

WORLD2DATA - ONE-PAGE TECHNICAL OVERVIEW

1. PROBLEM & CHALLENGE

Many AI and robotics teams cannot quickly convert real-world video into structured 3D data that is simulation-ready. Existing pipelines often require LiDAR, manual labeling, or brittle one-off scripts. This slows iteration and blocks reproducible evaluation.

2. TARGET AUDIENCE

Primary users:

- Robotics and embodied-AI teams needing fast world modeling.
- Simulation/digital-twin engineers who need OpenUSD-compatible scene assets.

Secondary users:

- Applied AI teams that need temporal object/state traces for downstream reasoning.

3. SOLUTION & CORE FEATURES

World2Data ingests 2D video and produces:

- Keyframes and camera trajectory.
- Dense 3D point cloud (MASt3R).
- Temporal scene graph with object/state metadata.
- OpenUSD export (`.usda`) for simulation workflows.
- Rerun recording (`.rrd`) for timeline playback and debugging.
- PLY export for standard point-cloud tools.

4. UNIQUE SELLING PROPOSITION (USP)

Single-video to OpenUSD with temporal scene context, confidence-aware reasoning hooks, and submission-ready visual outputs in one pipeline. No specialized capture hardware is required.

5. IMPLEMENTATION & TECHNOLOGY

- Geometry: MASt3R multi-view reconstruction.
- Semantics/Detection: YOLOv8 + SAM3 integration path + Gemini context.
- Reasoning: Gemini-backed semantic/causal augmentation with graceful fallback.
- Output/Interop: OpenUSD (`.usd-core`) + Rerun + JSON scene graph.
- Reliability: pytest suite with fast tests and overnight long-run mode.
- Architecture path to production: OpenUSD layer-based composition protocol (`.scene.usda` + role layers + overrides).

6. RESULTS & IMPACT

Recent run artifacts demonstrate:

- ~4.0M reconstructed 3D points.
- 120 temporal frames spanning the full source clip.
- Structured scene graph and USD export generated end-to-end.
- Robust completion even under external API quota constraints (degraded mode).

7. KNOWN PITFALLS & MITIGATION

- Pitfall: stale `.rrd` files can cause old timeline playback; mitigation is explicit artifact refresh before demo.
- Pitfall: monolithic USD output limits collaborative merges; mitigation is layered OpenUSD protocol with deterministic sublayer order.
- Pitfall: model-context caps in reasoning paths can hide late-scene details; mitigation is explicit full-span checks and protocolized provenance.

If we had 24 more hours: enforce namespace/provenance CI checks and complete strict per-component USD layer authoring.