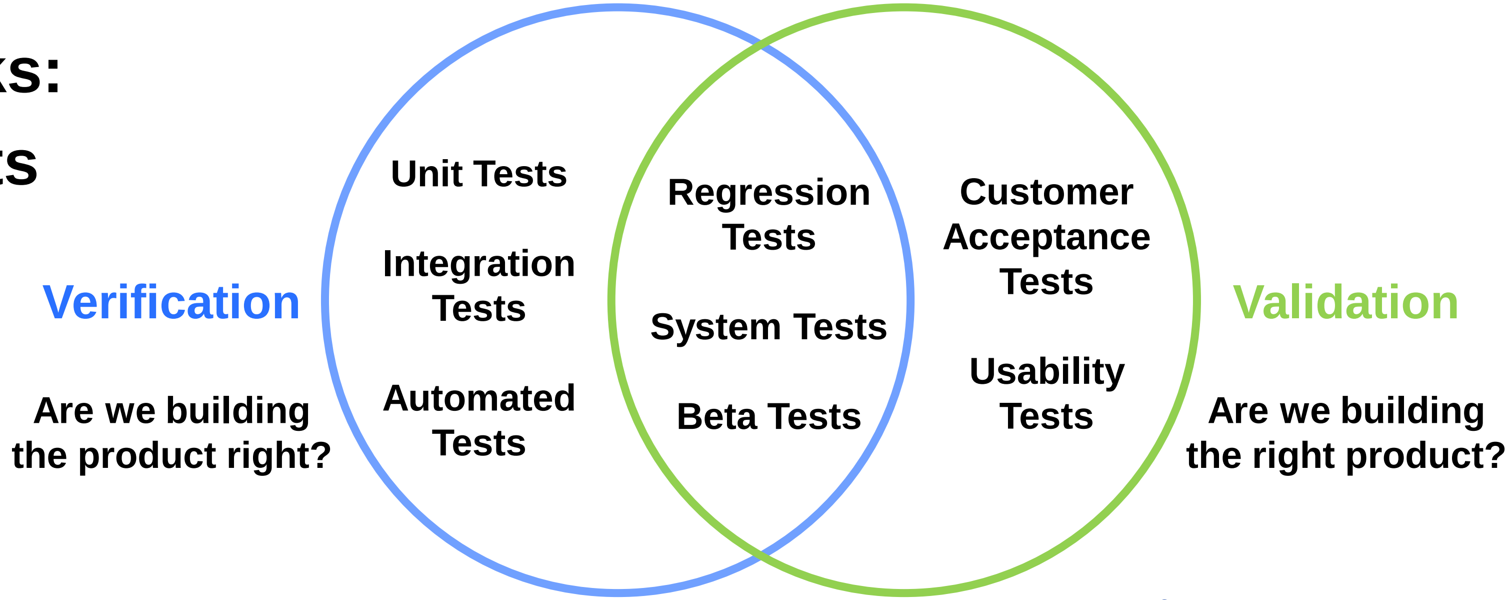
- **Multiple projects built on and/or extending ROS as a core autonomy architecture:**
  - **Transport and mining truck fleets**
  - **Subsurface vehicles**
  - **Multiple aerial platforms**
- **Significant investment in vehicle and sensor modeling in Gazebo:**
  - **Segway RMP 440 LE (off-road platform)**
  - **Caterpillar 777F (mining haul truck)**
  - **Neptec OPAL (3D lidar sensor)**
- **ROS/Gazebo commonly requested by customers, agencies, proposal calls, etc.**

Lockheed Martin Autonomous Systems (LMAS)

Littleton, CO, USA

Autonomy Validation and Verification

- **Validating and verifying autonomy software is essential to ensuring specification fulfillment, proper functionality, and safety criticality**
- **Performance of autonomous robots is notoriously difficult to guarantee:**
  - **How can we test all reasonable real-world scenarios?**
  - **Do tests "fail" or do they produce "new, unexpected behavior"?**
  - **Learning adds additional layers of complexity, time, and possible outcomes**
- **Pure simulation has drawbacks:**
  - **Computational requirements**
  - **Time vs. Quality vs. Cost**
  - **Reality gap**



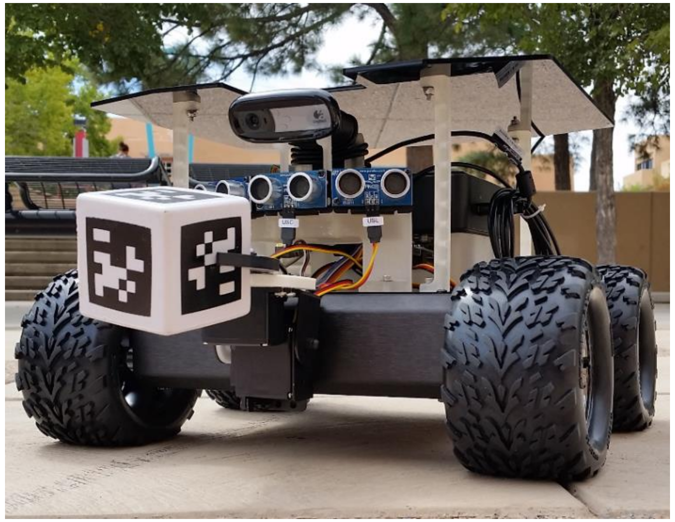
A Solution: Live/Virtual/Constructive Simulation

- Run ROS autonomy software on live asset
  - Minimize reality gap
- Represent live asset in Gazebo using virtual avatar
  - Minimize computational requirements (only model essentials)
  - Minimize development costs (only model essentials)
- Map live asset and virtual avatar into unified constructive environment
  - Requires accurate tracking of live asset
- Provides for rapid testing of real robots in challenging environments
- Simulates realistic collisions with minimal danger to physical systems

Example LVC Instantiation

Example LVC Instantiation

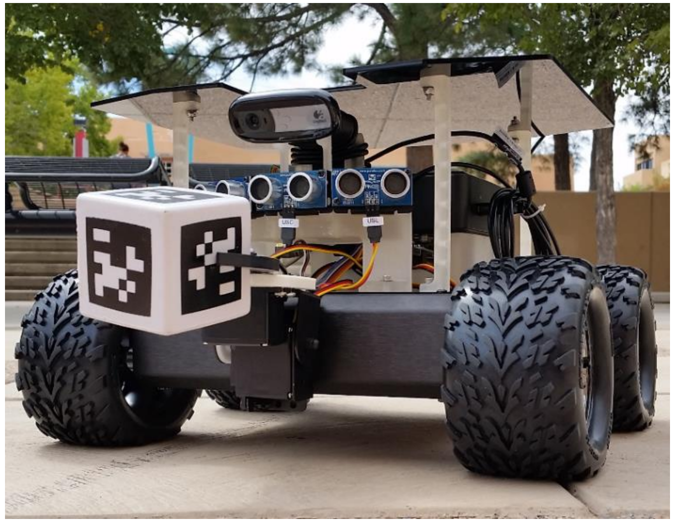
Physical Asset

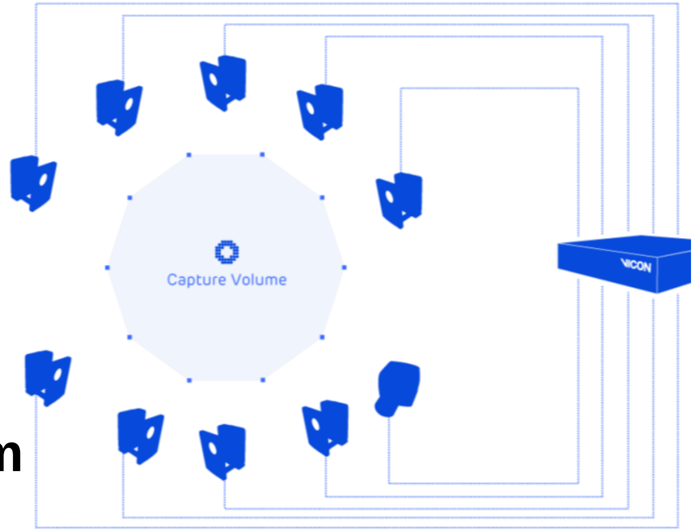


1. Swarmie robotic hardware platform (NASA/UNM/Swarming Technologies)

Example LVC Instantiation

Physical Asset



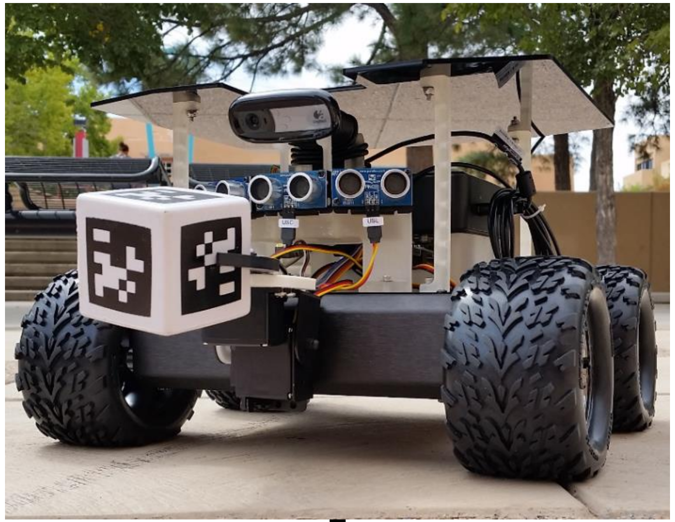


2. Vicon Vantage system
(24 V16 cameras)

Motion Capture

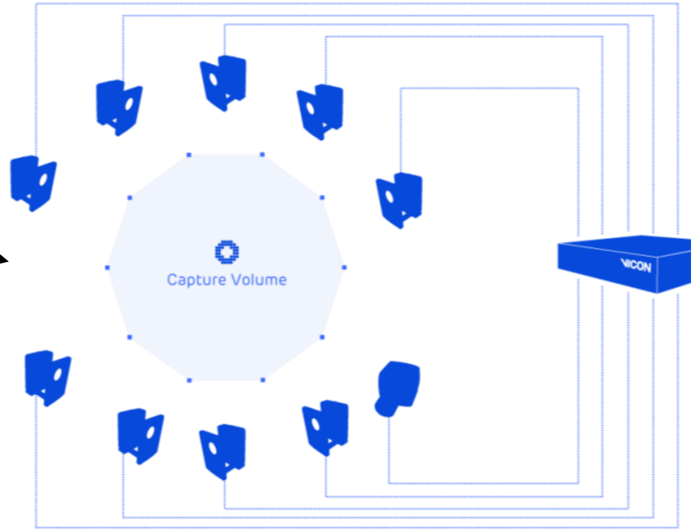
Example LVC Instantiation

Physical Asset



Movement

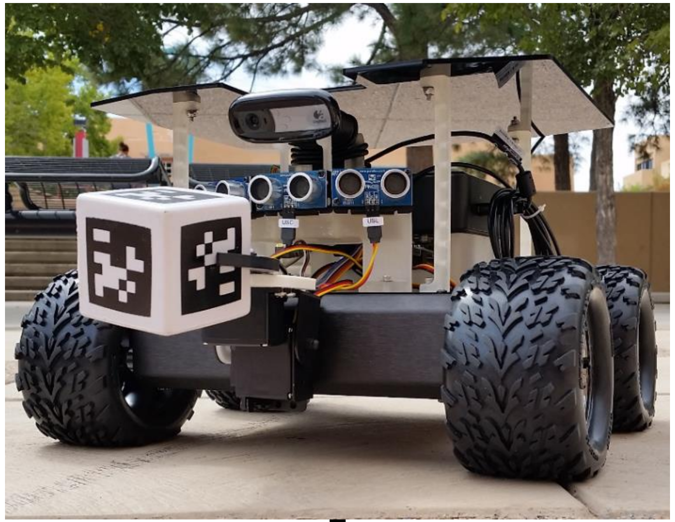
3. Infrared cameras detect passive markers



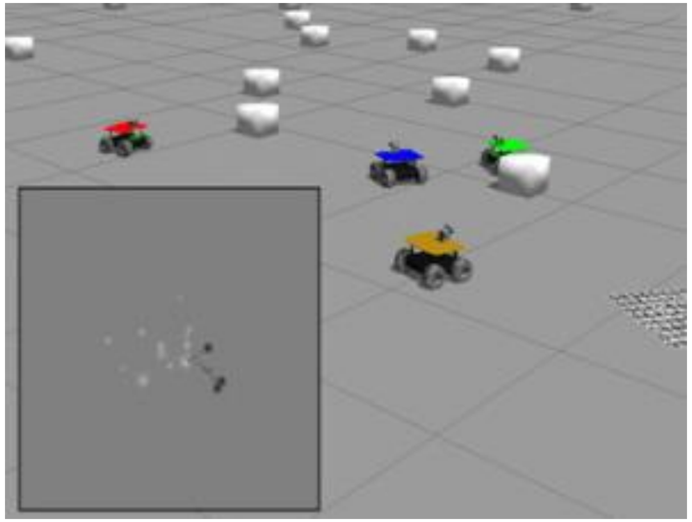
Motion Capture

Example LVC Instantiation

Physical Asset

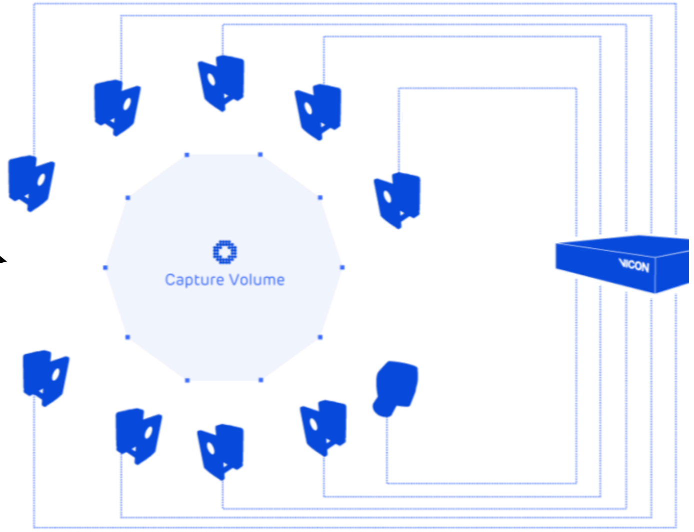


Virtual Avatar



4. Unactuated Gazebo model with required sensor simulators

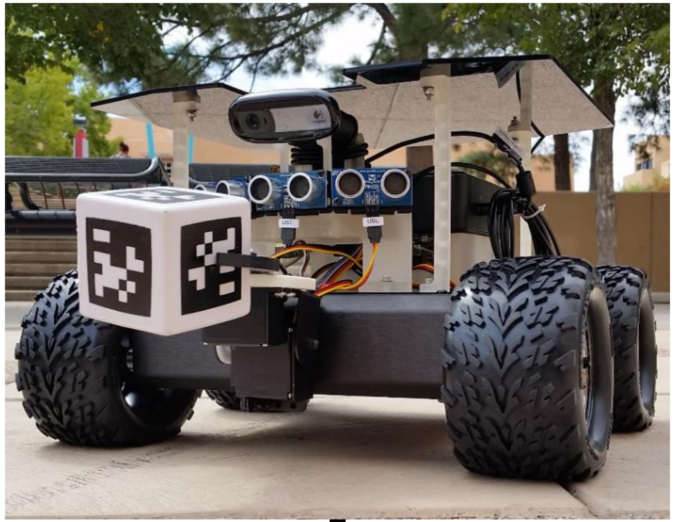
Movement



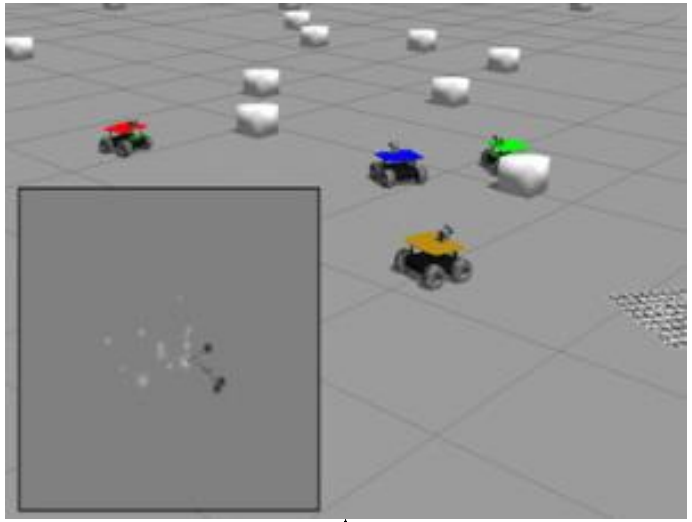
Motion Capture

Example LVC Instantiation

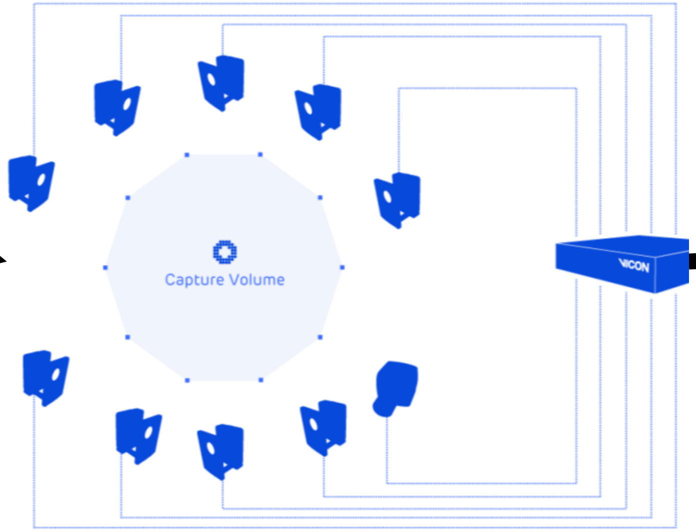
Physical Asset



Virtual Avatar



Movement



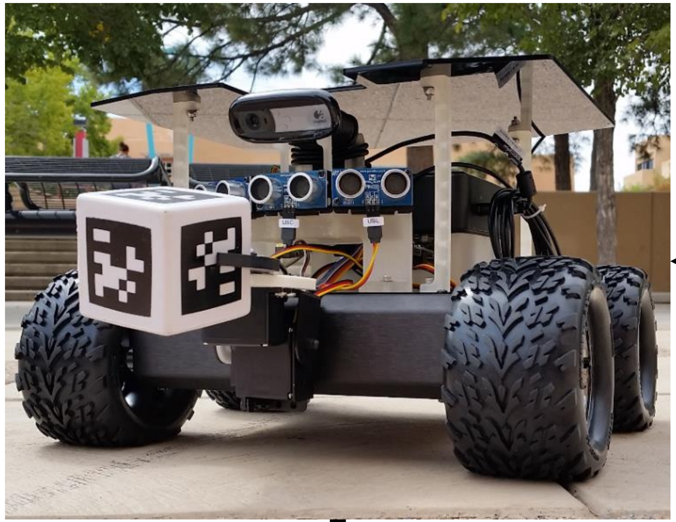
Position Data

5. Gazebo/ROS plugin maps real robot pose to avatar

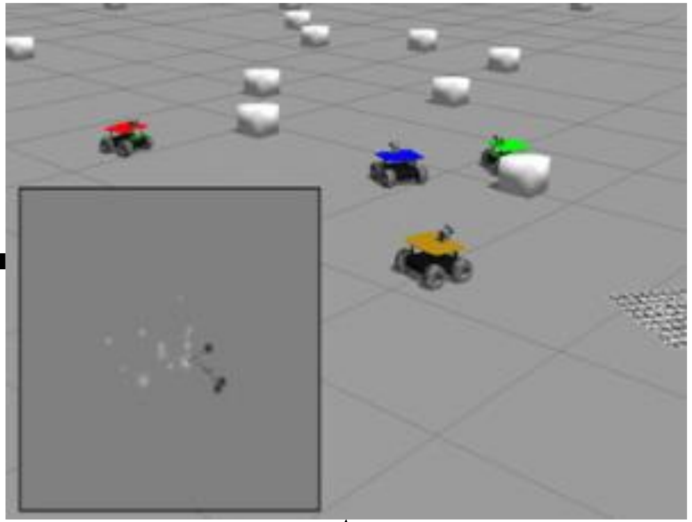
Motion Capture

Example LVC Instantiation

Physical Asset



Virtual Avatar

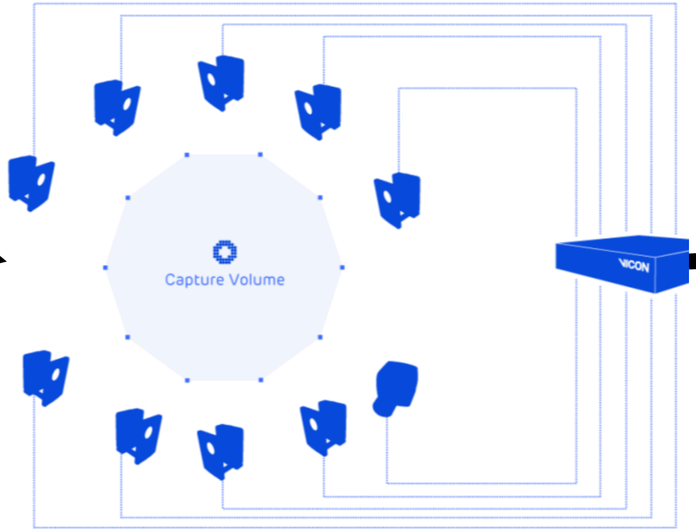


6. Real robot gets sensed Gazebo data

Sensor Data

Movement

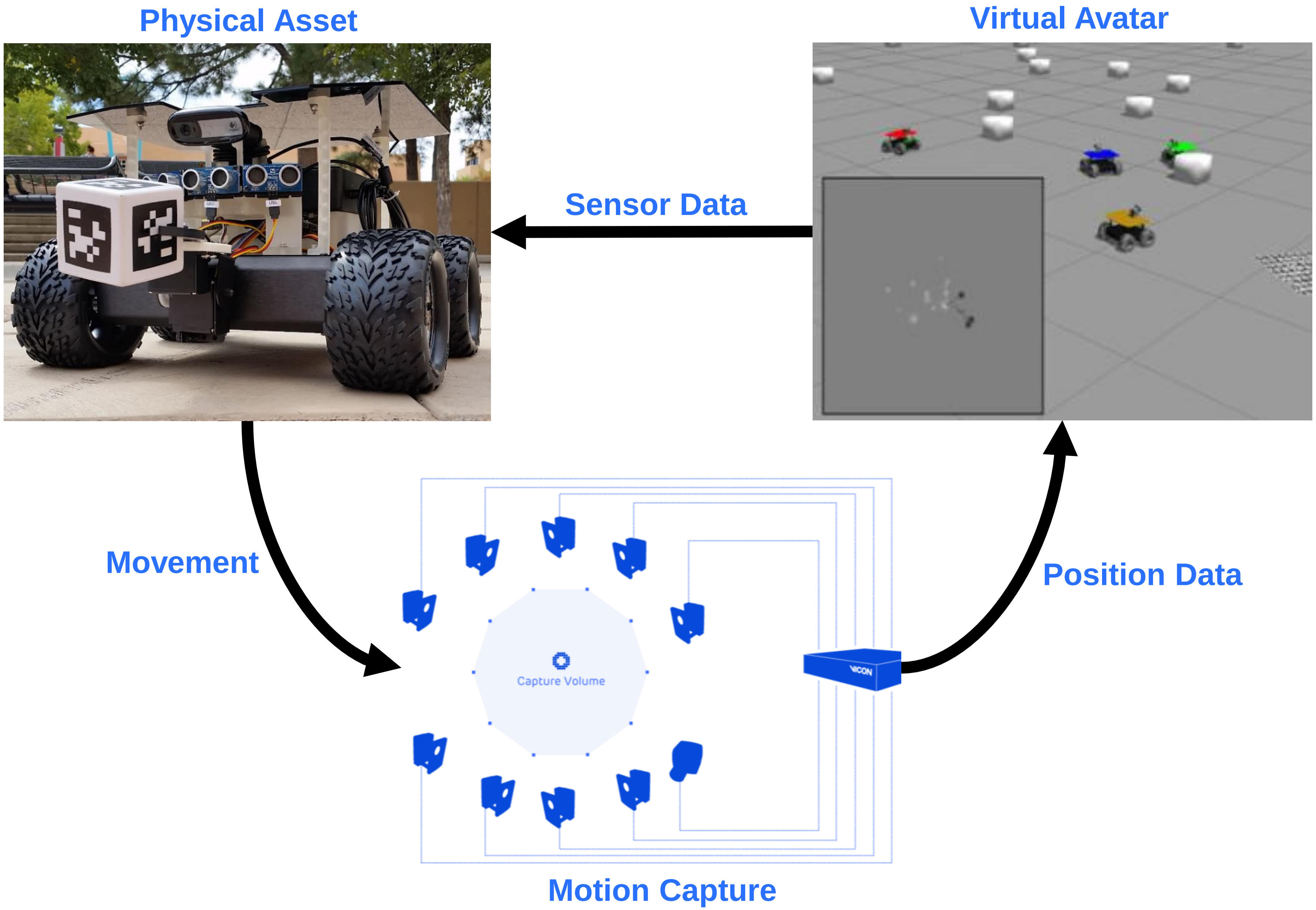
Position Data



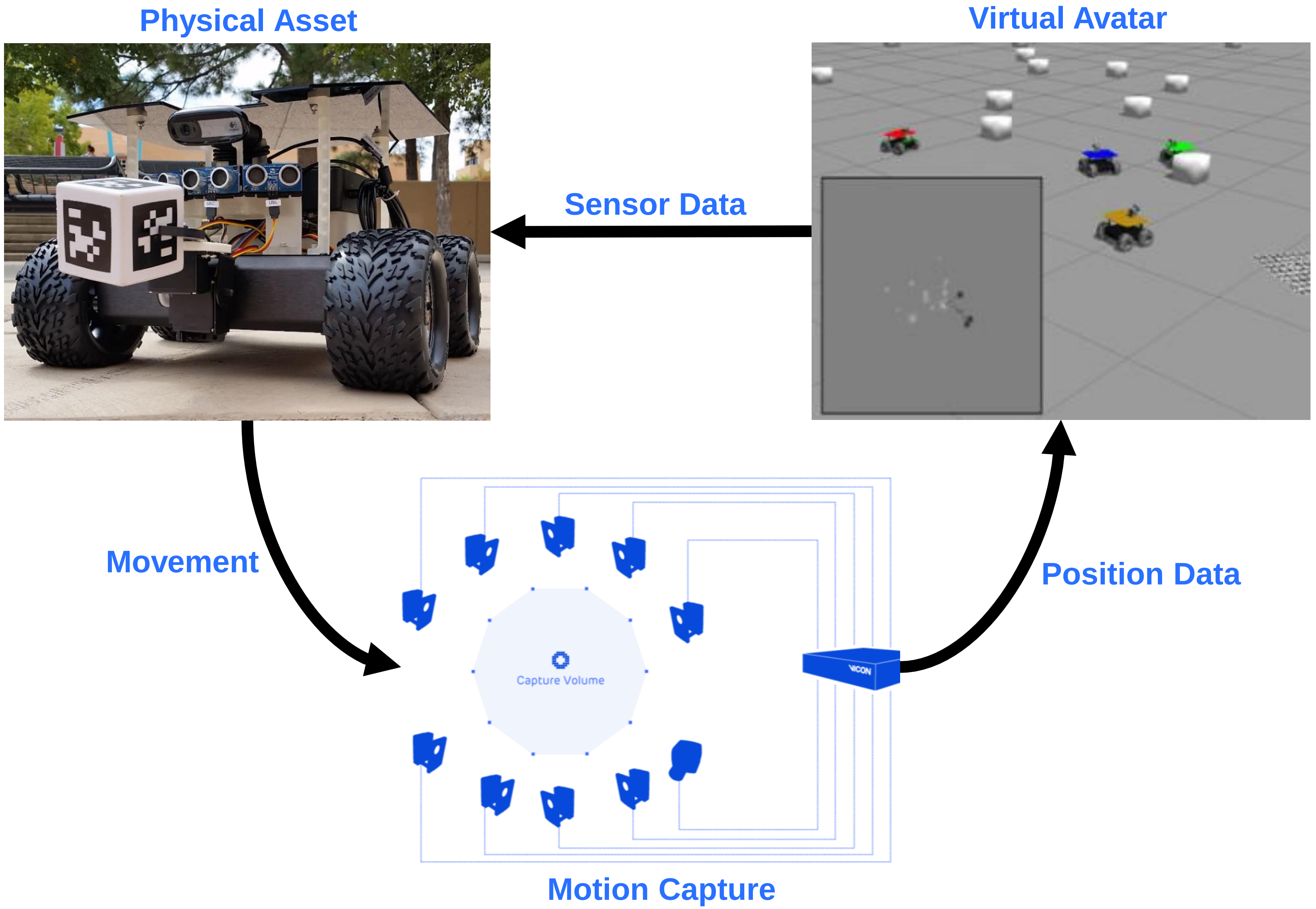
Motion Capture

Example LVC Instantiation

7. Real robot trusts simulated sensors and makes decisions as if it were inside simulated environment



Example LVC Instantiation



Implementation Details

- ROS 2D Navigation Stack on-board Swarmie (i.e. move_base):
  - Odometry input produced by encoders and IMU
  - Lidar input produced by Velodyne VLP-16
  - SLAM (gmapping package) generates occupancy grid map (offline)
  - AMCL localizes against stored occupancy map (online)
- Gazebo avatar model (unactuated)
  - VLP-16 sensor (velodyne_simulator package) replicates real lidar
  - Avatar set to static with gravity disabled
  - gazebo::physics::Entity::SetWorldPose() used to map real pose to sim

Demonstration Video Screenshots

Discussion/Future Work

- **Low-cost, low-risk, prototype demonstration of LVC architecture**
- **Opening up new possibilities for autonomy validation and verification:**

Limitations	Improvements
Lidar fusion brute force; Depends on SLAM	Intelligently fuse real and sim. point clouds
Only lidar implemented; Restricts learning, etc.	Phys./Sim. camera fusion with realistic render
Vicon needs stable, indoor space; Small robots	Investigate DGPS, others, for outdoor tracking
Demo tests only entry-level ROS functionality	Extend to production-level ROS codebases
Evaluations are hand-designed, one-off	Configure automated (Monte Carlo) testing

Wrap-Up/Credits

Questions?

Thanks: **Ryker Chute (Video Production/Editing)**

LMAS (General Support)