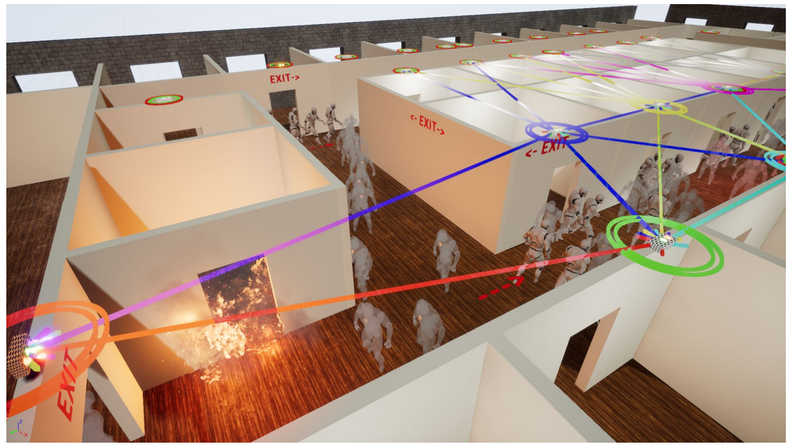
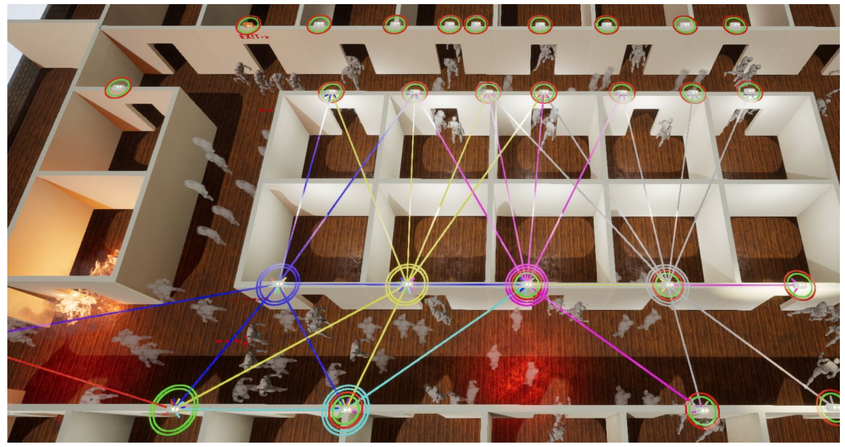
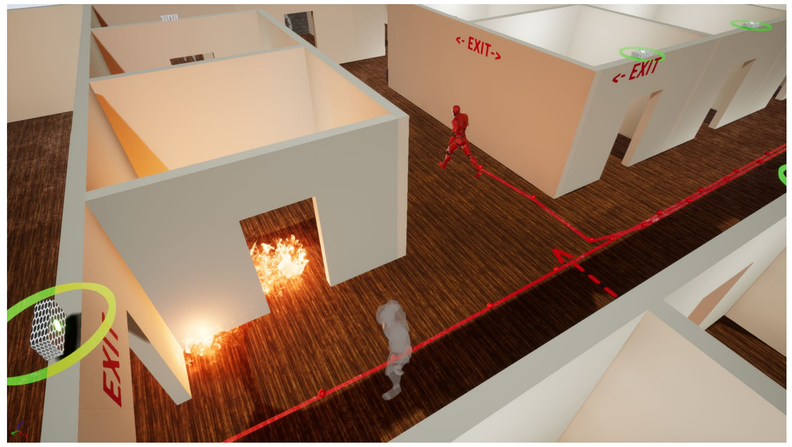
Imperial College London

DejaVu: Visual Diffing of Cyber-Physical Systems

Fergus Leahy, Naranker Dulay
Imperial College London, UK







Problem

- Testing CPS in the real-world is becoming more challenging
  - More devices, more services, more environmental interaction.
  - Testing is increasingly integrated into the environment.
- Traditional techniques:
  - **Code-deploy-test cycle** - *time consuming, costly, health/safety, slow iterations*
  - **Simulation** - *Poor environmental context using static traces or fabricated data*
  - Both require: Logs, logs and more logs.

Simulating CPS in Virtual Worlds

- Co-simulation approach:
  - Contiki Cooja WSN/CPS simulator simulates devices/nodes.
  - Unreal Engine 4, video game engine simulates physics, human mobility and the environment.
- **Dynamic simulations** of a CPS deployment within a virtual model of the target environment.
- **Features:**
  - **Mobility (**Dynamic and Reactive)
  - **Phenomena-on-demand**
  - **Time-control** (Slow/Pause/Fast)
  - **Visualisations** (Radio/meta-data)

Visual Diffing

- Enables viewers to intuitively see visual differences between two or more instances of a simulation incorporated directly into the visual medium.
- Inspired by uses in sports and video games:
  - Ghosts, visually adding 1 or more competitors to a scene to provide intuitive visual indications of progress and provide excitement, tension and entertainment.

Visual Diffing Techniques

- **Ghosts**: Enable two simulations' visual activity to be displayed and compared. Shown left and above.
  - Ghosts can be seen taking alternate paths.
- **Paths**: Visually log routes taken by individuals or crowds through an environment.
- **Colour & Size**: Visually alert viewers to differences in non-visual meta-data between simulations (sensor data, radio traffic, etc).

Replays

- Visual Diffs require recording entire simulation for full reconstruction and comparison:
  - Including data from virtual environment, sensors, and radio network.
  - Recorded at 60FPS for smooth simulation replay
  - Gigabytes of data/hour.
  - Uses an efficient segmented-compressed recording scheme.
- Enables full 360 replays, with full time-control (Rewind/Checkpoints).

Case Study

- CPS-enhanced fire detection and evacuation.
- Traditional method: utilises static signs, fixed routes and evacuee intuition - ***bad***.
- CPS method: Nodes detect fire and calculate safe exit route for evacuees - ***good***.
- Dynamically navigate evacuees away from hazards, towards safe exits, updating if hazards evolve.
- Using Visual diffs we can compare differences in before and after simulations directly
  - Images above show unaided ghost evacuees walking to the closest exit - towards danger, while solid evacuees are being directed towards safer routes.

Performance

- 30 Sensors & 60 Evacuees can be simulated up to 3x real-time.
- 2 Simulations can be played back simultaneously or 1 live, 1 recorded.
- Segmented compression scheme enables efficient storage and allows for long-term simulation tests.
- Running on a Xeon 6Core, 32GB RAM, GTX 970.

Replay Recording memory usage

	1min	1hr	24hr	7days
Raw	40MB	1.5GB	36GB	252GB
Compressed	1.6MB	45MB	1.1GB	8GB