DIGIMASQ AI

RESEARCH RFC #1

Persistent Generative Worlds

Addressable X:Y:Z:T state, independent generative realization, localized mutation, and historical reconstruction

Status Discussion Draft **Date** September 2026

Can an authored or AI-generated world be rendered again and again - from different viewpoints, after localized change, and at different points in its history - while remaining the same world?

Core thesis

The image is not the world. It is a view of the world.

The proposal is not that voxels are new. They are not. The research question is whether independently addressable spatial-temporal regions - potentially partitioned with voxels and grouped into persistent entities - can become reusable generative context that can be realized, changed, and re-rendered on demand.

1. The Problem

Generative AI has become remarkably good at creating convincing places. But creating a place once and maintaining that **same place** over time are different problems.

Generate a room. Render it from another viewpoint. Return tomorrow. Open a door. Move a chair. Break a window. Damage one wall. Allow vegetation to grow.

> **Can the system change exactly what changed and then generatively render the world again while everything else remains spatially and visually consistent?**

A high-quality second image is not enough. If the table moves, a window changes shape, or the room subtly changes dimensions simply because the generator composed a new frame, the system produced another plausible scene - not another view of the same persistent world.

2. An Adjacent Problem: Large Authored Worlds

Traditional game-engine pipelines solve an adjacent class of problems: representing enormous authored worlds, streaming only what is needed, and repeatedly rendering coherent views as the player moves through them.

Sucker Punch / large authored-world problem	Persistent generative-world problem
How do we represent and render an enormous authored world efficiently and coherently?	How do we represent an enormous authored **or AI-generated** world so it can be generatively rendered repeatedly, efficiently, coherently, and consistently over time?

Sucker Punch has publicly described world-building, fine-grained streaming, GPU-driven placement and vegetation systems used for *Ghost of Tsushima*. The comparison here is conceptual: persistent generative worlds add the requirement that some world content may be authored or generated while still needing stable identity and repeatable rendering.

© 2026 Digimasq AI, Inc. - Conceptual research discussion - implementation details intentionally omitted

3. Separate the World from the Rendering

A useful conceptual shift is to stop treating the generated image as the authoritative artifact.

Image-centric	World-centric
`Prompt -> generation -> image`	`World state -> view request -> generative realization -> rendered view`
The image is effectively the product.	The persistent world is the product; the image is one observation of it.

4. Addressable Spatial-Temporal State: X:Y:Z:T

Voxels are not the new idea. They have long been used in graphics, simulation, mapping, scientific computing, and games.

> **The proposal explored here is different: can independently addressable portions of a persistent world become independently generatable and re-renderable units of visual state?**

The underlying address space is best thought of as **X:Y:Z:T**. X, Y, and Z identify a region of space. T identifies the relevant point in world time or an event boundary.

A voxel - or another partitioning mechanism - can provide an addressable spatial volume within that model. The important property is not the voxel itself. It is what becomes possible when spatial-temporal regions can be independently identified, associated with persistent entities, visually realized when required, retained, changed, and re-rendered.

Research direction: persistent, independently addressable spatial-temporal regions as reusable generative context.

© 2026 Digimasq AI, Inc. - Conceptual research discussion - implementation details intentionally omitted

5. Voxels, Groups, and Persistent Entities

A voxel should not be confused with an entity. A wall, tree, vehicle, person, building, or landscape feature may occupy many spatial units while retaining one persistent identity.

Persistent entity
spatial occupancy
voxel / region grouping across X:Y:Z
state at T
generative realization when required

This allows generative rendering to operate on portions of a world while preserving the identity and state of the larger entities those portions comprise.

View-dependent realization

Spatial identity does not by itself solve visual identity. The same region may expose different surfaces from different viewpoints. An isolated object may be observable from many directions; a component embedded inside a building may expose only one relevant face.

That suggests another layer of indexing or realization: not a different world per viewpoint, but view-dependent visual information derived from the same persistent spatial-temporal state. The open question is how much of that visual information should be realized eagerly versus lazily as viewpoints are encountered.

> **The important constraint is that a new viewpoint must reveal the same underlying entity and world state, not cause the generator to invent a replacement.**

© 2026 Digimasq AI, Inc. - Conceptual research discussion - implementation details intentionally omitted

6. Localized Mutation and Generative Re-rendering

Once world state is independently addressable, localized mutation becomes the central generative question.

> **If a door opens, a building is damaged, an object moves, or vegetation changes, can we modify only the affected world state and then generatively render it again while unaffected regions preserve their spatial and visual identity?**

Persistent world at T0
state change / event
affected entity + spatial region
localized update
persistent world at T1
generative render from requested viewpoint

Not regenerate the scene.
Re-render the changed world.

7. Temporal Reconstruction and Anti-Revisionism

Because T is part of the address space, the system should be able to reconstruct and render the world as it existed at a prior time or event boundary.

If the current world contains a repaired building, a historical rendering from before the repair should show the damaged building. If a chair moved after Event 500, a rendering immediately before Event 500 should show the prior position.

> **Anti-revisionism principle: the present must not silently rewrite the past.**

A subtle case appears when a historical viewpoint was never previously rendered. The visual result may be generated for the first time today, but it must remain constrained by the world state that existed then. The rendering can be new; the historical state it depicts cannot be.

© 2026 Digimasq AI, Inc. - Conceptual research discussion - implementation details intentionally omitted

8. Open Research Questions

The model raises several questions worth challenging:

1. Is X:Y:Z:T the right address space for a reusable generative world?
2. Where does voxel partitioning help, and where does it become the wrong abstraction?
3. What belongs at the spatial-cell level versus the persistent-entity level?
4. How should related voxels or regions be grouped into entities without making localized updates too coarse?
5. How should view-dependent visual information be indexed or retained without pre-generating every possible viewpoint?
6. Can localized world changes reliably produce localized generative updates without visual drift outside the affected region?
7. How should newly exposed surfaces be generatively realized while remaining consistent with the entity's established identity?
8. How should lighting, weather, time, growth, damage, and movement alter appearance without changing unrelated geometry?
9. Can meshes, NeRFs, Gaussian splats, signed-distance fields, or other representations complement addressable spatial-temporal state?
10. How do we render historical viewpoints that were never previously observed without creating revisionist history?

9. Request for Comments

The hard problem may no longer be generating a convincing world once. It may be maintaining a world that remains itself across viewpoint, localized change, time, and repeated observation.

> **Is the next major problem in generative 3D creating better worlds - or making sure the world is still there the second time you look at it?**

References / precedent

1. Matthew Pohlmann, Sucker Punch Productions, "Samurai Landscapes: Building and Rendering Tsushima Island on PS4," GDC 2021. GDC Vault

2. Adrian Bentley, Sucker Punch Productions, "Zen of Streaming: Building and Loading Ghost of Tsushima," GDC 2021. GDC Vault

3. Eric Wohllaib, Sucker Punch Productions, "Procedural Grass in Ghost of Tsushima," GDC 2021. GDC Vault

Discussion draft. This document intentionally focuses on the problem space and conceptual research direction. Product architecture, persistence mechanics, storage design, reconstruction algorithms, and other implementation details are intentionally omitted.

© 2026 Digimasq AI, Inc. - Conceptual research discussion - implementation details intentionally omitted