ISSN: 2769-545X

Research Article

Advances in Machine Learning & Artificial Intelligence

A Unified ISR World Model: Vossels, Voxels, Mipvols, and Reinforcement Learning

Greg Passmore*

PassmoreLab, Austin, Texas, USA

***Corresponding Author**
Greg Passmore, PassmoreLab, Austin, Texas, USA.

Submitted: 2025, Oct 03; **Accepted:** 2025, Nov 05; **Published:** 2025, Nov 14

Citation: Passmore, G. (2025). A Unified ISR World Model: Vossels, Voxels, Mipvols, and Reinforcement Learning. *Adv Mach Lear Art Inte, 6*(4), 01-32.

Abstract

ISR enterprises must integrate measurements from many sensing modalities, radar, EO/IR, LiDAR, SIGINT, acoustics, cyber, weather, and jamming, each with its own resolution, structure, and format. Traditional systems often discard richness by reducing inputs to tracks, maps, or occasional images, leaving little for machine learning to exploit. This paper introduces Vossels (Volumetric Singular Spectral Elements), which were designed to address sensor discordance by storing raw sensor data in a unified atomic structure, ensuring that the full information content is preserved for artificial intelligence and reinforcement learning.

A Vossel encodes $(x, y, z, t, \lambda_i, a_j)$ as a singular volumetric spectral element, capturing space, time, spectral channels, and auxiliary attributes. Integrated into voxel grids and mipvol hierarchies, Vossels support multiresolution storage, retrieval, and time-series segmentation. Mipvols accelerate access by adapting resolution to projected target size, increasing speed without discarding data relevant to the task. This provides consistent, resolution-preserving storage that eliminates modality-specific silos, simplifies the mental model, and creates a centralized repository optimized for AI exploitation.

Because raw data is retained rather than simplified, reinforcement learning agents can operate directly on the richness of ISR streams, identifying correlations across modalities and scales. The memory manager and frustum extractor keep massive datasets tractable by keeping local neighborhoods in GPU VRAM or AI context windows while paging global volumes efficiently. This allows adaptive policies to evolve—learning when to refine, when to approximate, and how to balance fidelity with performance.

This paper defines the Vossel structure, the supporting voxel/mipvol hierarchy, and the mathematics of extraction and fusion. Kriging interpolation is presented for high-fidelity value estimation, view frustum extraction for adaptive retrieval, and multi-modal fusion strategies for co-registration. By design, Vossels enable AI and reinforcement learning to exploit ISR data in its full richness, transforming integration from siloed processes into a coherent, self-optimizing system.

Keywords: ISR Data Fusion, Multi-Modal Sensor Integration, Vossel Framework, Voxel Grid Representation, Mipvol Hierarchies, Kriging Interpolation, View Frustum Extraction, Reinforcement Learning for ISR, Multiresolution Data Management, EO/IR and Radar Fusion, Cyber and Jamming Integration, Scalable ISR Visualization

1. Introduction

Intelligence, Surveillance, and Reconnaissance (ISR) systems generate data from a wide spectrum of sources: electro-optical and infrared imagery, synthetic aperture radar, LiDAR point clouds, electronic emissions, cyber events, environmental fields, and multi-spectral or hyperspectral cubes. Each of these modalities arrives in a different structural form, rasters, volumes, tracks, signals, or logs, and are produced by sensors with unique geometries, resolutions, and statistical characteristics. Existing data structures handle these inputs in siloed ways: rasters are stored in grid-based formats, trajectories in tree indices, point clouds in unordered frames, and signals in time-frequency records. This fragmentation forces each modality into a heterogeneous pipeline, with separate methods for storage, indexing, subsampling, visualization, and fusion.

The result is inefficiency at multiple levels. At the computational level, data must be resampled or converted into secondary structures before joint analysis, incurring overhead and discarding fidelity. At the operational level, analysts are required to mentally reconcile disparate formats, shifting between images, tracks, and logs without a unified model of the environment. At the algorithmic level, reinforcement learning and other machine learning methods are forced to operate on reduced fidelity products, maps, tracks, or downsampled cubes, rather than the raw sensor measurements that carry the richest information content. These reductions are often irreversible, eliminating the anisotropy, blur, spectral spread, and uncertainty kernels that arise naturally from the physics of sensing.

Prior attempts at unification, such as temporal raster compression, trajectory indexing trees, or common point-cloud frames, have emphasized efficiency at the cost of fidelity. Compression schemes discard measurement geometry, point frameworks assume isotropy, and neural encoders abstract data into task-specific embeddings that no longer preserve the physical basis of observation. Format standards (e.g., HDF5, NGSI-LD, SensorML/O&M) improve interoperability but act as containers rather than representations, leaving the problem of how to capture the measurement itself unresolved.

The core problem, therefore, is the absence of a representation that simultaneously preserves raw sensor fidelity, supports efficient computation and memory management, and scales across heterogeneous ISR modalities. A solution would ideally unify sensing and non-sensing data within a consistent schema, maintain statistical and geometric detail, and remain tractable for real-time fusion, distributed processing, and AI exploitation. This is the gap that Vossels are designed to address.

2. Prior Work

2.1. Spatio-Temporal-Spectral Data Structures

Existing spatio-temporal data structures optimize for specific domains such as rasters, trajectories, or remote sensing imagery. Vossels differ by acting as a general abstraction: every sensor sample, regardless of type, is stored as a volumetric singular spectral element with spatial, temporal, and spectral attributes. This universality reduces the fragmentation of modality-specific data structures and creates a common foundation for fusion, interpolation, visualization, and AI-driven analysis. However, let us look in more detail.

The Temporal k²raster provides a compact representation and indexing method for temporal rasters [1]. The 2-3TR-tree extends this concept to trajectories of moving objects, supporting indexing of paths through continuous space [2]. A sliced representation model has been introduced to handle continuously changing spatial objects, partitioning them into temporal slices to capture evolution over time [3]. Adaptive tree structures have also been proposed that can dynamically switch between spatial and temporal indexing depending on query demands, improving efficiency under variable workloads [4]. More recently, spatio-temporal frameworks have been applied to problems such as clustering of space debris and spacecraft path optimization, demonstrating applicability in orbital dynamics and aerospace domains [5]. Parallel to these indexing methods, other work has focused on visualization and integration. Large wall displays have been used as a platform for handling video data in spatio-temporal form, while integrated storage structures have been designed for multi-dimensional remote sensing images to enable efficient retrieval across spectral and spatial dimensions [6,7]. Collectively, these methods are directed at the fundamental problem articulated by: the complexity inherent in representing, storing, and retrieving multidimensional data that evolves simultaneously in space, time, and spectral domains [8].

Let's take a closer look at a few of the methods described above.

2.2. Temporal k²-Raster

The Temporal k²-raster provides a compressed structure for time-varying rasters, but it assumes a 2D grid and a single scalar per cell [1]. In contrast, Vossels generalize to full 3D volumes with multi-spectral channels and auxiliary attributes, enabling representation of volumetric sensor elements such as LiDAR beams or radar resolution cells rather than just raster pixels.

An extension, the temporal k²-raster (tk²r) is designed to manage time-varying raster datasets. It represents information as a hierarchy of rasters, each cell storing a single scalar value indexed by spatial position and time, expressed as $R(x, y, t)$. This design is efficient for uniform gridded data such as land cover classification or temperature fields, where each sample is a homogeneous, single-valued measurement.

Vossels differ in both structure and scope. A Vossel is defined as an atomic record in the space where (x, y, z) are spatial coordinates, t is time, $\{\lambda_i\}$ are spectral channels, and $\{a_j\}$ are auxiliary attributes such as amplitude, polarization, or classification confidence. Each Vossel therefore represents not just a raster cell value but a multi-dimensional, modality-agnostic sample that can encode the geometry, variance, and physical observables of a sensor measurement.

This distinction has several implications. First, tk²r assumes a 2D raster base, whereas Vossels natively extend into three-dimensional volumes, allowing representation of LiDAR point clouds, volumetric SAR cells, or acoustic lobes. Second, tk²r stores a single scalar per cell, while Vossels can carry full spectral vectors and metadata attributes. Finally, tk²r is tied to regular grids, whereas Vossels act as unifying containers that can accommodate heterogeneous ISR modalities—optical pixels, radar resolution cells, SIGINT emissions, jamming fields, or cyber events—within the same structure.

Although useful, tk²r is optimized for uniform raster streams, while Vossels generalize to heterogeneous, high-dimensional ISR data. This generalization allows all sensing and non-sensing inputs to be stored in one consistent framework, enabling fusion, interpolation, and reinforcement learning over datasets that would otherwise remain siloed.

2.3. 2-3TR-Tree

The 2-3TR-tree was introduced to manage spatio-temporal trajectories by extending balanced search tree concepts into the temporal domain [2]. Each node maintains bounding rectangles over spatial extents coupled with temporal intervals, allowing efficient queries such as "which trajectories pass through region R during $[t_0, t_1]$?". Insertions and deletions are handled dynamically, making the 2-3TR-tree suitable for databases where objects move continuously and queries require path retrieval, intersection tests, and spatio-temporal joins. Its design emphasizes efficient indexing of $P(t) = (x(t), y(t))$ paths, typically with a small number of attributes beyond position.

Vossels diverge from this design in several fundamental ways. First, the 2-3TR-tree treats trajectories as first-class entities and compresses motion into linked path segments. By contrast, Vossels preserve each raw measurement with allowing full volumetric representation and λ_i, a_j encoding spectral and auxiliary attributes such as polarization, amplitude, confidence, or modulation. This distinction enables Vossels to capture not just the path of an object, but also its evolving physical observables across modalities.

Second, while the 2-3TR-tree is efficient for point-like trajectories, it is not designed for measurements that extend spatially, such as radar resolution cells, LiDAR beam footprints, or EO pixels. Vossels explicitly include projected geometry and statistical kernels, which allows anisotropic and blurred sensor footprints to be represented as probability distributions within a voxel volume. This feature is critical for ISR data, where sensing rarely yields perfect point measurements.

Finally, in terms of computational scope, the 2-3TR-tree is bounded by the requirements of trajectory indexing and retrieval. Vossels generalize this by enabling not only motion queries but also fusion, interpolation, and learning. A moving object is represented as a sequence of Vossels that can be grouped into voxels, accumulated into mipvol hierarchies, and queried via frustum extraction. This allows the same data to support both classical trajectory queries and advanced reinforcement learning tasks, where the agent must reason over evolving spatial signatures, temporal correlations, and multi-modal observables.

While the 2-3TR-tree provides an efficient indexing structure for trajectory databases, it is tied to position-centric representations. Vossels expand the abstraction to full volumetric, spectral, and attribute-rich measurements, enabling multi-modal exploitation and AI-driven analysis.

2.4. Sliced Representation Models

The sliced representation model addresses temporal change by storing spatial objects as a sequence of slices, each slice representing the geometry at a particular time step [3]. Continuity is enforced by linking successive slices, which enables queries over discrete states such as the geometry of an object at time t or its transformation between t_0 and t_1. This design is effective for surface-based or polygonal objects whose changes can be approximated as a series of snapshots, but it requires resampling all data into aligned temporal layers.

The limitation of this approach is that it assumes regular updates and bounded geometries, which does not reflect the irregular, asynchronous, and heterogeneous character of ISR data. Sensors rarely align to a common temporal grid, and their measurements often include uncertainty, blur, or probabilistic boundaries that cannot be easily expressed as crisp surfaces in slices.

By contrast, the Vossel framework avoids fixed temporal layering and instead supports continuous temporal evolution through direct incorporation of time into its volumetric-spectral schema. Rather than maintaining geometry in discrete snapshots, measurements are organized hierarchically into voxels and mipvols for efficient binning, retrieval, and interpolation. This preserves irregular sensor timing and allows data fusion across modalities without forcing artificial synchronization. Sliced models provide continuity across discrete frames, whereas Vossels enable scalable continuity across both time and modality in a unified representation.

2.5. Adaptive Tree Structures

Adaptive tree structures attempt to balance spatial and temporal indexing by dynamically switching between them based on query type [4]. For example, queries constrained in time may favor temporal indexing, while spatial queries rely on partitioning the data domain into regions or cells. This adaptability improves performance for workloads that shift between spatial and temporal dominance, but it requires the system to continuously monitor query patterns and restructure indices.

The weakness of this approach is that it still relies on rigid indexing strategies tied to either space or time as primary keys. ISR data is not easily reduced to such partitions, since it contains spectral vectors, auxiliary observables, and probabilistic kernels that span beyond simple spatio-temporal ranges. Adapting tree nodes can mitigate some inefficiencies, but the data remains locked into point or cell abstractions with limited extensibility.

Vossels extend beyond this paradigm by incorporating space, time, and spectrum simultaneously at the atomic level. Organizational structures such as voxels and mipvols are applied secondarily, not as the primary indexing basis. This duality of atomic records and volumetric grouping enables efficient memory management and distributed computation while preserving the multidimensional richness of the original measurement. Adaptive trees improve query flexibility within spatio-temporal domains, but Vossels provide a foundation where fusion, interpolation, and learning can occur across modalities without being bound to tree restructuring.

2.6. Spatio-Temporal indexing

Applications in space debris clustering and path optimization demonstrate the utility of spatio-temporal indexing for specialized domains [5]. Vossels provide a broader foundation: instead of task-specific indexing, they unify all ISR modalities—including imagery, radar, SIGINT, jamming, and cyber—into one structure that supports both trajectory-style queries and spectral fusion.

2.7. Applications-Oriented Structures

A number of spatio-temporal-spectral data structures have been built for specific workloads. Space-operations frameworks target orbital trajectories and collision risk, tuning indexes to predictive clustering and path optimization [5]. For high-throughput media and remote sensing, virtual large wall displays emphasize tiled, parallel playback of video streams, while integrated multi-dimensional image stores organize remote-sensing cubes with chunked tiling, pyramid levels, and metadata indexes to accelerate read-heavy raster analytics [6,7]. These systems show clear performance wins but share two traits: they assume regular grids (or tightly controlled tiling) and require modality-specific adaptation when data deviates from uniform rasters.

Samet's survey frames the indexing landscape—quadtrees, k-d trees, and higher-dimensional variants extended into time—prioritizing compression and query cost over preservation of measurement physics [8]. That abstraction is geometric and algorithmic, not semantic.

Vossels extend these ideas without inheriting their constraints. They retain raw observables (including anisotropy and uncertainty) and then layer voxels/mipvols for scalable storage and query, rather than forcing all inputs into uniform rasters or tiles. As a result, the same store supports reinforcement-learning policies and automated fusion across imaging, radar, LiDAR, SIGINT, cyber, and environmental feeds—combining fidelity preservation with computational efficiency in distributed or heterogeneous ISR environments.

2.8. Unifying Data Structures

Attempts at unifying spatio-temporal-spectral data structures can be grouped into three broad categories. The first category is volumetric stores, such as voxel grids, octrees, and their multi-resolution derivatives. These approaches are effective for spatial partitioning and memory efficiency, particularly when handling large image stacks, LiDAR scans, or SAR volumes, and they enable hierarchical querying. However, they require all incoming samples to be discretized into a fixed volumetric lattice. This resampling erases the native footprint, anisotropy, and statistical properties of the original measurements, reducing their utility for estimation-theoretic operations.

The second category is common point-cloud frames, which express heterogeneous sensor inputs as sets of 3D points with associated attributes. These frames facilitate alignment and registration across modalities and are widely used in robotics and geospatial pipelines. Extensions such as probabilistic projection maps and semantic voxel representations attempt to capture uncertainty and classification information. Yet point-based abstractions generally assume isotropic samples and do not encode blur, coherence, or spectral spread. Their memory footprint also grows rapidly without aggressive compression or clustering, making them less practical for ISR-scale data.

The third category is modality-agnostic neural encoders, where raw sensor data are projected into shared feature embeddings using learned networks [6,7]. This provides flexibility for classification, detection, and cross-modal reasoning, as the representation is automatically adapted to downstream tasks. However, embeddings abstract away the physical basis of the measurement. Once converted, the original footprint, point spread function, and spectral fidelity are lost. This limits their use in physically grounded fusion and estimation, or in reinforcement learning contexts where raw observables provide richer training signals.

3. Format Standards

A parallel line of work has pursued format and interoperability standards, includ-ing HDF5 for structured multi-dimensional storage, NGSI-LD for linked data integration, and SensorML/O&M for metadata and sensor descriptions. LLVM-inspired intermediate forms have been proposed to improve modularity and transformation pipelines. These standards reduce friction between systems and enable scalable data exchange, but they act primarily as containers rather than unifying representations. Like the categories above, they achieve efficiency or interoperability by forcing all measurements into uniform points, cells, or embeddings, thereby stripping away much of the richness that originates in the sensor physics.

4. Vossel Distinction

Vossels extend beyond these approaches by retaining the atomic measurement with its projected geometry, statistical kernel, anisotropy, and spectral attributes, embedding it directly into volumetric and temporal context. Unlike voxel and octree methods, they do not rely on a single discretization but allow aggregation into voxel bins and mipvol hierarchies as secondary organizational layers. Unlike point-cloud frameworks, they encode polarization, coherence, and uncertainty rather than assuming isotropy. Unlike neural encoders, they unify modalities without discarding raw fidelity or requiring task-specific retraining. Finally, unlike format standards, they constitute a mathematically grounded representation rather than a passive container.

The duality of storage—raw spectral elements organized into volumetric hierarchies—provides computational efficiency and scalable memory management independent of GPU versus CPU execution. This design enables distributed and cluster computing, with voxel binning and mipvol hierarchies supporting efficient query and adaptive fidelity. The result is a framework that preserves the richness of sensor physics while remaining computationally tractable at ISR scale, and that directly supports reinforcement learning agents and automated fusion methods operating over heterogeneous, high-dimensional inputs.

Structure	Domain	Representation	Strengths	Limitations	Vossel Contrast
Temporal k²raster (Cerdeira-Pena et al. 2018)	2D rasters over time	Hierarchy of rasters, one scalar per cell $R(x, y, t)$	Compact, efficient for uniform grids	Limited to 2D, single scalar, assumes homogeneity	Vossels generalize to $(x, y, z, t, \lambda_i, a_j)$, capturing 3D, multispectral, and auxiliary attributes with variance
2-3TR-tree (Abdelguerfi et al. 2002)	Moving objects	Spatio-temporal tree combining 2- and 3-way branching	Efficient trajectory indexing and retrieval	Restricted to discrete path storage; does not capture measurement physics	Vossels encode full sensor samples (geometry, PSF, spectral spread) rather than just paths
Sliced representation (Forlizzi et al. 2000)	Continuously changing spatial objects	Sequential slices of object states	Maintains continuity over time	Inherits discretization artifacts; unsuitable for high-dimensional ISR	Vossels model continuous distributions via kernels, preserving anisotropy and spectral detail
Visualization and storage (Wang et al. 2003; Zhang et al. 2017)	Video walls, remote sensing images	High-performance visual and multi-dimensional image access	Optimized for human use and imagery	Limited to rasterized video or image cubes	Vossels combine visualization with mathematically grounded representation, enabling fusion and RL across modalities
Vossels	ISR multi-modal	Volumetric singular spectral elements $(x, y, z, t, \lambda_i, a_j)$ + voxel/mipvol hierarchy	Preserves sensor fidelity; supports fusion, scalable memory management, RL integration	Higher storage requirements than compressed schemes; emphasizes fidelity over compression	Provides universal, physically grounded abstraction across ISR modalities

5. Comparative Analysis

Prior spatio-temporal-spectral data structures have emphasized indexing, com-pression, or specialized representations for efficiency. The Temporal k²raster achieves compact representation of time-varying rasters but is limited to uniform 2D grids with a single scalar per cell, optimized for homogeneous raster streams such as temperature or land cover fields [1]. The 2-3TR-tree was designed to index moving object trajectories by combining two- and three-way branching factors, enabling efficient retrieval of discrete paths but offering little support for continuous measurement geometry or multi-modal fusion [2]. Sliced representation models attempt to preserve continuity of spatial objects across time by storing sequential “slices” of state, but they inherit discretization artifacts and are poorly suited to heterogeneous, high-dimensional ISR modalities [3]. Other work has emphasized visualization and storage, such as virtual large wall displays for video data and integrated multi-dimensional storage structures for remote sensing images [6,7]. These approaches provide high-performance access for specialized im-agery but remain tied to rasterized or modality-specific forms.

Whereas these structures optimize for specific data types—rasters, trajectories, videos, or image cubes—Vossels act as a universal abstraction. Each measurement, whether optical, radar, LiDAR, SIGINT, cyber, or environmental, is represented as a volumetric singular spectral element with spatial, temporal, spectral, and auxiliary attributes. This reduces fragmentation across modalities, simplifies fusion, and provides a consistent basis for reinforcement learning and adaptive ISR analysis. Unlike voxel and octree methods, Vossels are not bound to a fixed discretization: voxel bins and mipvol hierarchies are applied only as secondary organizational structures, leaving the atomic measurement intact. Unlike pointcloud frameworks, they do not assume isotropy but explicitly preserve anisotropy, coherence, blur, polarization, and spectral spread. Unlike neural encoders, they unify modalities without discarding raw fidelity or requiring retraining on task-specific embeddings. Finally, unlike passive container standards such as HDF5, NGSI-LD, or SensorML/O&M, Vossels constitute a mathematically grounded representation that retains the physics of the sensing process.

This distinction is operationalized by embedding atomic spectral elements directly into volumetric and temporal context, while applying statistical kernels, view frustum extraction, kriging interpolation, and multiresolution mipvol hierarchies to manage data adaptively. By retaining the projected geometry and uncertainty kernel of each measurement, Vossels maintain the dual representation of geometric footprint and statistical variance. This allows heterogeneous inputs to be normalized into a single data model while preserving the physics of collection. In practice, this means that radar resolution cells, optical pixels, LiDAR beams, acoustic lobes, SIGINT emissions, cyber events, and environmental fields can all be ex-pressed within the same structure, consistently aligned and fused.

The dual structure—atomic Vossels embedded within voxel and mipvol hierarchies—provides computational efficiency and scalable memory management independent of CPU versus GPU execution. Voxels serve as spatial bins that cluster measurements for locality of reference, while mipvols provide multiresolution detail management, ensuring that queries and visualization operate only at the reso-lution required. This design enables intrinsic reduction without discarding raw measurements: high-level tasks can exploit voxel-level aggregates, while precision analysis reverts to the underlying Vossels. On CPUs, voxel binning accelerates cache coherence; on GPUs, mipvol subsampling reduces VRAM demand; and in distributed settings, voxel partitions map naturally across clustered environments. The result is a fidelity-preserving yet computation-aware framework suitable for ISR exploitation at scales ranging from embedded platforms to global clusters.

6. Vossels and Reinforcement Learning

Reinforcement learning benefits directly from this design. Agents can exploit the voxel–mipvol hierarchy for curriculum training, beginning with coarser mipvol levels to learn policies over large-scale patterns before refining them on raw Vossel data. This reduces training time, provides natural hierarchical policy development, and eliminates the need for external resampling. By contrast, structures such as the Temporal k²raster or trajectory trees achieve efficiency primarily through compression and indexing, forcing simplification into uniform rasters or trajectory records. Vossels avoid this reduction, preserving the underlying measurement physics while still enabling scalable management. This makes them well suited for reinforcement learning agents that require rich, high-dimensional inputs and consistent semantics across sensing domains.

Vossels also provide significant advantages beyond reinforcement learning. They act as hyperdimensional state elements in which spatial coordinates, temporal attributes, modality-specific measurements, environmental modifiers, degraders, and uncertainty descriptors are co-embedded in a unified structure. Each Vossel is therefore a pre-normalized representation, ensuring that all channels are inherently aligned in the same coordinate system and update cycle. This removes the need for extensive preprocessing, including normalization, co-registration, and cross-modal alignment, which is required in conventional ISR pipelines. The result is reduced computational overhead, lower latency, and direct applicability for machine learning and spiking architectures.

A major benefit of this unified design is that analytic operations need not be performed on fragmented or separately conditioned products. Segmentation, recognition, correlation, and prediction operate directly on the Vossel substrate, which is already fused and semantically coherent. This prevents errors that arise when sub-systems attempt to reconcile outputs that were produced under inconsistent assumptions. It also allows reinforcement learning agents to operate on data that is physically grounded, temporally consistent, and enriched with environmental and degradative context.

Another advantage is that Vossels encode degraders and environmental factors alongside direct sensor returns. Jamming, cyber-induced network degradation, atmospheric absorption, scattering, diffusion, and weather are not handled externally but are represented as explicit channels within the same compositional framework. Updates to these channels condition observations in place, reducing sensor confidence or modifying likelihoods locally rather than requiring separate bookkeeping. This integration produces a more resilient operational model: when one modality is compromised, other modalities are weighted accordingly, and confidence is adjusted without external intervention.

The structure also facilitates adaptive resolution management. Coarser mipvols allow global context to be preserved, while raw Vossels maintain the full measurement physics for finegrained exploitation. This hierarchy ensures that both reinforcement learning agents and analytic processes can scale from broad situational understanding to detailed forensic analysis without abandoning the common substrate. In practice, this improves training efficiency, supports multiscale optimization, and ensures continuity between coarse decision-making and finegrained evidence.

Vossels extend beyond compression or indexing to provide a universal, physically grounded abstraction for ISR. They preserve raw sensor fidelity, capture environmental and degradative effects in the same representational space, support adaptive multiresolution storage, and enable reinforcement learning and automated fusion methods over heterogeneous, high-dimensional data. By shifting emphasis from storage efficiency alone to a representation optimized for fusion, adaptability, and AI-driven exploitation, Vossels provide broader operational capability and a more direct path from sensing to decision than any prior spatio-temporal-spectral structure.

7. Vossel Design

ISR sensor data originates across the electromagnetic spectrum and may be expressed in the form of points, images, or depth maps depending on the modality. Each sensor operates at a different target or ground resolution, and therefore the resulting measurements are heterogeneous in scale. To preserve fidelity, the system stores the data at full native resolution. However, subsequent analytical or visualization tasks do not always require this level of detail, and so the data can later be retrieved at an appropriate resolution. Because many applications require efficient access to neighboring measurements, the data must be organized in a clustered fashion that facilitates local searches.

The measurements collected by each sensor are not uniform in their structure. In addition to spatial position, described by (x, y, z) coordinates, a sample may include amplitude, polarization, color, phase, spectral shift, or other physical observables, and each record carries a time stamp t. Thus, an individual measurement is represented as a Vossel:

$$v = x, y, z, t, \lambda_1, \ldots, \lambda_n, a_1, \ldots, a_m,$$

where λ_i denotes the spectral bands and a_j the auxiliary attributes.

Before the data is integrated into a common framework, each sensor feed is subjected to preprocessing steps. These include sensor-specific geometric and photometric corrections, amplitude normalization, and local operations such as gamma correction and sharpening. Once normalized, the feeds are routed into the Vossel Integrator, which performs single-channel mosaic buildup, multichannel registration, warping, rectification, and ultimately aggregation into the world model state.

As part of this integration, the data is binned into voxels to provide a computational grouping mechanism. A voxel is defined as

$$V_{i,j,k} = [x_0 + i\Delta x,, x_0 + (i+1)\Delta x) \times [y_0 + j\Delta y,, y_0 + (j+1)\Delta y) \times [z_0 + k\Delta z,, z_0 + (k+1)\Delta z),$$

with i, j, k indexing the grid. Each Vossel belongs uniquely to one voxel:

$$\forall v \in \mathcal{E},; \exists (i,j,k) : (x,y,z) \in V_{i,j,k}.$$

These voxels are further organized into mipvol hierarchies, enabling multiresolution access. Because each Vossel is time stamped, data can be accumulated into a single voxel structure and then segmented into time-series voxel volumes. The segmentation interval is application-defined: for low-volume sensors, a time slice may represent one minute, whereas for high-volume sensors, one second slices are more appropriate. A time-sliced voxel set is therefore:

$$\mathcal{V}[t_0, t_1] = v \in \mathcal{E} \mid t_0 \le t \le t_1.$$

When analysis or visualization is required, a frustum extractor defines the selection of voxels or mipvols from the world model. A frustum query $\boldsymbol{F}$ with near plane h and far plane y extracts:

$$\mathcal{F}(F, [t_0, t_1]) = v \in \mathcal{E} \mid (x, y, z) \in F,; t_0 \le t \le t_1.$$

The fidelity of this extraction can be adapted to the performance requirements of the task. For low-performance, low-fidelity applications, Vossels may be used directly as statistical representations of voxels, i.e.,

$$V_{i,j,k}^{\text{stat}} = \frac{1}{N_{i,j,k}} \sum_{v \in V_{i,j,k}} f(v).$$

For high-fidelity operations, kriging interpolation is applied to estimate desired values at specific coordinates (x, y, z, t):

$$\hat{f}(x, y, z, t) = \sum_{n=1}^{N} w_n(x, y, z, t), f(v_n),$$

with weights w_n determined by the spatial–temporal covariance model.

The degree of granularity is also adjustable during frustum extraction. Typically, lower resolution mipvol levels are used for regions that are farther from the hither plane, with higher resolution reserved for the near field. Yon planes may be set to limit the overall data volume or may be defined adaptively, either globally or on a per-channel basis, depending on the deterioration of signal-to-noise ratio with distance.

Through this sequence of operations, ISR data is transformed from heterogeneous sensor feeds into a coherent world model that can be queried at appropriate temporal and spatial scales. The combination of formal primitives (Vossels), organizational structures (voxels, mipvols), and extraction operators (frustum selection, kriging interpolation) provides both efficiency and adaptability, ensuring that analysis and visualization tasks can operate on the resolution and fidelity best suited to the mission requirements.

8. Vossel Definition

A Vossel (Volumetric Singular Spectral Element) is an atomic spatio-spectral-temporal sample with real-world coordinates and associated footprint and uncertainty, which we can define as:

$$e_n = \big(\mathbf{p}_n,\ \mathcal{F}_n,\ \Sigma_n,\ \boldsymbol{\lambda}_n,\ \mathbf{a}_n\big),$$

where

- $\mathbf{p}_n = (x_n, y_n, z_n, t_n) \in D \times \mathbb{T}$ is the nominal space–time center,
- $\mathcal{F}_n \subset \mathbb{R}^3$ is the projected footprint envelope (deterministic support),
- Σ_n is a positive semidefinite kernel/covariance describing spatial (and optionally temporal/spectral) uncertainty,
- $\boldsymbol{\lambda}_n \in \mathbb{R}^{B_n}$ is the spectral vector (possibly band-dependent length B_n),
- $\mathbf{a}_n \in \mathbb{R}^{M_n}$ collects auxiliary attributes (e.g., intensity, phase, polarization, SNR, class/confidence).

This definition preserves the dual nature of a measurement: $\mathcal{F}$ captures determinis-tic footprint geometry, Σ and captures stochastic variance around it.

9. Structural Relationships

The Vossel is defined as the atomic sensing element, but its utility emerges only when it is related to higher-order organizational structures. The voxel grid provides the discretized support in which Vossels are embedded. Defined over a spatial domain with fixed origin and cell sizes, the voxel grid partitions space into half-open volumes, ensuring that the domain of interest can be exhaustively covered by a finite set of cells. This provides the scaffolding for indexing and aggregation without requiring that the raw measurements themselves be resampled.

The element set ε is the collection of all Vossels in the scene, each carrying its spatio-temporal position, footprint, uncertainty kernel, and spectral attributes. These elements are positioned continuously in space, but for computational tractability they must be assigned to discrete bins in the grid. The element–voxel assignment formalizes this step: each Vossel is mapped deterministically to a voxel by integer division of its coordinates relative to the grid origin and cell sizes. This produces a one-to-one containment when variance is negligible, but more generally requires a probabilistic assignment when the uncertainty kernel spans multiple cells.

The probabilistic assignment leads directly to the definition of occupancy. Rather than a binary indicator, the fractional occupancy $\chi_{i,j,k}(n)$ is computed by integrating the Vossel's uncertainty kernel over the voxel volume. In the limit of negligible variance this reduces to 0 or 1, but when the Vossel footprint overlaps several cells the occupancies act as weights distributed across them. The triple-sigma occupancy ensures conservation by guaranteeing that the sum of weights across all voxels for a given Vossel equals one, thereby preserving the total element count under probabilistic spreading.

Self-containment follows as a consistency condition. Provided that the voxel grid covers the sensing domain, the set of all Vossel centers lies within the union of voxel supports, and for every element there exists at least one cell containing it. This ensures that the voxelized representation is complete and that no measurement is lost or discarded in the embedding process.

Aggregation over voxels extends this containment by allowing functions of the elements to be accumulated cell by cell. Any per-element attribute—such as amplitude, polarization energy, or classification score—can be summed within a voxel using the occupancy weights. Conservation identities guarantee that the global sum of any attribute over all Vossels equals the sum over all voxel aggregates, ensuring exactness and enabling efficient reduction operations.

A multiresolution hierarchy, or mipvol construction, generalizes the grid into a pyramid of scales. By doubling the voxel size at each level, parent volumes subsume the contents of their children while maintaining strict count consistency. This allows adaptive fidelity: queries can be answered coarsely at high levels or precisely at the base resolution, and reinforcement learning can exploit coarse-to-fine training curricula. The dual representation, atomic Vossels preserved in continuous space, organized hierarchically into voxels and mipvols, creates a framework that is simultaneously fidelity-preserving, computationally efficient, and scalable across hardware architectures.

9.1. Voxel Grid

Let the spatial domain be $D \subset \mathbb{R}^3$ with origin (x_0, y_0, z_0).

Fix cell sizes Δx, Δy , $\Delta z > 0$ and extents $N_x, N_y, N_z \in \mathbb{Z} > 0$.

9.2. Definition of a voxel

A voxel (half-open cuboid cell) is

$$V_{i,j,k} = [x_0 + i\Delta x,\ x_0 + (i+1)\Delta x) \times [y_0 + j\Delta y,\ y_0 + (j+1)\Delta y) \times [z_0 + k\Delta z,\ z_0 + (k+1)\Delta z),$$

with integer indices

$$i \in [0, N_x - 1], \quad j \in [0, N_y - 1], \quad k \in [0, N_z - 1] \cap \mathbb{Z}.$$

The collection of all voxels partitions the grid support:

$$\operatorname{supp}(\mathcal{G}) = \bigcup_{i=0}^{N_x-1} \bigcup_{j=0}^{N_y-1} \bigcup_{k=0}^{N_z-1} V_{i,j,k}.$$

By construction, the voxels are **disjoint except at boundaries**:

$$V_{i,j,k} \cap V_{i',j',k'} = \varnothing \quad \text{if } (i,j,k) \neq (i',j',k').$$

9.3. Indexing Map

Define the index map $\phi : D \rightarrow \mathbb{Z}^3$ by

$$\phi(x,y,z) = \left(\left\lfloor \frac{x-x_0}{\Delta x} \right\rfloor, \left\lfloor \frac{y-y_0}{\Delta y} \right\rfloor, \left\lfloor \frac{z-z_0}{\Delta z} \right\rfloor \right).$$

For any point $\mathbf{r} = (x, y, z) \in D$, $\phi(\mathbf{r}) = (i, j, k)$ yields the voxel index such that $\mathbf{r} \in V_{i,j,k}$.

This mapping is surjective onto the voxel index set $\{0,\ldots,N_x - 1\} \times \{0,\ldots,N_y - 1\} \times \{0,\ldots,N_z - 1\}$.

9.4. Volume and Centroid

The volume of a voxel is constant:

$$\mathrm{Vol}(V_{i,j,k}) = \Delta x\, \Delta y\, \Delta z.$$

The centroid is

$$\mathbf{c}_{i,j,k} = \Big(x_0 + (i+\tfrac{1}{2})\Delta x,\;\; y_0 + (j+\tfrac{1}{2})\Delta y,\;\; z_0 + (k+\tfrac{1}{2})\Delta z \Big).$$

These centroids form a lattice of points embedded in D.

9.5. Indicator Functions

Define the indicator of voxel membership:

$$\mathbf{1}_{i,j,k}(\mathbf{r}) = \begin{cases} 1 & \text{if } \mathbf{r} \in V_{i,j,k}, \\ 0 & \text{otherwise.} \end{cases}$$

Then any measurable function $f : D \to \mathbb{R}$ can be decomposed as

$$f(\mathbf{r}) = \sum_{i,j,k} f(\mathbf{r})\, \mathbf{1}_{i,j,k}(\mathbf{r}).$$

Integration reduces to

$$\int_D f(\mathbf{r})\, d\mathbf{r} = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sum_{k=0}^{N_z-1} \int_{V_{i,j,k}} f(\mathbf{r})\, d\mathbf{r}.$$

9.6. Voxel Occupancy (Counts)

Given an element set $\varepsilon = \{e_n\}$ with positions $\mathbf{r}_n = (x_n, y_n, z_n)$, define the voxel count

$$C_{i,j,k} = \sum_{n=1}^{N} \mathbf{1}_{i,j,k}(\mathbf{r}_n).$$

Conservation identity:

$$\sum_{i,j,k} C_{i,j,k} = N.$$

In the probabilistic case with soft occupancies $\chi_{i,j,k}(n) \in [0, 1]$, the generalized count is

$$C_{i,j,k} = \sum_{n=1}^{N} \chi_{i,j,k}(n), \qquad \sum_{i,j,k} C_{i,j,k} = N.$$

10. Non-Cartesian Voxel Domains

Voxels need not be restricted to Cartesian lattices. Certain classes of problems are better expressed in alternate domains such as Fourier space, spherical–polar coordinates, or other non-Euclidean geometries. In general form, a voxelized field can be defined as a mapping

$$V : \Omega \to \mathbb{R}^n$$

where Ω is a discrete sampling of a continuous manifold $\mathcal{M}$ equipped with a coordinate metric g. The metric determines the adjacency and locality relations among voxels. For standard volumetric grids, $\mathcal{M} = \mathbb{R}^3$ and g is Euclidean, but this is only a special case.

In Fourier space, the voxel indices correspond to frequency coordinates (k_x, k_y, k_z), and the field V (k_x, k_y, k_z) encodes spatial frequency amplitude and phase rather than position. This formulation is advantageous for convolution, filtering, and correlation, where differential operators become multiplicative.

In spherical or polar voxelizations, the coordinates are expressed as (r, θ, ϕ), and the discretization is uniform in angular rather than linear space. Such arrangements are efficient for radiance fields, omnidirectional sensing, or any geometry exhibiting radial symmetry.

More generally, non-Cartesian lattices can be defined on curvilinear or manifold-adapted spaces, such as cylindrical (r, ϕ, z), log-polar, or geodesic grids. These support applications in flow visualization, astronomical mapping, and imaging where anisotropy or curvature makes uniform Euclidean sampling inefficient.

Under this view, a voxel is simply a discrete tensor element defined over an arbitrary coordinate manifold, with topology and neighborhood structure inherited from g. The voxel concept therefore extends naturally to any domain where continuity and adjacency can be formalized, independent of whether the underlying space is Cartesian, spectral, or curved.

11. Multiresolution Hierarchy

Define a sequence of grids $\mathcal{G}^{(\ell)}$ indexed by level $\ell \in \mathbb{Z}_{\geq 0}$.
At level ℓ the cell sizes are

$$\Delta x^{(\ell)} = 2^{\ell} \Delta x, \quad \Delta y^{(\ell)} = 2^{\ell} \Delta y, \quad \Delta z^{(\ell)} = 2^{\ell} \Delta z.$$

Each voxel at level ℓ is the union of eight voxels at level $\ell - 1$:

$$V^{(\ell)}_{i,j,k} = \bigcup_{\alpha,\beta,\gamma \in \{0,1\}} V^{(\ell-1)}_{2i+\alpha,\, 2j+\beta,\, 2k+\gamma}.$$

Counts aggregate consistently:

$$C^{(\ell)}_{i,j,k} = \sum_{\alpha,\beta,\gamma \in \{0,1\}} C^{(\ell-1)}_{2i+\alpha,\, 2j+\beta,\, 2k+\gamma}.$$

Thus, total conservation holds at every level:

$$\sum_{i,j,k} C^{(\ell)}_{i,j,k} = N \quad \forall \ell.$$

The voxel grid provides a partition of space into uniform volumetric bins with explicit index mapping, centroid locations, indicator functions, and count conservation. By embedding Vossel elements into this grid, either deterministically via index maps, or probabilistically via kernelized occupancies, it becomes possible to aggregate, query, and downsample measurements while preserving global consistency. The multiresolution extension forms the basis of mipvol hierarchies (more on this later), ensuring scalable representation across CPU, GPU, and distributed architectures.

12. Element Set

Let the spatial domain be $D \subset \mathbb{R}^3$ and time domain $\mathbb{T} \subset \mathbb{R}$.
As a reminder, a Vossel is an atomic sample

$$e_n \;=\; \big(\, \mathbf{p}_n,\; \mathcal{F}_n,\; \Sigma_n,\; \boldsymbol{\lambda}_n,\; \mathbf{a}_n \,\big),$$

where

- $\mathbf{p}_n = (x_n, y_n, z_n, t_n) \in D \times \mathbb{T}$ is the nominal space–time center,
- $\mathcal{F}_n \subset \mathbb{R}^3$ is the projected footprint envelope (deterministic support),
- Σ_n is a positive semidefinite kernel/covariance describing spatial (and optionally temporal/spectral) uncertainty,
- $\boldsymbol{\lambda}_n \in \mathbb{R}^{B_n}$ is the spectral vector (possibly band-dependent length B_n),
- $\mathbf{a}_n \in \mathbb{R}^{M_n}$ collects auxiliary attributes (e.g., intensity, phase, polarization, SNR, class/confidence).

The element set is

$$\mathcal{E} = \{e_n\}_{n=1}^{N}, \qquad N < \infty.$$

13. Coordinate and Attribute Projections

Define projection maps:

$$\pi_x(e_n) = x_n,\;\; \pi_y(e_n) = y_n,\;\; \pi_z(e_n) = z_n,\;\; \pi_t(e_n) = t_n,$$

$$\pi_\lambda(e_n) = \boldsymbol{\lambda}_n,\;\; \pi_a(e_n) = \mathbf{a}_n.$$

Concatenate $\mathbf{r}_n = (x_n, y_n, z_n)$ and $\mathbf{p}_n = (\mathbf{r}_n, t_n)$.

14. Empirical Measures Over Space–Time and Spectrum

The empirical space–time measure of centers:

$$\widehat{\mu}_{ST} \;=\; \frac{1}{N}\sum_{n=1}^{N} \delta_{\mathbf{p}_n},$$

with $\delta_{\mathbf{p}n}$ the Dirac at $\mathbf{p}_n$.

If a per-element nonnegative weight w_n (e.g., quality, confidence) is available:

$$\widehat{\mu}_{ST}^{(w)} \;=\; \frac{1}{\sum_n w_n}\sum_{n=1}^{N} w_n\, \delta_{\mathbf{p}_n}.$$

For a fixed band index b (or spectral response R_b):

$$\widehat{\mu}_{ST}^{(b)} \;=\; \frac{1}{Z_b}\sum_{n=1}^{N} \mathbf{1}\{b \in e_n\}\, w_{n,b}\, \delta_{\mathbf{p}_n}, \quad Z_b = \sum_{n} \mathbf{1}\{b \in e_n\}\, w_{n,b}.$$

15. Kernelized Occupancy (soft presence in a region)

Let $K_n(\cdot\,; \mathbf{p}_n, \Sigma_n)$ be the normalized spatial kernel induced by Σ_n around $\mathbf{r}_n$.

For any region $A \subset D$:

$$\chi_A(n) \;=\; \int_A K_n(\mathbf{q}; \mathbf{p}_n, \Sigma_n)\, d\mathbf{q} \;\in [0,1],$$

with partition of unity

$$\sum_{\{V_{i,j,k}\}} \chi_{V_{i,j,k}}(n) = 1.$$

This reduces to a hard indicator when $\Sigma_n \to 0$ and $\mathcal{F}_n \subset A$.

16. Neighborhoods and Selection Sets

Given a query $\mathbf{p}_0 = (\mathbf{r}_0, t0)$ and selection parameters

$$\rho > 0 \ \ \text{(spatial radius)}, \quad \tau > 0 \ \ \text{(temporal half-width)}, \quad F \subset D \ \ \text{(view frustum)},$$

define the neighbor index set

$$\mathcal{N}(\mathbf{p}_0; \rho, \tau, F) = \Big\{ n \ : \ \|\mathbf{r}_n - \mathbf{r}_0\| \le \rho, \ \ |t_n - t_0| \le \tau, \ \ \mathbf{r}_n \in F \Big\}.$$

A band/material filtered neighborhood is

$$\mathcal{N}_{b,m} = \big\{ n \in \mathcal{N} : \ b \in e_n, \ \ m_n = m \big\},$$

where m_n is an optional material label.

17. Empirical Moments (per band or attribute)

For a scalar per-element function $f(e_n)$:

$$\widehat{\mu}_f \ = \ \frac{1}{N} \sum_{n=1}^{N} f(e_n), \qquad \widehat{\sigma}_f^2 \ = \ \frac{1}{N-1} \sum_{n=1}^{N} \big(f(e_n) - \widehat{\mu}_f \big)^2.$$

Over a neighborhood $\mathcal{N}$:

$$\widehat{\mu}_f(\mathcal{N}) = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} f(e_n), \qquad \widehat{\sigma}_f^2(\mathcal{N}) = \frac{1}{|\mathcal{N}| - 1} \sum_{n \in \mathcal{N}} \big(f(e_n) - \widehat{\mu}_f(\mathcal{N}) \big)^2.$$

For multiband vectors $\boldsymbol{\lambda}_n \in \mathbb{R}^B$:

$$\widehat{\Gamma}_\lambda \ = \ \frac{1}{N} \sum_{n=1}^{N} \big(\boldsymbol{\lambda}_n - \bar{\boldsymbol{\lambda}} \big) \big(\boldsymbol{\lambda}_n - \bar{\boldsymbol{\lambda}} \big)^\top, \quad \bar{\boldsymbol{\lambda}} = \frac{1}{N} \sum_n \boldsymbol{\lambda}_n.$$

18. Cross-Covariances (space–time vs. spectrum)

Let $g(\mathbf{p}_n)$ be a scalar function on space–time and $h(\boldsymbol{\lambda}_n)$ a scalar spectral function.

$$\widehat{\mathrm{Cov}}[g, h] \ = \ \frac{1}{N} \sum_{n=1}^{N} \Big(g(\mathbf{p}_n) - \bar{g} \Big) \Big(h(\boldsymbol{\lambda}_n) - \bar{h} \Big).$$

This quantifies coupling useful for co-kriging or feature selection.

19. Element Graphs (for fusion/association)

Define a proximity graph $G = (V, E)$ with $V = \{1, \ldots, N\}$ and

$$(n, m) \in E \iff \|\mathbf{r}_n - \mathbf{r}_m\| \le \rho_s, \ \ |t_n - t_m| \le \rho_t.$$

Optionally weight edges by similarity:

$$w_{nm} \;=\; \underbrace{\exp\Big(-\tfrac{\|\mathbf{r}_n-\mathbf{r}_m\|^2}{2\alpha^2}\Big)\exp\Big(-\tfrac{(t_n-t_m)^2}{2\beta^2}\Big)}_{\text{space–time affinity}} \cdot \underbrace{\frac{\boldsymbol{\lambda}_n^\top \boldsymbol{\lambda}_m}{\|\boldsymbol{\lambda}_n\|\,\|\boldsymbol{\lambda}_m\|}}_{\text{spectral similarity}}.$$

This graph supports track-building, clustering, and factor-graph fusion.

20. Element-Level Likelihoods

Given a forward model $\mathcal{M}$ producing expected attribute $f_\theta(\mathbf{p}, \boldsymbol{\lambda})$ with parameter $\boldsymbol{\theta}$ and noise covariance Σ_n:

$$\ell_n(\theta) \;\propto\; \exp\Big(-\tfrac{1}{2}\big(f(e_n) - f_\theta(\mathbf{p}_n, \boldsymbol{\lambda}_n)\big)^\top \Sigma_n^{-1}\big(f(e_n) - f_\theta(\mathbf{p}_n, \boldsymbol{\lambda}_n)\big)\Big).$$

This leads to MAP/ML estimators $\hat{\theta}$ and uncertainty propagation over ε.

21. Element Histograms and Densities

For a measurable set $A \subset D \times \mathbb{T}$ and spectral gate $B \subset \mathbb{R}^B$:

$$H(A, B) \;=\; \sum_{n=1}^{N} \mathbf{1}\{\mathbf{p}_n \in A\}\,\mathbf{1}\{\boldsymbol{\lambda}_n \in B\}.$$

Kernel density estimate (KDE) in space–time:

$$\hat{f}(\mathbf{p}) \;=\; \frac{1}{N}\sum_{n=1}^{N} \kappa_{ST}(\mathbf{p} - \mathbf{p}_n),$$

extendable to joint KDEs with spectrum or attributes for adaptive neighbor selection.

22. Element Formalism

The expanded element set formalism treats ε not just as a list of points but as a structured, measurable collection with:

(i) projection maps for coordinates and attributes;
(ii) empirical measures for counting and weighting;
(iii) kernelized occupancies for soft spatial presence;
(iv) neighborhood definitions for local inference;
(v) moment and covariance operators across bands and space–time;
(vi) similarity graphs for fusion; and
(vii) likelihoods for model-based estimation.

This mathematical scaffolding is what enables consistent kriging, co-kriging, frustum queries, fusion/association, and reinforcement learning policy training directly over Vossels.

23. Element–Voxel Assignment

Each Vossel element e_n has a nominal position $\mathbf{p}_n = (x_n, y_n, z_n) \in D$.
Given a voxel grid $\mathcal{G}$ with origin (x_0, y_0, z_0) and cell sizes $(\Delta x, \Delta y, \Delta z)$, we assign each element to a voxel either deterministically or probabilistically.

24. Deterministic Index Assignment

For $\mathbf{p}_n \in \mathrm{supp}(\mathcal{G})$, define

$$i_n = \left\lfloor \frac{x_n - x_0}{\Delta x} \right\rfloor, \quad j_n = \left\lfloor \frac{y_n - y_0}{\Delta y} \right\rfloor, \quad k_n = \left\lfloor \frac{z_n - z_0}{\Delta z} \right\rfloor.$$

By the half-open voxel definition, membership is unique:

$$\mathbf{p}_n \in V_{i_n,j_n,k_n}.$$

Equivalently, the mapping ϕ: $\varepsilon \rightarrow \{0,\ldots,N_x - 1\} \times \{0,\ldots,N_y - 1\} \times \{0,\ldots, N_z - 1\}$ sends each e_n to exactly one (i, j, k).

25. Probabilistic Assignment with Uncertainty Kernels

When an element has non-negligible footprint or uncertainty Σ_n, its presence may straddle voxel boundaries.
Let $P(\mathbf{q} \mid \mathbf{p}_n, \Sigma_n)$ be the normalized spatial kernel induced by Σ_n around $\mathbf{p}_n$.
Define the occupancy weight of element n in voxel (i, j, k) as

$$\chi_{i,j,k}(n) \; = \; \int_{V_{i,j,k}} P(\mathbf{q} \mid \mathbf{p}_n, \Sigma_n)\, d\mathbf{q}.$$

This satisfies:
– **Range:** $0 \leq \chi_{i,j,k}(n) \leq 1$ for all (i, j, k).
– **Partition of unity:**

$$\sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sum_{k=0}^{N_z-1} \chi_{i,j,k}(n) = 1.$$

– **Deterministic limit:** if $\Sigma_n \rightarrow 0$ and $\mathbf{p}_n$ lies strictly inside a single voxel, then

$$\chi_{i,j,k}(n) \to \begin{cases} 1, & (i,j,k) = (i_n, j_n, k_n), \\ 0, & \text{otherwise.} \end{cases}$$

A common anisotropic Gaussian choice is

$$P(\mathbf{q} \mid \mathbf{p}_n, \Sigma_n) = \frac{1}{(2\pi)^{3/2}\,|\Sigma_n|^{1/2}} \exp\!\left(-\tfrac{1}{2}(\mathbf{q}-\mathbf{r}_n)^\top \Sigma_n^{-1}(\mathbf{q}-\mathbf{r}_n)\right), \qquad \mathbf{r}_n = (x_n, y_n, z_n).$$

26. Occupancy-Based Counts and Conservation

Define voxel counts by summing occupancies across elements:

$$C_{i,j,k} = \sum_{n=1}^{N} \chi_{i,j,k}(n).$$

Global conservation holds in both deterministic and probabilistic cases:

$$\sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sum_{k=0}^{N_z-1} C_{i,j,k} = N.$$

Thus, deterministic assignment yields one-hot membership , while probabilistic assignment yields soft membership that respects uncertainty and footprint overlap, both consistent with total element conservation.

27. Triple-Sigma Occupancy

Let's go over triple-sigma occupancy for one gaussian vossel. Let the voxel grid have unit cells and origin at (0, 0, 0). Consider a single Vossel e_n whose spatial kernel is an axis-aligned 3D Gaussian with mean near a voxel corner so that mass spills into neighbors:
– Mean (spatial center): $\boldsymbol{\mu} = (\mu_x, \mu_y, \mu_z) = (0.45, 0.45, 0.45)$
– Standard deviations: $\sigma_x = \sigma_y = \sigma_z = \sigma = 0.30$
– Kernel: $K(\mathbf{q}) = \mathcal{N}(\mathbf{q} \mid \boldsymbol{\mu},\ \mathrm{diag}(\sigma^2, \sigma^2, \sigma^2))$

A voxel is

$$V_{i,j,k} = [i, i+1) \times [j, j+1) \times [k, k+1).$$

The probabilistic occupancy of this Vossel in $V_{i,j,k}$ is

$$\chi_{i,j,k}(n) = \iiint_{V_{i,j,k}} K(\mathbf{q})\, d\mathbf{q}.$$

Because the kernel is separable (diagonal covariance), this factorizes into 1D Gaussian CDF differences along each axis:

$$\chi_{i,j,k}(n) = \left[\Phi\left(\tfrac{i+1-\mu_x}{\sigma}\right) - \Phi\left(\tfrac{i-\mu_x}{\sigma}\right)\right] \cdot \left[\Phi\left(\tfrac{j+1-\mu_y}{\sigma}\right) - \Phi\left(\tfrac{j-\mu_y}{\sigma}\right)\right] \cdot \left[\Phi\left(\tfrac{k+1-\mu_z}{\sigma}\right) - \Phi\left(\tfrac{k-\mu_z}{\sigma}\right)\right],$$

where $\Phi(u) = \frac{1}{2}\left[1 + \operatorname{erf}(u/\sqrt{2})\right]$ is the standard normal CDF.

28. 1D Slice Weights (shared for x, y, z)

With $\mu = 0.45$ and $\sigma = 0.30$,

$$\begin{aligned} w_{[-1,0)} &= \tfrac{1}{2}\left[\operatorname{erf}\left(\tfrac{0-\mu}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\tfrac{-1-\mu}{\sqrt{2}\sigma}\right)\right], \\ w_{[0,1)} &= \tfrac{1}{2}\left[\operatorname{erf}\left(\tfrac{1-\mu}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\tfrac{0-\mu}{\sqrt{2}\sigma}\right)\right], \\ w_{[1,2)} &= \tfrac{1}{2}\left[\operatorname{erf}\left(\tfrac{2-\mu}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\tfrac{1-\mu}{\sqrt{2}\sigma}\right)\right]. \end{aligned}$$

Numerically (with $\sqrt{2}\sigma \approx 0.424$),

$$w_{[-1,0)} \approx 0.0689, \quad w_{[0,1)} \approx 0.8980, \quad w_{[1,2)} \approx 0.0330,$$

which sum to ≈ 1.000 per axis.

29. 3D Voxel Occupancies (Top Contributors)

Since $\chi_{i,j,k}(n)$ is the product of the three 1D weights for the corresponding *x, y, z* intervals, the largest masses occur near the mean voxel $[0, 1)^3$.

– **Center voxel** $(i, j, k) = (0, 0, 0)$:

$$\chi_{0,0,0} = w_{[0,1)}^3 \approx 0.898^3 \approx 0.725.$$

– **Face neighbors (one axis in neighbor cell),** e.g. (1, 0, 0), (0, 1, 0), (0, 0, 1):

$$\chi_{1,0,0} \approx w_{[1,2)} w_{[0,1)}^2 \approx 0.0266,$$

while negative-side faces (−1, 0, 0), (0,−1, 0), (0, 0,−1) give

$$\chi_{-1,0,0} \approx w_{[-1,0)} w_{[0,1)}^2 \approx 0.0555.$$

– **Edge neighbors (two axes in neighbor cells),** e.g.

$$\chi_{1,1,0} \approx w_{[1,2)}^2 w_{[0,1)} \approx 0.0010, \quad \chi_{1,-1,0} \approx w_{[1,2)} w_{[-1,0)} w_{[0,1)} \approx 0.0020.$$

– **Corner neighbors (three axes in neighbor cells):**

$$\chi_{1,1,1} \approx w_{[1,2)}^3 \approx 3.6 \times 10^{-5}, \qquad \chi_{-1,-1,-1} \approx w_{[-1,0)}^3 \approx 3.3 \times 10^{-4}.$$

30. Compact Occupancy Table (approximate)

Voxel (i, j, k)	$\chi_{i,j,k}(n)$
(0,0,0)	0.725
(-1,0,0), (0,-1,0), (0,0,-1)	0.0555 each
(1,0,0), (0,1,0), (0,0,1)	0.0266 each
(1,1,0), (1,0,1), (0,1,1)	0.0010 each
(1,-1,0) and perms	0.0020 each
(-1,-1,0) and perms	0.0043 each
(1,1,1)	3.6×10^{-5}
(-1,-1,-1)	3.3×10^{-4}

The sum over all voxels equals (within rounding) 1.0 because the kernel is normalized:

$$\sum_{i,j,k} \chi_{i,j,k}(n) = 1.$$

31. Triple-Sigma Truncation

Because a 3D Gaussian concentrates ≈ 99.7% of its mass within 3σ along each axis, occupancy can be truncated to voxels within $\pm 3\sigma$ of $\boldsymbol{\mu}$:

$$\mathcal{N}(e_n; 3\sigma) = \big\{(i, j, k) : [i, i+1) \cap [\mu_x - 3\sigma, \mu_x + 3\sigma] \neq \emptyset, \text{ and similarly for } y, z\big\}.$$

Then

$$\sum_{(i,j,k) \in \mathcal{N}(e_n; 3\sigma)} \chi_{i,j,k}(n) \approx 0.997.$$

32. Takeaways

- Separable kernels reduce 3D occupancy integrals to products of 1D CDF differences.
- Most mass stays in the home voxel and face neighbors; edges and corners carry small tails.
- Triple-sigma truncation provides efficient computation with negligible loss.
- These $\chi_{i,j,k}(n)$ feed directly into voxel counts, weighted aggregates, and kriging/co-kriging estimators.

33. Global Conservation Identity

The voxel counts are defined as aggregated occupancies:

$$C_{i,j,k} = \sum_{n=1}^{N} \chi_{i,j,k}(n)$$

where $\chi_{i,j,k}(n)$ is the occupancy weight of element e_n in voxel $V_{i,j,k}$.

Step 1: Total Count Over Grid

Summing across all voxels in the grid gives

$$\sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sum_{k=0}^{N_z-1} C_{i,j,k} = \sum_{i,j,k} \sum_{n=1}^{N} \chi_{i,j,k}(n)$$

Step 2: Exchange the Order of Summation

By linearity, the order of summation can be swapped:

$$\sum_{i,j,k}\sum_{n=1}^{N}\chi_{i,j,k}(n)=\sum_{n=1}^{N}\sum_{i,j,k}\chi_{i,j,k}(n)$$

Step 3: Deterministic Case
If each element e_n lies fully inside a single voxel (hard assignment), then

$$\sum_{i,j,k}\chi_{i,j,k}(n)=1$$

since exactly one voxel receives membership $\chi_{i,j,k}(n) = 1$ and all others receive 0.
Therefore,

$$\sum_{i,j,k}C_{i,j,k}=\sum_{n=1}^{N}1=N$$

Step 4: Probabilistic Case
If elements are modeled with uncertainty kernels Σ_n, then occupancies are fractional but normalized:

$$\sum_{i,j,k}\chi_{i,j,k}(n)=1\quad\text{for each } n$$

because $\chi_{i,j,k}(n)$ partitions the unit mass of the distribution $P(\mathbf{q} \mid \mathbf{p}_n, \Sigma_n)$ across the voxel grid.

Hence,

$$\sum_{i,j,k}C_{i,j,k}=\sum_{n=1}^{N}1=N$$

34. Interpretation
This identity shows that the voxelized representation is conservative: the total count of elements is invariant under discretization.

Whether occupancy is hard (one-hot) or soft (fractional), the sum over all voxels always recovers exactly the total number of elements N.

35. Triple-Sigma Interpretation
When the uncertainty kernel Σ_n is modeled as a Gaussian covariance, the occupancy weights $\chi_{i,j,k}(n)$ are defined as integrals of the Gaussian probability density over each voxel $V_{i,j,k}$:

$$\chi_{i,j,k}(n)=\iiint_{V_{i,j,k}}\mathcal{N}(\mathbf{q}\mid\mathbf{p}_n,\Sigma_n)\,d\mathbf{q}.$$

Step 1: Gaussian Mass Concentration
For a univariate Gaussian distribution, approximately 99.7% of the probability mass lies within ±3σ of the mean. In three dimensions, the joint kernel is separable along principal axes, so nearly all of the probability is concentrated inside the cube:

$$[\mu_x-3\sigma_x,\ \mu_x+3\sigma_x]\times[\mu_y-3\sigma_y,\ \mu_y+3\sigma_y]\times[\mu_z-3\sigma_z,\ \mu_z+3\sigma_z].$$

Step 2: Restricting the Neighborhood
Define the triple-sigma neighborhood of element en as the set of voxels whose support overlaps this region:

$$\mathcal{N}(e_n;3\sigma)=\Big\{(i,j,k):V_{i,j,k}\cap[\mu_x-3\sigma_x,\mu_x+3\sigma_x]\times[\mu_y-3\sigma_y,\mu_y+3\sigma_y]\times[\mu_z-3\sigma_z,\mu_z+3\sigma_z]\neq\emptyset\Big\}.$$

Step 3: Conservation Within Triple-Sigma
By construction, the sum of occupancies restricted to this neighborhood captures nearly all of the Vossel's probability mass:

$$\sum_{(i,j,k)\in\mathcal{N}(e_n;3\sigma)} \chi_{i,j,k}(n) \approx 0.997.$$

The small remainder outside this region corresponds to the Gaussian tails, which can be safely neglected for most ISR applications.

Step 4: Practical Implication
This triple-sigma truncation provides a principled cutoff for voxel assignment in probabilistic mode. It ensures:
– Computational tractability by limiting the number of voxels evaluated per element.
– Statistical fidelity by preserving almost all of the probability mass.
– Consistency with Gaussian probability bounds, giving an interpretable error margin.

In effect, each Vossel's probabilistic spread can be localized to a finite set of voxels without losing more than 0.3% of the probability mass. This guarantees both efficiency and fidelity in voxelized aggregation, kriging, and reinforcement learning workflows.

36. Set Containment

A voxel grid partitions the spatial domain $D \subseteq \mathbb{R}^3$ into disjoint cells $\{V_{i,j,k}\}$.
Each Vossel e_n has a nominal center $\mathbf{p}_n = (x_n, y_n, z_n)$.
To ensure well-defined indexing, every element must fall inside the support of the grid.

Step 1: Grid Support
The union of all voxels defines the domain covered by the grid:

$$\operatorname{supp}(\mathcal{G}) = \bigcup_{i=0}^{N_x-1} \bigcup_{j=0}^{N_y-1} \bigcup_{k=0}^{N_z-1} V_{i,j,k}.$$

This is the finite region of space where voxelization is valid.

Step 2: Containment Requirement
The set of all Vossel centers $\{\mathbf{p}_n\}_{n=1}^{N}$ is contained within the grid support if

$$\{\mathbf{p}_n\}_{n=1}^{N} \subseteq \operatorname{supp}(\mathcal{G}).$$

That is, every element lies in at least one voxel.

Step 3: Element-to-Voxel Existence
Equivalently, this requirement can be stated in elementwise form:

$$\forall n \in \{1, \dots, N\},\ \exists (i,j,k)\ \text{ such that }\ \mathbf{p}_n \in V_{i,j,k}.$$

This guarantees that each Vossel has a unique voxel assignment under the half-open voxel boundary convention.

Step 4: Boundary Handling
Because voxel intervals are half-open (closed on the lower bound, open on the upper), boundary cases are uniquely assigned.

For example, a point at $x = x_0 + (i + 1)\Delta x$ belongs to voxel $V_{i+1,j,k}$, not $V_{i,j,k}$, avoiding ambiguity at shared faces.

Step 5: Probabilistic Occupancy Case
When Vossels carry uncertainty kernels Σ_n, set containment extends to distributional support.
Instead of a single voxel, the kernel spreads mass across neighbors:

$$\sum_{i,j,k} \chi_{i,j,k}(n) = 1,$$

so that total membership is still contained within the grid, even if distributed probabilistically.
Set containment enforces that all Vossels are represented within the voxelized domain.

Deterministically, each point belongs to exactly one voxel; probabilistically, occupancy spreads across multiple voxels but remains normalized to the grid support.

This property guarantees global conservation of mass and ensures that subsequent aggregation, interpolation, and fusion remain well-posed.

37. Aggregation Over Voxels

Once Vossels are assigned to voxels, we can aggregate per-element quantities into voxel-level summaries.

This provides a bridge from atomic measurements to gridded statistics while preserving conservation.

Step 1: Per-Element Function
Let $f : \varepsilon \rightarrow \mathbb{R}$ be any scalar-valued function defined on the element set.
Examples include:
– $f(e_n)$ = intensity or amplitude,
– $f(e_n)$ = confidence score,
– $f(e_n)$ = indicator of a material class.

Step 2: Voxel-Wise Aggregation
For voxel (i, j, k), define the aggregated statistic:

$$S_{i,j,k} = \sum_{n=1}^{N} f(e_n)\, \chi_{i,j,k}(n).$$

Here $\chi_{i,j,k}(n)$ is the occupancy of element e_n in voxel $V_{i,j,k}$, which equals 1 if deterministic membership holds, or a fractional weight in the probabilistic case.

Step 3: Deterministic Case
If $\chi_{i,j,k}(n) \in \{0, 1\}$, then $S_{i,j,k}$ is simply the sum of $f(e_n)$ over all elements whose centers fall in voxel (i, j, k).

Step 4: Probabilistic Case
If e_n spans multiple voxels, each voxel receives a fractional contribution:

$$0 \leq \chi_{i,j,k}(n) \leq 1, \qquad \sum_{i,j,k} \chi_{i,j,k}(n) = 1.$$

Thus, $S_{i,j,k}$ becomes a weighted contribution proportional to the overlap of the uncertainty kernel with the voxel.

Step 5: Global Conservation
Summing across all voxels yields:

$$\sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sum_{k=0}^{N_z-1} S_{i,j,k} = \sum_{n=1}^{N} f(e_n) \sum_{i,j,k} \chi_{i,j,k}(n).$$

By normalization of occupancies, $\sum_{i,j,k} \chi_{i,j,k}(n) = 1,$ so:

$$\sum_{i,j,k} S_{i,j,k} = \sum_{n=1}^{N} f(e_n).$$

This states that the total aggregated value over the grid equals the total over all elements, guaranteeing no loss or duplication.

Step 6: Interpretation

– If $f(e_n) = 1$, then $S_{i,j,k}$ counts elements, and conservation says the total count across voxels is N.
– If $f(e_n)$ is intensity, then $S_{i,j,k}$ represents voxel-level energy accumulation, conserved across the grid.
– If $f(e_n)$ is a class indicator, then $S_{i,j,k}$ is a soft histogram of class membership over space.

Aggregation over voxels generalizes counting to arbitrary per-element functions.

It supports deterministic or probabilistic assignments, while ensuring global conservation of sums.

This mechanism enables voxelized statistics such as densities, intensities, or class distributions, forming the basis for kriging, interpolation, and reinforcement learning rewards.

38. Multiresolution (MipVols)

A voxel grid can be organized into a hierarchy of progressively coarser levels, called mipvols (multiresolution volumes). This enables adaptive queries, visualization, and reinforcement learning by controlling the resolution at which data are accessed.

Step 1: Definition of Levels

For base cell sizes Δx, Δy, Δz, the level- ℓ grid has expanded cell sizes:

$$\Delta x^{(\ell)} = 2^{\ell}\Delta x, \qquad \Delta y^{(\ell)} = 2^{\ell}\Delta y, \qquad \Delta z^{(\ell)} = 2^{\ell}\Delta z,$$

where $\ell \in \mathbb{Z}_{\geq 0}$ is the hierarchy index.

- $\ell = 0$ is the native resolution (original voxel grid).
- $\ell = 1$ merges $2 \times 2 \times 2$ voxels into one parent.
- Larger ℓ values represent increasingly coarser approximations.

Step 2: Parent–Child Relation (Octree Form)

Each voxel at level ℓ corresponds to a union of eight child voxels at level $\ell - 1$:

$$V_{i,j,k}^{(\ell)} = \bigcup_{\alpha,\beta,\gamma \in \{0,1\}} V_{2i+\alpha,\, 2j+\beta,\, 2k+\gamma}^{(\ell-1)}.$$

This is exactly the octree construction: each parent voxel contains $2^3 = 8$ children.

Step 3: Count Consistency

Let $C_{i,j,k}^{(\ell)}$ denote the occupancy count (or aggregated statistic) in voxel (i, j, k) at level ℓ.

Then the parent count is the sum of its children:

$$C_{i,j,k}^{(\ell)} = \sum_{\alpha,\beta,\gamma \in \{0,1\}} C_{2i+\alpha,\, 2j+\beta,\, 2k+\gamma}^{(\ell-1)}.$$

This guarantees conservation of totals across levels.

Step 4: Global Conservation Across Levels

Summing across all voxels at level ℓ:

$$\sum_i \sum_j \sum_k C_{i,j,k}^{(\ell)} = N, \qquad \forall \ell.$$

Thus, no matter how coarse the hierarchy, the total number of elements represented remains exactly N.

Step 5: Interpretation
– At $\ell = 0$, each Vossel contributes at its native voxel.
– At $\ell = 1$, contributions are aggregated into $2 \times 2 \times 2$ blocks, yielding a coarser but still conserved representation.
– At higher ℓ, the hierarchy provides a pyramid of representations, enabling algorithms to operate at appropriate levels of detail.

Step 6: Applications
– Visualization: low-resolution mipvols allow fast rendering of large datasets.
– Queries: hierarchical search reduces computation, as coarse queries prune regions before refinement.
– Learning: reinforcement learning agents can train on coarse levels first, then refine at fine levels (curriculum training).
– Storage: mipvols provide a natural compression hierarchy while retaining conservation guarantees.

The mipvol hierarchy organizes voxel grids into multiresolution levels with exact conservation of counts and attributes. This provides adaptive fidelity, computational efficiency, and scalability across CPUs, GPUs, and clusters, while ensuring that total measurements remain consistent across all levels.

39. Abstract Representation of a Vossel Sample

Unifying heterogeneous sensor feeds requires more than reducing each observation to a spatial point. Every measurement carries a finite region of influence dictated by sensor physics and projection geometry. This influence is represented in two layers. The first is the projected geometry, a deterministic approximation of the footprint imposed by the sensing system. The second is the statistical variance, which captures the uncertainty, blur, and distortion introduced by optics, propagation, and the environment.

Projected geometry provides a first-order mapping of how a measurement interacts with the world. LiDAR pulses expand as cones terminating in circular or elliptical patches on terrain. Optical pixels, under camera models, project to quadrilaterals on the ground corresponding to detector array positions. SAR resolution cells are oriented ellipses, defined by range resolution along one axis and azimuth along another. These geometric approximations enable disparate sensors to be aligned in a shared model. Importantly, F_r is not a hard boundary but an envelope of strongest influence. True sensitivity distributions extend beyond this region, overlapping adjacent measurements.

The physical response is always softer than geometry alone suggests. Optical pixels exhibit Gaussian-like point spread functions (PSFs), causing gradual roll-off and overlap between neighboring pixels. LiDAR returns carry beam spread and sub-sampling artifacts that extend influence beyond the ideal cone. Radar and acoustic samples smear anisotropically with incidence angle, terrain slope, and propagation medium. These effects cannot be reduced to geometry and require a probabilistic description.

For this reason, each measurement is paired with a statistical kernel Σ_r that models the distribution of influence around its nominal footprint. In isotropic cases, this may be a circular Gaussian centered on the projection. In anisotropic cases, it becomes an oriented ellipse stretched along sensor geometry or environmental gradients. Beyond Gaussians, certain sensors exhibit higher-order or multimodal kernels: optical Airy disks, radar speckle, or acoustic side lobes. Variance extends naturally to time (latency, timestamp jitter) and spectrum (wavelength calibration error, quantization), broadening the kernel concept beyond space alone.

The duality of projected geometry F_r and statistical variance Σ_r defines a generalized sensor model. The geometry encodes the deterministic footprint implied by the measurement system, while the variance describes the uncertainty cloud that softens this boundary. This abstraction makes LiDAR pulses, SAR cells, EO pixels, and acoustic lobes mathematically equivalent: volumetric elements with a footprint and a probability distribution. On this basis, one can construct unified operations for fusion, normalization, and multi-modal extraction across all sensing modalities.

40. Voxels and MipVols

The Vossel abstraction defines the atomic measurement, but voxels and their mipvol hierarchies provide the computational structure that makes the framework operational at scale. Each Vossel is assigned to a voxel grid, introducing spatial locality and enabling efficient neighborhood search. This binning step is essential, since direct querying of billions of raw samples across ISR datasets would be computationally prohibitive.

The mipvol hierarchy extends this organization into multiple resolutions. Fine-scale fidelity is preserved where the task requires detail, while coarser levels accelerate retrieval in regions where the projected target size does not demand native resolution. In this way, the hierarchy provides adaptive fidelity, balancing precision against computational load.

Although a Vossel is independently defined, voxel and mipvol layers form the scaffolding required for tractable implementation. They ensure scalability, efficient retrieval, and manageable memory use, making the representation viable for enterprise-level ISR exploitation.

41. Formalization

Each sensor reading r is represented as a Vossel sample V_r, which has two components:
1. **Projected Geometry (Footprint)** – the first-order geometric envelope of the region covered by the measurement.
2. **Statistical Variance (Uncertainty Kernel)** – the second-order stochastic representation of positional error, blur, and distortion.

42. Projected Geometry

Let $\mathbf{p} = (x, y, z)$ be the nominal measurement location.
Define the footprint shape F_r as the envelope in space where the sample is expected to have influence.

This footprint is never truly bounded by sharp edges; instead, it is modeled as an idealized form that represents the primary extent of the sample, while acknowledging that the physical sensitivity function has gradual roll-off and overlap with adjacent samples.

Examples:
- **LiDAR:** F_r is modeled as a cone truncated at terrain with radius set by beam divergence. In reality, the beam has a Gaussian profile, producing a soft-edged circular patch with energy spread beyond the nominal radius. LiDAR beam divergence, unlike EO/IR, is narrower than sampling distance, leading to unavoidable subsampling artifacts.

- **EO/IR Pixel:** F_r is modeled as a quadrilateral (or ellipse) projected onto terrain from camera geometry. Physically, the pixel integrates light according to the optical point spread function (PSF), leading to Gaussian-like roll-off and overlap between adjacent pixels.

- **SAR Resolution Cell:** F_r is modeled as an oriented ellipse defined by range and azimuth resolution. The actual response includes sidelobes and speckle, which smear energy outside the ideal ellipse.

- **Acoustic/ELINT Sample:** F_r is modeled as a lobe or sector volume derived from beam pattern. True reception patterns often have side lobes and irregularities that extend sensitivity beyond the modeled volume.

Formally, the footprint can be described as:

$$\mathcal{F}_r = T_r(\Omega_0)$$

where Ω_0 is a canonical shape (circle, ellipse, quadrilateral, cube) and T_r is the sensor-specific transform (projection, warping, terrain interaction, atmospheric distortion).

The statistical variance term Σ_r then augments F_r by encoding the blurred, probabilistic distribution of energy around and beyond this idealized footprint.

43. Statistical Variance (Sphere of Confusion)

Each footprint[2] is associated with an uncertainty kernel Σ_r, typically a covariance matrix in $\mathbb{R}^3$, though it may be extended to $\mathbb{R}^4$ to also capture temporal uncertainty as well.
– **Isotropic noise:** $\Sigma_r = \sigma^2 I \rightarrow$ spherical circle of confusion.
– **Anisotropic distortion:** Σ_r elliptical, stretched along terrain slope or projection angle.
– **Atmospheric or propagation spread:** elongation of Σ_r along line-of-sight.
– **Spectral or radiometric variance:** additional components can be modeled for calibration error or noise in $\{\lambda_i\}$ or $\{a_j\}$.

The probability density of the true measurement location $\mathbf{q}$ is:

$$P(\mathbf{q}|\mathbf{p}, \Sigma_r) \propto \exp\left(-\tfrac{1}{2}(\mathbf{q}-\mathbf{p})^\top \Sigma_r^{-1}(\mathbf{q}-\mathbf{p})\right).$$

44. Composite Representation

A Vossel sample V_r is then defined as:

$$V_r = \left(\mathbf{p},\ \mathcal{F}_r,\ \Sigma_r,\ \{\lambda_i\},\ \{a_j\}\right)$$

Where:
– $\mathbf{p} = (x, y, z, t)$: nominal center
– F_r: projected geometry (footprint envelope)
– Σ_r : statistical variance kernel (uncertainty/confusion)
– $\{\lambda_i\}$: spectral/temporal attributes[3]
– $\{a_j\}$: sensor-specific metadata (intensity, phase, Doppler, etc.)

45. Interpretation

– **Projected Geometry** = how the sensor paints the world (physical footprint).
– **Statistical Variance** = confidence in the placement of that footprint in space, extended if needed into time and spectral dimensions[4,5].
– Together they define a probabilistic volumetric element (a Vossel) with both extent and noise.
– This abstraction allows heterogeneous sensors to be integrated consistently, with uncertainty explicitly represented in the same mathematical (and computer data structure) framework.

46. Full Form

Let's take the Vossel element data and combine it with voxel coherency to get our expanded voxel-band-time process:

For resolution level ℓ, voxel (i, j, k), spectral band b, and time bin τ , define

$$X^{(\ell)}_{i,j,k,b,\tau} = \frac{\sum_{n=1}^{N} \chi^{(\ell)}_{i,j,k}(n)\,\theta_\tau(n)\,\eta_b(n)\,z_{n,b}}{\sum_{n=1}^{N} \chi^{(\ell)}_{i,j,k}(n)\,\theta_\tau(n)\,\eta_b(n)}$$

And expand our notation to clarify implementation and show how spatial, temporal, and spectral weights arise from integrals over kernels and regions and eliminating any confusion over what χ, θ, η mean. As such, let's put into the expanded form:

$$X^{(\ell)}_{i,j,k,b,\tau} = \frac{\sum_{n=1}^{N} \underbrace{\left(\int_{V^{(\ell)}_{i,j,k}} \phi_n(\mathbf{q})\,1_\Omega(\mathbf{q})\,d\mathbf{q}\right)}_{\text{spatial occupancy } \chi^{(\ell)}_{i,j,k}(n)} \underbrace{\left(\int_{\mathcal{T}_\tau} \psi_n(t)\,dt\right)}_{\text{temporal occupancy } \theta_\tau(n)} \underbrace{\left(\int_{\Lambda_b} R_b(\lambda)\,d\mu_n(\lambda)\right)}_{\text{spectral mixing } \eta_b(n)} z_{n,b}}{\sum_{n=1}^{N} \left(\int_{V^{(\ell)}_{i,j,k}} \phi_n(\mathbf{q})\,1_\Omega(\mathbf{q})\,d\mathbf{q}\right) \left(\int_{\mathcal{T}_\tau} \psi_n(t)\,dt\right) \left(\int_{\Lambda_b} R_b(\lambda)\,d\mu_n(\lambda)\right)}$$

This traverses all voxels, bands, and time bins, and for each, sums over all elements with weights that incorporate spatial occupancy, temporal occupancy, spectral mixing, and mipmap additivity. The numerator aggregates all element contributions; the denominator normalizes by total occupancy weight.

47. Mipmap Consistency in the Weights

This ensures that aggregation across levels is mathematically consistent. The spatial occupancy $\chi^{(\ell)}$ is additive over its eight children in the octree, and the full weight $w^{(\ell)}$, when combined with temporal and spectral terms, inherits the same child→parent recursion. This means that parent voxels represent the exact sum of their children, not an approximation. By making this explicit, we guarantee that queries and fusion operations remain conservative across resolutions, and rein-forcement learning agents can transition between coarse and fine levels without losing consistency or violating conservation laws.

Spatial occupancy is additive across the octree:

$$\chi^{(\ell)}_{i,j,k}(n) = \sum_{\alpha,\beta,\gamma \in \{0,1\}} \chi^{(\ell-1)}_{2i+\alpha,\,2j+\beta,\,2k+\gamma}(n).$$

Therefore the full weight

$$w^{(\ell)}_{i,j,k,b,\tau}(n) = \chi^{(\ell)}_{i,j,k}(n)\,\theta_\tau(n)\,\eta_b(n)$$

inherits the same child→parent recursion:

$$w^{(\ell)}_{i,j,k,b,\tau}(n) = \sum_{\alpha,\beta,\gamma\in\{0,1\}} w^{(\ell-1)}_{2i+\alpha,\,2j+\beta,\,2k+\gamma,\,b,\tau}(n).$$

48. Global Conservation Identity

This guarantees that no information is lost as data are aggregated through the voxel–mipmap hierarchy. The weighted sums of voxel-level values $X^{(\ell)}$ reproduce the global total of all spectral measurements $z_{n,b}$, independent of resolution level ℓ. This ensures that coarse levels are exact redistributions of fine-level data, not approximations. By making this explicit, we preserve conservation across scales, enabling fusion, interpolation, and learning processes to operate without introducing artificial gain or loss in the underlying measurements.

With $W^{(\ell)}_{i,j,k,b,\tau} = \sum_n w^{(\ell)}_{i,j,k,b,\tau}(n)$,

$$\sum_{i=0}^{N_x-1}\sum_{j=0}^{N_y-1}\sum_{k=0}^{N_z-1}\sum_{b=1}^{B}\sum_{\tau} W^{(\ell)}_{i,j,k,b,\tau}\,X^{(\ell)}_{i,j,k,b,\tau} = \sum_{n=1}^{N}\sum_{b=1}^{B} z_{n,b}.$$

This holds at every level ℓ by additivity of the weights over the hierarchy.

49. Supporting Definitions

Spatial kernel (projected geometry + uncertainty)

$\phi_n(\mathbf{q}) \geq 0$ is a normalized 3D kernel encoding the element's projected footprint and uncertainty:

$$\int_{\mathbb{R}^3} \phi_n(\mathbf{q})\,d\mathbf{q} = 1.$$

Typical construction: pushforward of a canonical PSF through the sensor projection and terrain/medium mapping, with anisotropy governed by Σ_n.

50. Temporal Kernel

$\psi_n(t) \geq 0$ is a normalized 1D kernel for acquisition time and its uncertainty:

$$\int_{\mathbb{R}} \psi_n(t)\,dt = 1.$$

51. Spectral Mixing

$R_b(\lambda)$ is the bandpass (spectral response) of band b. μ_n is the element's spectral measure. The mixing coefficient is

$$\eta_b(n) = \int_{\Lambda_b} R_b(\lambda)\,d\mu_n(\lambda).$$

For band-isolated $z_{n,b}$, set $\eta_b(n) = 1$.

52. Selection Mask

$1_\Omega(\mathbf{q}) \in \{0, 1\}$ restricts to regions of interest (e.g., the view frustum). Omit or set to 1 for full-domain accumulation.

53. Variable and Index Recap

Elements and measurements

e_n: the n-th Vossel (element).
$\mathbf{p}_n = (x_n, y_n, z_n, t_n)$: nominal center of e_n.
F_n: projected footprint envelope of e_n.

Σ_n: uncertainty kernel parameters for e_n.
$z_{n,b}$: scalar value of e_n in band b (radiance, intensity, etc.).
$\mu_n(\lambda)$: spectral measure of e_n (discrete spectrum or density).

Spatial grid and hierarchy
$V_{i,j,k}^{(\ell)}$: voxel AABB at level ℓ and indices (i, j, k).
N_x, N_y, N_z : voxel counts per axis at base level.
ℓ: mipmap level, $\ell = 0$ finest; parent covers union of 8 children.

Temporal partition
$T_\tau = [\tau_0, \tau_1)$: time bin τ ; bins form a partition of the interval of interest.

Spectral partition
$b \in \{1,\dots, B\}$: spectral band index.
Λ_b: spectral support of band b.
$R_b(\lambda)$: spectral response of band b.

Kernels and weights
$\phi_n(\mathbf{q})$: spatial kernel (PSF + geometry) of e_n.
$\psi_n(t)$: temporal kernel of e_n.
$\eta b(n)$: spectral mixing coefficient of e_n into band b .

$\chi_{i,j,k}^{(\ell)}(n) = \int_{V_{i,j,k}^{(\ell)}} \phi_n(\mathbf{q}) 1_\Omega(\mathbf{q})\, d\mathbf{q}$: spatial occupancy.

$\theta_\tau(n) = \int_{T_\tau} \psi_n(t)\, dt$: temporal occupancy.

$w_{i,j,k,b,\tau}^{(\ell)}(n) = \chi_{i,j,k}^{(\ell)}(n)\, \theta_\tau(n)\, \eta_b(n)$: full occupancy weight.

$W_{i,j,k,b,\tau}^{(\ell)} = \sum_{n=1}^{N} w_{i,j,k,b,\tau}^{(\ell)}(n)$: total weight in a voxel–band–time bin.

Field variables
$X_{i,j,k,b,\tau}^{(\ell)}$: voxelized field value (weighted mean) at level ℓ, voxel (i, j, k), band b, time bin τ .

Selection
$1\Omega(\mathbf{q})$: optional spatial mask (e.g., view frustum).

54. Special Cases

Deterministic point elements
Set $\phi_n(\mathbf{q}) = \delta(\mathbf{q} - \mathbf{p}_n^{(xyz)})$ and $\psi_n(t) = \delta(t - t_n)$ to recover hard assignment. Then $\chi_{i,j,k}^{(\ell)}(n) \in \{0, 1\}$ and $\theta_\tau(n) \in \{0, 1\}$.

Mass accumulation instead of averages
Use the unnormalized numerator as a mass field; conservation across mip levels follows from the additive recursion of $w_{i,j,k,b,\tau}^{(\ell)}(n)$.

Non-Gaussian Kernels
Replace ϕ_n by an appropriate PSF (Airy, speckle-aware, multimodal) without changing the equations; normalization is preserved.

55. Sensors Considered

The sensors listed demonstrate the flexibility of the Vossel abstraction. A single atomic representation can cover imaging, radar, LiDAR, acoustic, seismic, magnetic, gravimetric, environmental, cyber, and jamming sources. Vossels are important because they preserve geometry and uncertainty while unifying different observables into one structure. All sensor feeds can be kept in one place and represented at their native resolution, even though the modalities vary widely in scale and character. This creates a centralized repository that allows radically different data types to coexist and be fused within a consistent world model.

Sensor / Data Class	Typical Subtypes / Examples	Primary Observables	Projected Geometry (Footprint)	Uncertainty Kernel Emphasis	VOSL Conversion Distinction
Acoustics (Air, Passive)	Single microphones, arrays	DOA, band power	Conical lobes / spheres	Wind advection, multipath	Frequency-dependent kernels; triangulate when multiple sensors
AIS / Cooperative Beacons	AIS, ADS-B, Mode-S, Blue Force trackers	Track reports	Point tracks with attributes	GNSS error	Correlated with ELINT/Hawkeye 360 to distinguish cooperative vs non-coop
Along-track Interferometric SAR (ATI)	Multi-channel SAR	Radial velocity	As SAR	Baseline and Doppler errors	Velocity bands with covariance; moving target flags
Aerial LiDAR	Pushbroom, scanning, bathymetric	Range, intensity	Conical beams intersect terrain, circular patches	Beam divergence, timing jitter	Multi-return pulses; swath footprints; DEM snapping
Atmospheric Weather Fields	Weather stations, NWP models, radiosondes	Wind speed, wind direction, temperature, humidity	3D grid fields	Sensor density, advection errors	Voxelized environmental layers; used for propagation corrections (acoustics, EO, radar)
Biological Aerosol Sensing	UV-LIF lidar	Fluorescence spectra	Range-sliced beams	Low SNR	Spectral bands; detection confidence attributes
Chemical Standoff (IR)	FTIR, hyperspectral	Gas absorption spectra	As EO pixel	Atmospheric distortions	Bands as gas features; retrieval metadata in attributes
COMINT	Communications intercepts	f_c, Δf, modulation	Rays, cones, or points	DOA/TDOA ambiguities	Spectral bands; demodulation metadata; triangulation lineage
Cyber / Network ISR	NOCs, cyberattack monitors	Latency, packet loss, intrusion alerts	Graph nodes & links mapped into space or logical topology	Asymmetry in node/path degradation	Vossels represent node health; attributes: throughput, latency, attack signature
DIAL / Raman Lidar	Differential absorption lidar	Column density	Range-resolved beams	Model error	Range-slice Vossels; on/off bands; inversion metadata

Distributed Acoustic Sensing (DAS)	Fiber-based strain sensing	Strain vs. distance	Along-fiber voxels	Coupling, noise	1D line of Vossels; broad lateral kernels
ELINT (incl. Hawkeye 360)	Radar intercepts, PRI, frequency; commercial RF geolocation	PRI, DOA, TDOA, FDOA, geolocated ellipses	Rays, cones, sectors, ellipses	Multipath, geometry ambiguity	Store emitter attributes; correlation with AIS possible
Environmental Meteorological Sensors	Towers, radiosondes	Wind, temperature, humidity	Points	Advection uncertainty	Attributes for propagation models
FOPEN Radar	UHF/VHF SAR	Backscatter through foliage	As SAR	Multipath, volumetric scattering	Larger cross-range variance; penetration flags
FMV (Full Motion Video)	EO/IR video	Pixels over time	As EO	Jitter, blur	Frame-wise Vossels; temporal smoothing
GNSS Reflectometry (GNSS-R)	GNSS bistatic	Surface reflectivity	Glint kernels	Geometry error	Loci footprints; sea/wetness indicators
Gravimetry	Airborne, satellite	Scalar gravity (mGal)	Very broad volumes	Drift, terrain	Coarse mip levels; terrain correction in attributes
Gravity Gradiometry	FTG, airborne	Tensor gradients	Broad footprints	Platform motion noise	Per-component bands; smoothing required
Ground Penetrating Radar (GPR)	Airborne, ground-based	Backscatter traces	Vertical traces aligned to terrain	Multipath, attenuation	SEGY traces converted to vertical Vossels; surface elevation corrections
GMTI/SMTI	Moving target indication radar	Detections, Doppler	Point ellipsoids	Range/Doppler/ angle errors	Point-like Vossels; velocity attributes; track linking
Hyperspectral ISR	VNIR/SWIR/ MWIR cubes	100–400 bands	As EO	Spectral misregistration, atmospheric absorption	Dense λ-vectors; optional PCA/ dimensionality reduction
Infrasound Arrays	Sub-20 Hz arrays	DOA, band energy	Large cones	Wind stratification	Very broad kernels; atmosphere model in attributes
InSAR	Two-pass, DInSAR	Phase, displacement	As SAR	Atmospheric phase, decorrelation	Height/ displacement bands; coherence attributes
ISAR (Inverse SAR)	Target-focused radar	Micro-Doppler images	Target-centric	Aspect, motion	Object-frame geometry; link to tracks

ISAR Micro-Doppler	Jet engine/rotor analysis	Micro-Doppler	Target-centric	Aspect uncertainty	Bands are micro-Doppler signatures; link to platform tracks
Jamming / EW Sources	Noise, barrage, smart jammers	Area of degraded SNR	Cones, lobes, isotropic spheres	Power distribution, multipath	Footprint = interference volume; attributes: jammer type, band, ERP
Laser Vibrometry	Coherent LDV	Surface vibration spectra	Surface patches	Speckle, angle sensitivity	Bands are vibration frequency bins; attach spectral displacement attributes
Lithology / Subsurface Materials	Geological survey, boreholes	Material type (sand, limestone, granite, etc.)	Volumetric cells tied to DEM depth	Mapping uncertainty, stratigraphic errors	Material type tags per voxel; used for propagation/ interaction models
Low-Light Imaging	EMCCD, ICCD, sCMOS	Radiance at high gain	As EO	Shot noise, motion blur	Radiometric variance captured; confidence weighting
Magnetometry	Scalar, vector magnetometers	nT fields	Broad regions	Diurnal variation, offsets	Baseline corrections; coarse kernels
Magnetic Anomaly Detection (MAD)	Airborne MAD systems	Magnetic dipole anomalies	Broad lobes	Platform field bias	Bands = magnetic residuals; motion compensation attributes
Magnetotellurics (MT)	MT, AMT systems	Impedance spectra	Very broad subsurface volumes	Natural field variance	Frequency-dependent kernels; long integration windows
Multispectral EO	5–30 bands	Band radiance	As EO	Band-dependent PSF, SNR	Band metadata per Vossel; per-band weights
Muon Tomography	Cosmic ray muon attenuation	Line integrals	Rays through volume	Sparse events	Voxelized path integrals; inversion metadata
Nuclear / Radiation ISR	Gamma/neutron detectors	Counts, spectra	Point + diffuse sphere	Shielding, background radiation	Energy-binned bands; dead-time correction; broad kernels
Oceanographic Fields	Buoys, ARGO floats, models	Salinity, temperature, currents	3D volumetric grids	Sparse sampling, advection	Voxelized water column attributes; used in sonar/ acoustic corrections

OTHR / HFSWR	HF surface/over-horizon radar	Surface currents, tracks	Broad sectors	Ionospheric refraction	Very large kernels; refraction attributes
Passive Millimeter-wave	30–300 GHz imagers	Brightness temperature	As EO	Large PSF, atmospheric effects	Radiometric noise emphasis; large elliptical footprints
Passive Sonar	Towed/fixed arrays	DOA, LOFARgrams	Rays/cones	SSP uncertainty	Rays/cones as Vossels; multipath hypotheses
Photogrammetry	Multi-view EO bundles	3D tie-points	Triangulated points	Tie-point error	Feature-Vossels; uncertainty from bundle adjustment
PolSAR	Quad-pol, compact-pol	Coherency matrices	As SAR	Channel cross-talk	Bands = matrix elements; Hermitian constraints
Radar (Marine Navigation)	X/S band	Surface backscatter	As SAR-lite	Sea state	Water-surface Vossels; clutter parameters
Resistivity (ERT)	Wenner, Schlumberger	Apparent ρ	Banana-shaped volume	Electrode placement	Multi-ellipsoid kernels; depth of investigation
SAR Imaging	Stripmap, Spotlight	Backscatter	Oriented resolution ellipses	Speckle, incidence angle	Backscatter Vossels; terrain projection
Seismic / UGS	Geophones, unattended ground sensors	Waveform amplitudes	Points at sensor nodes	Very large spatial uncertainty	Temporal precision; broad kernels; cross-node correlation
Sonar Imaging	Side-scan, MBES, SAS	Backscatter, bathymetry	Bottom ellipses / volumetric lobes	SSP refraction, grazing angle	Bottom snapping; water-column vs bottom attributes
Space Object Tracking	EO and radar space surveillance	Angles, range, Doppler	Track points	Orbit fit uncertainty	Track chains; covariance propagation
Space Weather Fields	Solar flux monitors, magnetometers	Solar wind, geomagnetic indices	Broad space-time volumes	Model/forecast error	Affects HF radar, satellites; stored as environmental grids
Space-based IR Early Warning	SBIRS-like systems	Brightness events	Large pixels	Cloud cover, atmosphere	Event Vossels with large kernels
Standard EO Cameras	Consumer, mapping	RGB pixels	Pixel quads projected onto terrain	DEM height, PSF	Per-pixel Vossels; radiometry metadata
SWIR Imaging	0.9–1.7 µm imagers	Radiance	As EO	Water vapor absorption	Bands = SWIR; atmospheric model attributes

Terahertz Imaging	0.3–10 THz imagers	Radiance, spectral features	As EO	Attenuation, multipath	Near-field footprints; short range
Terrestrial LiDAR (TLS)	Tripod, polar scan	Range, reflectance	Polar sampling geometry	Occlusion, registration	Very high density; station metadata
Thermal ISR (MWIR/LWIR)	Airborne pods, ground imagers	Radiance / brightness temp	As EO	Emissivity, atmos attenuation	Add σ_T; calibration drift attributes
Through-wall Radar	UWB imaging	Range profiles	Wall-aligned ellipses	Multipath	Ellipsoids behind walls; building model attributes
Tomographic SAR (TomoSAR)	Multi-baseline stacks	3D reflectivity volumes	Elliptical stacks	Vertical resolution limits	Bands per altitude bin; sparsity priors
UWB Localization	Impulse radio, RF ranging	Ranging/TDOA	Spheres/shells	Clock sync	Iso-range shells; multi-node intersection
WAMI (Wide Area Motion Imagery)	Persistent EO	Coarse pixels, high frame rate	Large projected quads	Registration error	Tight temporal bins; motion emphasis
Wi-Fi/5G Sensing	CSI, device-free localization	Range/angle changes	Volumetric near nodes	Multipath	Small lobes; anomalies mapped as Vossels

56. Clarifications
Clarifications are necessary since the terminology employed here overlaps with conventional computer graphics. Voxels may either hold data values directly or serve as spatial organizers with pointers. In medical imaging, voxels typically contain measured values, whereas in ray tracing they are used to spatially organize geometry for rapid culling. Confusion arises when both interpretations are used interchangeably. Another challenge of voxel-based processing is adaptive resolution, often referred to as the “teapot in the stadium problem.” Vossels address this issue through mipvols, which provide scalable multiresolution structures. It should also be noted that voxels are not restricted to Cartesian space; they may occupy spherical, Fourier, or octahedral domains, among others. Recent work by Chan and Haldar demonstrates the use of voxels in k-space for medical imaging, a trend that applies here as well [9]. Vossels can be extended into temporary or permanent alternate domains as required, maintaining consistency of representation across coordinate systems.

57. Ongoing Work
Current work is focusing on the detailed mechanics of Vossel implementation. The mathematical and structural framework established here is being extended into full integration processes. Efforts are underway to construct voxel hierarchies from raw sensor feeds, including projection, kernel integration, uncertainty propagation, and occupancy weighting. Retrieval procedures are being demonstrated, showing how spatio–temporal–spectral queries are resolved using direct kriging, hybrid voxel–kriging, and mipmap-based approximations.

Research is also addressing ingestion of ISR modalities—ELINT, EO/IR, SAR, LiDAR, acoustic, seismic, environmental, cyber, and jamming—into the Vossel model. This includes transforms from native data formats to Vossel-compatible representations, with emphasis on preserving geometry, uncertainty, and spectral content. Localized spherical voxel spaces are more attuned to sensor polar resolutions, and for simulation, Plücker Space is convenient in ray intersection tests [10]. In parallel, the design is being developed with an eye toward hardware acceleration, particularly through FPGA-based prototypes and potential ASIC implementations. Embedding Vossel ingestion and fusion directly into hardware reduces the computational resources required, making it feasible for in-field edge use where size, weight, and power constraints limit conventional processing.

System-level integration is also being advanced, including view-frustum extraction pipelines for interactive visualization, distributed storage for large ISR data streams, and optimization of mipmap consistency across networked systems. The aim is to make the Vossel framework directly usable in ISR fusion engines, simulation environments, and operational planning tools.

These activities are moving the emphasis from theoretical formulation to reproducible implementation, establishing precise algorithms, benchmarks, and hardware pathways so that Vossel-based fusion can be deployed and evaluated in live ISR environments.

References

1. Cerdeira-Pena, A., de Bernardo, G., Fariña, A., Paramá, J. R., & Silva-Coira, F. (2018, September). Towards a compact representation of temporal rasters. In *International Symposium on String Processing and Information Retrieval* (pp. 117-130). Cham: Springer International Publishing.
2. Abdelguerfi, M., Givaudan, J., Shaw, K., & Ladner, R. (2002, November). The 2-3TR-tree, a trajectory-oriented index structure for fully evolving valid-time spatio-temporal datasets. In *Proceedings of the 10th ACM international symposium on Advances in geographic information systems* (pp. 29-34).
3. Forlizzi, L., Güting, R. H., Nardelli, E., & Schneider, M. (2000). A data model and data structures for moving objects databases. *ACM SIGMOD Record, 29*(2), 319-330.
4. Nishida, S., Saiwaki, N., & Naka, A. (2002, November). An adaptive spatio-temporal data management structure for efficient search. In *Proceedings 6th IEEE International Workshop on Robot and Human Communication. RO-MAN'97 SENDAI.* IEEE.
5. Reddy, S., Kothapalli, A., Varshini, K. S. S., Nandhan, R., & KV, V. (2024, November). Space Debris Clustering and Avoidance Using Chrono-Spatial Computational Frameworks. In *2024 International Conference on Intelligent Computing and Emerging Communication Technologies (ICEC)* (pp. 1-6). IEEE.
6. Wang, Y., Hijikata, Y., & Nishida, S. (2003, March). A study on a spatio-temporal data structure for managing video data from monitoring cameras. In *17th International Conference on Advanced Information Networking and Applications, 2003. AINA 2003*. (pp. 454-461). IEEE.
7. Zhang Lifu, Chen Hao, Sun Xuejian, Fu Dongjie, & Tong Qingxi. (2017). Design of integrated storage structure for multi-dimensional remote sensing data in terms of space, time and spectrum. Journal of Remote Sensing, 21(1).
8. Samet, H. (2006). *Foundations of multidimensional and metric data structures.* Morgan Kaufmann.
9. Chan, C. C., & Haldar, J. P. (2025). A New k-Space Model for Non-Cartesian Fourier Imaging. *arXiv preprint arXiv:2505.05647*.
10. Marmitt, G., & Slusallek, P. (2005). Fast ray traversal of unstructured volume data using plucker tests. *Computer Graphics Lab, Saarland University. Technical report.*