International Journal of Advanced Engineering Research and Science (IJAERS)
Peer-Reviewed Journal
ISSN: 2349-6495(P) | 2456-1908(O)
Vol-13, Issue-8; Aug, 2026
Journal Home Page Available: https://ijaers.com/
Article DOI: https://dx.doi.org/10.22161/ijaers.138.3

A Grid-Based Spatial Representation of Logistics Warehouses Using 3rd-Order Tensors

Zine El Abidine Falouti

Independent Researcher and e-Logistics Lecturer
Email: zine.falouti@gmail.com

Received: 05 Jul 2026,
Received in revised form: 03 Aug 2026,
Accepted: 07 Aug 2026,
Available online: 12 Aug 2026

Keywords*— *Warehouse Spatial Modeling, Third-Order Tensors, Spatial Discretization, Logistics Automation, Smart Warehousing.

***Abstract*— *The digital representation of a logistics warehouse within Enterprise Resource Planning (ERP) systems, particularly Warehouse Management Systems (WMS), is mainly based on operational data and warehouse activities. While these systems help improve inventory management, reduce operational costs, shorten delivery times, and increase distribution efficiency, they do not provide a mathematical representation of the warehouse's physical space. As a result, researchers have limited support for developing and testing optimization methods that depend on the spatial organization of the warehouse. This research proposes a mathematical three-dimensional (3D) grid model in which each storage cell is represented by its spatial coordinates and an associated weight. The proposed model provides a simple and generic digital representation of a logistics warehouse that can be used as a basis for research on warehouse optimization, including storage allocation, routing, robot navigation, and space utilization.*

I. INTRODUCTION

1.1 Background

Warehouse automation has become an increasingly important objective for manufacturers, logistics providers, and distribution centers as they respond to the rapid growth of e-commerce and the increasing global demand for goods. Robotics and intelligent systems are playing a central role in this transformation, requiring accurate digital representations of warehouse environments before deployment. The adoption of these technologies is driven by rising customer expectations, shorter delivery times, and stronger global competition (Winkelhaus & Grosse, 2020) [1]. Although many organizations have invested in warehouse automation, their primary concerns have traditionally focused on the trade-offs between implementation cost, service levels, and operational flexibility, as reported in a survey of 27 warehouse automation projects in the United Kingdom (Baker & Halim, 2007) [2]. More recently, Autonomous Mobile Robots (AMRs) have emerged as a key technology in smart warehouses, supporting navigation, material handling, and real-time operational decision-making (Keith & La, 2024) [3]. As warehouse automation continues to evolve and new optimization methods are developed, there is an increasing need for a mathematical spatial representation of warehouse environments that provides researchers with a common foundation for designing and evaluating optimization algorithms.

1.2 Research Gap

The digital representation of a logistics warehouse in current Enterprise Resource Planning (ERP) and Warehouse Management System (WMS) platforms extends beyond inventory management to include inbound and outbound operations, material and information flows, quality management, financial processes, human resources, and equipment management. These systems provide a comprehensive operational view of warehouse activities and support day-to-day business decisions. The physical organization of the warehouse is typically described through logical structures such as shelves, aisles, storage locations, and functional zones, which are sufficient for managing warehouse operations. However, these representations are not intended to provide a generic mathematical description of the warehouse space. As a result, spatial relationships between storage locations are often implicit rather than explicitly defined through a unified coordinate-based model. Consequently, researchers developing optimization algorithms frequently need to construct their own spatial representations before addressing problems such as storage allocation, robot navigation, path planning, and space utilization. This limitation highlights the need for a generic mathematical spatial model that can serve as a common foundation for research in warehouse optimization.

1.3 Research Objectives

The primary objective of this research is to develop a mathematical representation of logistics warehouse physical space through a grid-based spatial modeling approach. To achieve this objective, the research aims to:

• Define a continuous mathematical representation of a logistics warehouse domain and its spatial boundaries.

• Transform the physical warehouse space into a finite three-dimensional grid structure through spatial discretization.

• Establish a formal representation of individual warehouse cells using spatial indices and coordinate mappings.

• Organize discrete spatial cells into matrix-based layers and construct a third-order tensor representation of the complete warehouse environment.

• Provide a structured mathematical framework capable of associating spatial cells with additional attributes through cell-level weighting.

II. LITERATURE REVIEW

2.1 Warehouse Representation

Covering more than three decades of research on Automated Storage and Retrieval Systems (AS/RS), the literature review by (Roodbergen, K. J., & Vis, I. F. A., 2009) [4] examined the major system classifications and their variations, including shuttle mechanisms, rack systems, storage layouts, and order-picking configurations. The review organized the literature into two principal research areas: physical design and control policies. Physical design focused on system characteristics such as the number of aisles, rack configurations, and storage capacity, while control policies addressed storage assignment, order batching, retrieval sequencing, and dwell-point positioning for idle cranes. Furthermore, [4] highlighted the growing need for dynamic warehouse models capable of meeting modern industrial requirements while reducing computational time. Although the reviewed studies provide effective models for warehouse design and operational optimization, they primarily represent warehouse environments through application-specific layouts and control strategies rather than through a generic mathematical spatial representation.

Driven by the rapid growth of e-commerce and logistics demand, traditional warehouses are transitioning into smart warehouses equipped with advanced technologies such as the Internet of Things (IoT), robotics, and automated systems. (Zhen & Li, 2022) [5] presented a comprehensive framework for categorizing research on smart warehouse operations based on four pillars: information interconnection, equipment automation, process integration, and environmental sustainability. Their review highlights automation as a key component of smart warehouse development, supported by the integration of digital information and intelligent systems. As warehouse automation continues to expand, these systems increasingly rely on accurate and structured data to support autonomous operation. In particular, spatial information has become essential for applications such as robot navigation, routing, and warehouse optimization, as also discussed by [3] and supported by the evolution of Logistics 4.0 presented by [2].

Current warehouse management systems represent warehouse environments through logical entities such as storage locations, shelves, racks, aisles, and operational zones. These representations are well suited for inventory management, order fulfillment, and warehouse control but are primarily designed to support operational processes rather than provide a generic mathematical description of the warehouse space. Consequently, spatial information is commonly embedded within application-specific layouts instead of being formalized as a reusable mathematical model.

2.2 Spatial Modeling

Converting spatial coordinates into structured mathematical matrices is a spatial modeling approach (Dray et al., 2006) [6] based on the analysis of spatial relationships through methods such as Principal Coordinates of Neighbour Matrices (PCNM) and Moran's Eigenvector Maps (MEM). The spatial representation begins with a set of (n) sampling locations, from which a pairwise Euclidean distance matrix is constructed:

$\boldsymbol{D} = [\boldsymbol{dij}]$ (1)

where $\boldsymbol{dij}$ represents the Euclidean distance between locations ***(i)*** and ***(j)***, resulting in an $(\boldsymbol{n} \times \boldsymbol{n})$ distance matrix. To emphasize local spatial relationships while limiting the influence of distant locations, a truncated distance matrix ($\boldsymbol{D*}$) is constructed using a predefined threshold distance ***(t)***:

$$\boldsymbol{d*ij} = \begin{cases} \boldsymbol{dij \ if \ dij \ \leq t} \\ \boldsymbol{4t \ if \ dij \ > t} \end{cases} \quad (2)$$

The resulting matrix preserves spatial connectivity among neighboring locations while reducing the contribution of distant spatial relationships.

While [6] primarily focuses on two-dimensional spatial representations through $(\mathbf{n} \times \mathbf{n})$ spatial relationship matrices for a single spatial structure, the underlying matrix-based representation can be extended to higher-order data structures when additional dimensions are introduced. For example, incorporating a temporal dimension transforms a spatial matrix into a third-order spatio-temporal tensor with dimensions ($\boldsymbol{n} \times \boldsymbol{n} \times \boldsymbol{T}$), where ($\boldsymbol{T}$) represents multiple time observations. This extension illustrates how spatial relationships represented as matrices can be integrated into higher-dimensional mathematical structures for modeling complex spatial systems.

Although [6] successfully transforms spatial relationships using distance matrices, its primary objective is statistical pattern analysis rather than building an explicit geometric environment. In contrast, our proposed framework defines the physical space itself as a discrete grid structure, representing two-dimensional planes as matrices and stacking them to construct higher-order tensors. This direct spatial mapping allows neighborhood relationships and routing distances between computational cells to be natively derived from the data structure, directly supporting future automation applications such as storage optimization, path planning, and autonomous robot navigation.

In the Theory of Cellular Automata (von Neumann & Burks, 1966) [7] the key principles of spatial representation are a discretization of space (The Lattice Grid), cell states as local material properties, local spatial interactions (neighborhoods), spatial homogeneity & translation invariance, emergence of motion and structure in cellular space, and finally a space as material and computation environment. Overall, [7] presents a space as a discrete, structured grid of elementary units called cells.

Space representation by Cellular Automata:

$$\boldsymbol{G} = \{\boldsymbol{c1}, \boldsymbol{c2}, \boldsymbol{c3}, \dots, \boldsymbol{cn}\} \quad (3)$$

The lattice grid in Cellular Automata has a very specific meaning. It is not just a visual grid; it is the underlying mathematical space where the cells of the automaton exist.

Instead of a continuous domain:

$$\Omega \subset \mathbb{R}^2 \quad (4)$$

it defines space as a set of discrete positions:

$$\boldsymbol{L} = \mathbb{Z}^{d} \quad (5)$$

For example, a 2D lattice:

$$\boldsymbol{L} = \{(\boldsymbol{i}, \boldsymbol{j}) : \boldsymbol{i}, \boldsymbol{j} \in \boldsymbol{Z}\} \quad (6)$$

Meanwhile a cellular automaton defines a state function:

$$\boldsymbol{s} : \boldsymbol{L} \rightarrow \boldsymbol{S} \quad (7)$$

Meaning that each position in the lattice has a state.

Example:

$$\boldsymbol{s}(\boldsymbol{i}, \boldsymbol{j}) \quad (8)$$

And the space becomes:

$$\begin{bmatrix} \boldsymbol{s11} & \boldsymbol{s12} & \boldsymbol{s13} \\ \boldsymbol{s21} & \boldsymbol{s22} & \boldsymbol{s23} \\ \boldsymbol{s31} & \boldsymbol{s32} & \boldsymbol{s33} \end{bmatrix} \quad (9)$$

In the framework proposed by [7], a continuous environment is abstracted into a lattice composed of elementary cells, where each cell has a defined position, state, and relationship with neighboring cells. While Cellular Automata primarily focus on the evolution of cell states through local interaction rules, the spatial discretization principle is relevant to the objective of this research. Similarly, this work represents a physical environment as a structured grid of cells, where each cell is defined by its spatial coordinates and associated attributes. However, instead of modeling dynamic state transitions, the proposed framework focuses on establishing a mathematical spatial representation in which two-dimensional grids can be expressed as matrices and extended into higher-dimensional structures through stacked matrices forming third-order or Nth-order tensors.

Volumetric cells (voxels) provide a spatial representation approach that is closer to the objective of this research, as the proposed framework aims to represent a logistics warehouse as a three-dimensional tensor structure. (Kaufman and Shimony, 1987) [8] introduced voxel-based representations in which continuous three-dimensional geometric objects are converted into discrete volumetric cells. A voxel can be represented by its spatial coordinates ***(i,j,k)***, where each elementary 3D unit occupies a defined volume and may contain an associated value or state. Their work presented a set of three-dimensional scan-conversion algorithms designed to transform continuous geometric primitives into discrete voxel-map representations stored within a Cubic Frame Buffer (CFB). These algorithms were developed as part of the CUBE Architecture; a hardware system designed for voxel-based 3D graphics. The proposed scan-conversion methods demonstrated the transformation of different geometric structures, including linear and planar primitives, curves, surfaces, and quadratic objects, into discrete volumetric representations.

[8] also emphasized the distinction between the spatial position of a voxel and the information stored within that

voxel. Their voxel-based algorithms separate the identification of voxel locations through positional information (VOXEL_POS) from the assignment of voxel attributes through writing operations (WRITE_VOXEL), where parameters such as color are associated with the generated volumetric cells. This separation between geometric representation and voxel attributes is relevant to the proposed framework, where each warehouse cell is defined not only by its spatial coordinates ***(i,j,k)*** but also by associated attributes or weights. Therefore, the voxel concept provides a foundation for representing a physical environment as a structured 3D space while allowing additional information to be attached to each spatial element.

2.3 Warehouse Structure

According to the warehouse design framework proposed by (Rouwenhorst et al., 2000) [9], the physical structure of a warehouse is determined through a series of strategic and tactical design decisions that aim to balance investment costs, storage capacity, and operational throughput. At the strategic level, these decisions include the selection of storage units and containers, storage systems, forward and reserve storage areas, and material handling and picking equipment. Together, these elements define the physical organization and spatial configuration of the warehouse.

Furthermore, [9] identified several physical constraints that influence warehouse design, including building dimensions, ceiling clearance, floor load capacity, and available floor space. The framework also highlights important trade-offs between storage capacity and throughput, as well as between investment cost and the level of automation. These physical characteristics and constraints define the geometric boundaries within which warehouse layouts are designed and provide the basis for constructing a mathematical spatial representation of the warehouse environment.

Warehouse layout design primarily focuses on the physical organization of warehouse facilities, including the positioning of loading docks, reception areas, storage locations, racking systems, rack heights, and slot dimensions (Albert et al., 2023) [10]. At its core, warehouse layout encompasses structural decisions regarding aisle and lane configurations, storage areas, facility organization, and department layouts, which collectively define the geometric arrangement of the warehouse environment. The literature also describes a variety of physical layout configurations, including conventional warehouse layouts, fishbone designs, cross-docking facilities, and different storage allocation strategies such as dedicated, shared, class-based, and random storage. Furthermore, warehouse structure is influenced by practical design considerations including rack dimensions, aisle widths, storage capacity, accessibility, and the integration of material handling equipment. Together, these physical elements establish the structural framework of a warehouse and define the spatial environment that can subsequently be represented through a mathematical grid-based model.

III. PROPOSED GRID-BASED SPATIAL REPRESENTATION FRAMEWORK

3.1 Physical Warehouse Space

A logistics warehouse physical space consists of multiple structural components, including storage locations, loading and unloading docks, and reception areas [10]. More detailed structural descriptions emphasize the importance of spatial decisions, such as dock positioning, storage stack height, and the selection of rack types and materials (Mohamud et al., 2023) [11]. From a global perspective, a storage area may be divided into two main zones: the reserve area, where products are stored in the most economical way, and the forward area, where products are positioned for efficient retrieval [9]. Additional spatial zones, such as pre-rack areas, are assigned to support warehouse organization and operational flow (ten Hompel & Schmidt, 2007) [12].

In a two-dimensional projection of the physical warehouse space (bird's-eye view), the layout can be represented as an arrangement of human interaction areas, storage areas, pre-docking zones, and reception areas. Furthermore, modern warehouse configurations, such as cross-docking structures, influence the spatial organization by modifying the relationship between receiving, storage, and shipping areas [12]. Therefore, a warehouse can be considered as a structured spatial domain composed of interconnected regions, providing the foundation for its transformation into a formal mathematical representation.

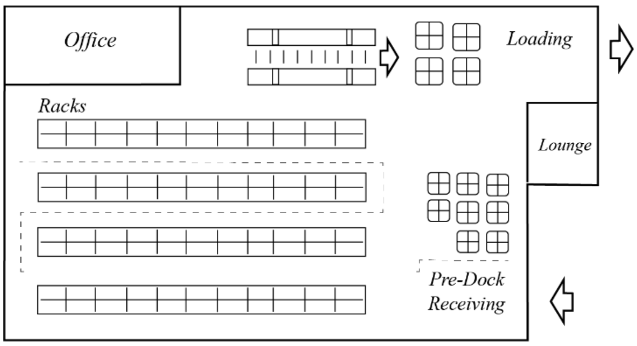

Fig. 1: Two-dimensional physical layout configuration of an example warehouse facility.

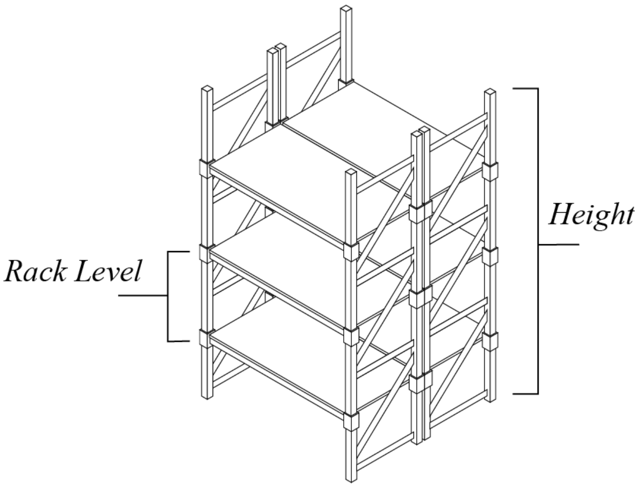

Fig. 2: Three-dimensional physical configuration of a multi-tier selective storage rack unit

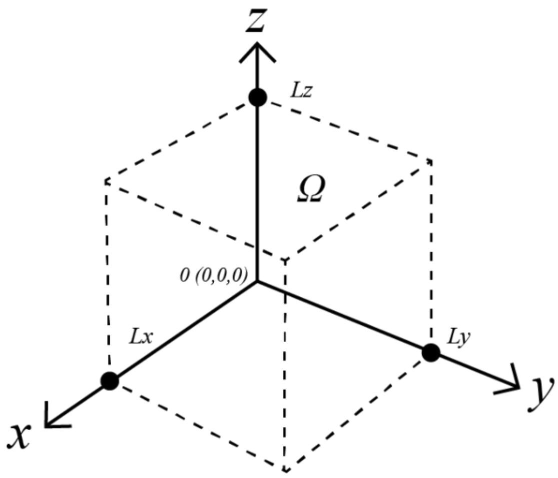

Fig. 3: Three-dimensional spatial domain Ω *and geometric boundary parameters.*

3.2 Spatial Discretization

Continuous physical domains must be transformed into discrete finite representations to enable numerical computation and algorithmic processing (Zienkiewicz et al., 2013) [13]. Accordingly, the warehouse physical space is modeled as a bounded three-dimensional domain:

$$\Omega = [\mathbf{0}, \mathbf{Lx}] \times [\mathbf{0}, \mathbf{Ly}] \times [\mathbf{0}, \mathbf{Lz}] \quad (10)$$

where L_x, L_y, and L_z represent the warehouse dimensions along the longitudinal, transversal, and vertical axes, respectively. This continuous domain is subsequently partitioned into discrete spatial elements through the definition of elementary intervals Δ_x, Δ_y, Δ_z representing the discretization resolution along each spatial axis.

The physical warehouse space Ω is defined by geometric boundaries along the three spatial axes, originating from the reference point (0,0,0). Unlike irregular or organic domains that may require approximation techniques for boundary representation in numerical methods, a logistics warehouse generally exhibits a structured geometric configuration. Therefore, the spatial limits can be directly defined by L_x, L_y, and L_z where these parameters represent the maximum extent of the domain along each axis. This bounded representation allows the continuous warehouse space to be discretized into finite spatial elements [13].

3.3 Grid Construction

As Ω was defined by its spatial bounds (10), the purpose of grid construction is to transform the continuous warehouse domain into a finite and quantifiable set of discrete spatial cells [13], (LeVeque, 2007) [14], and (Samet, 2006) [15].

The horizontal subdivisions of the warehouse domain are defined as:

$$\boldsymbol{Nx} = \frac{\boldsymbol{Lx}}{\boldsymbol{\Delta x}}, \ \boldsymbol{Ny} = \frac{\boldsymbol{Ly}}{\boldsymbol{\Delta y}} \quad (11)$$

where *Nx* and *Ny* denote the number of discrete grid cells along the longitudinal (x) and transversal (y) axes, respectively. The parameters Δx and Δy define the spatial resolution of the grid along the horizontal plane, where smaller values provide a finer spatial representation.

Unlike the horizontal dimensions, the vertical dimension is represented by the discrete structural organization of warehouse rack levels. Therefore, the vertical positions are defined as:

$$\boldsymbol{Z} = \{\boldsymbol{z_1}, \boldsymbol{z_2}, \dots \boldsymbol{z_{Nz}}\}$$

where:

$$\boldsymbol{N_z} = |\boldsymbol{Z}|$$

represents the total number of available vertical rack levels within the warehouse structure.

The total number of discrete spatial cells composing the three-dimensional representation is then defined as:

$$\boldsymbol{N(total)} = \boldsymbol{Nx} \times \boldsymbol{Ny} \times \boldsymbol{Nz} \quad (12)$$

Equation (12) quantifies the total number of individual cells or elements comprising the spatial mesh, establishing the structural finite baseline for subsequent mapping and tensor operations.

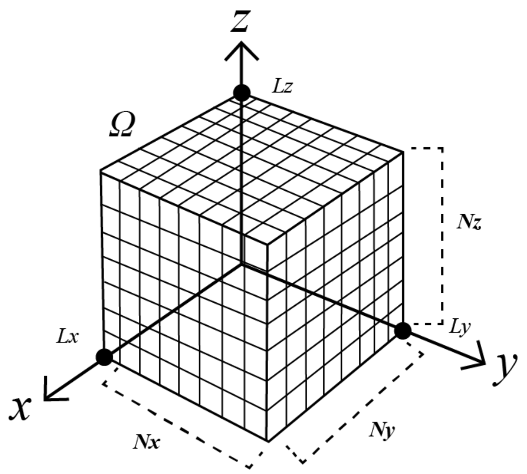

Fig. 4: Discretized three-dimensional grid mesh configuration illustrating the structural subdivision of the continuous domain Ω into a finite set of discrete spatial cells defined by intervals Eq. (11).

3.4 Matrix Representation

Following the structural grid construction, each discrete spatial cell is assigned an addressable index to enable its precise localization within the partitioned domain $\boldsymbol{\Omega}$ [14]. Each cell of the grid is represented as an indexed spatial element:

$\boldsymbol{C_{ijk}} = \boldsymbol{(x_i, y_j, z_k)}$ (13)

where $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ represent the discrete integer indices that denote the position of the cell along the longitudinal, transversal, and vertical axes, respectively. These indices are bounded by the total directional subdivisions of the grid framework such that:

$\boldsymbol{i} \in \{\boldsymbol{1}, \boldsymbol{2}, \dots, \boldsymbol{Nx}\}$

$\boldsymbol{j} \in \{\boldsymbol{1}, \boldsymbol{2}, \dots, \boldsymbol{Ny}\}$

$\boldsymbol{k} \in \{\boldsymbol{1}, \boldsymbol{2}, \dots, \boldsymbol{Nz}\}$

The mapping vector $(\mathbf{xi}, \mathbf{yj}, \mathbf{zk}) \in \mathbb{R}^{\mathbf{3}}$ defines the exact spatial midpoint coordinates of the discrete element $\mathrm{C_{ijk}}$ within the continuous Cartesian domain Ω, establishing a direct relationship between physical coordinate values and their corresponding discrete matrix positions.

For a fixed vertical index ***(k)***, the discrete spatial cells are organized into a two-dimensional matrix, a mathematical structure commonly used to represent ordered data in rows and columns (Strang, 2016) [16]. The matrix preserves the spatial arrangement of the cells along the longitudinal and transversal directions while maintaining a constant vertical coordinate. Accordingly, the matrix representation of the $\boldsymbol{k^{th}}$ warehouse level is defined as:

$$\boldsymbol{M_k} = \begin{bmatrix} \boldsymbol{C_{11k}} & \boldsymbol{C_{12k}} & \cdots & \boldsymbol{C_{1N_yk}} \\ \boldsymbol{C_{21k}} & \boldsymbol{C_{22k}} & \cdots & \boldsymbol{C_{2N_yk}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C_{N_x1k}} & \boldsymbol{C_{N_x2k}} & \cdots & \boldsymbol{C_{N_xN_yk}} \end{bmatrix} \quad (14)$$

$$\boldsymbol{M_k} \in \mathbb{R}^{\boldsymbol{N_x \times N_y}}$$

where M_k represents the matrix corresponding to the k^{th} vertical layer and contains N_x rows and N_y columns.

The rows of *Mk* correspond to the longitudinal (x) axis, while the columns correspond to the transversal (y) axis. The fixed index k identifies the vertical warehouse level. Consequently, each matrix *Mk* represents a complete two-dimensional spatial slice of the warehouse, preserving the positional relationships among all discrete cells within that level.

3.5 Third Order Tensor Representation

To unify the individual two-dimensional spatial slices *Mk* into a single, global mathematical structure, the discretized warehouse environment is modeled as a third-order tensor. In multilinear algebra, higher-dimensional arrays are formally defined as tensors, which extend the concepts of scalars, vectors, and matrices to an arbitrary number of dimensions (Kolda & Bader, 2009) [17]. By stacking the complete sequence of horizontal matrix layers along the vertical mode, the entire physical volume of the facility is encapsulated within a global spatial tensor $\boldsymbol{X}$:

$\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{N_x \times N_y \times N_z}}$ (15)

The tensor $\boldsymbol{X}$ is formed by the frontal slices corresponding to each vertical level *k*, such that:

$\boldsymbol{X_{:,:,k}} = \boldsymbol{M_k}, \ \forall \boldsymbol{k} \in \{\boldsymbol{1}, \boldsymbol{2}, \cdots, \boldsymbol{N_z}\}$ (16)

where $(\boldsymbol{X_{:,:,k}})$ denotes the ***k-th*** frontal slice of the tensor along mode-3. Consequently, any specific element within this tensor is addressed by its multi-index coordinate as $(\boldsymbol{X_{ijk}} = \mathbf{C_{ijk}})$, where the indices $\boldsymbol{i}, \boldsymbol{j}$, and $\boldsymbol{k}$ correspond to the longitudinal position, transversal position, and vertical level of the physical grid structure.

This algebraic framework provides an explicit, discrete indexed representation where structural spatial relationships are preserved computationally. By abstracting the physical facility layout into a third-order tensor, the grid-based model can be directly integrated into optimization pipelines, serving as a structured data canvas for multidimensional algorithms [17].

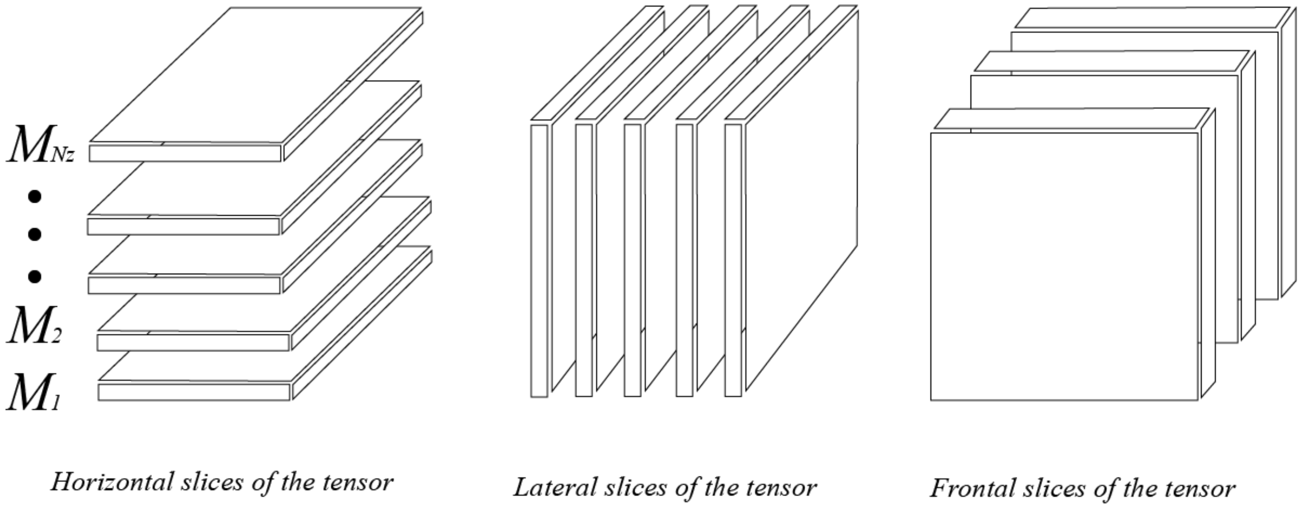

Fig. 5: Geometric slicing configurations of a third-order tensor, where the horizontal decomposition mode (left) physically instantiates the sequential vertical stacking of discrete warehouse floor levels and storage tiers Mk into the global computational domain X.

3.6 Weight Assignment

To transform the spatial tensor $\boldsymbol{X}$ from a purely geometric representation into a semantic spatial structure, each discrete cell is assigned a scalar weight representing a selected operational attribute associated with that location. In logistics applications, such attributes may represent inventory importance, demand intensity, occupancy, accessibility, or other operational indicators. The assignment of storage locations to products is commonly studied through the Storage Location Assignment Problem (SLAP), where spatial decisions influence retrieval efficiency and operational performance (Accorsi et al., 2014) [18].

The entries of the tensor are populated according to a weighted representation where each spatial cell $\boldsymbol{C_{ijk}}$ receives a value:

$$\boldsymbol{X_{ijk}} = \boldsymbol{\omega}(\boldsymbol{C_{ijk}}) \cdot \boldsymbol{\gamma}(\boldsymbol{k}) \quad (17)$$

Where $\boldsymbol{\omega}(\boldsymbol{Cijk}) \geq \boldsymbol{0}$ represents the base weight associated with the content or importance of the spatial cell, while:

$$\boldsymbol{\gamma}(\boldsymbol{k}) \in [\boldsymbol{0}, \boldsymbol{1}]$$

represents a vertical weighting modifier associated with the rack level *k*.

The parameter $\boldsymbol{\omega}(\boldsymbol{Cijk})$ may be derived from inventory-related indicators such as SKU demand frequency or turnover rate using classification approaches such as ABC analysis (Accorsi et al., 2014) [18]. $\boldsymbol{\gamma}(\boldsymbol{k})$ introduces the influence of the vertical position within multi-level storage structures. For accessibility-oriented applications, lower rack levels generally provide improved ergonomics and reduced retrieval effort, while higher levels may introduce additional handling constraints (Derhami et al., 2020) [19]. Therefore, $\boldsymbol{\gamma}(\boldsymbol{k})$ can be configured according to the operational characteristics of the warehouse environment.

Cells representing empty or non-operational locations may be assigned a null value:

$$\boldsymbol{Xijk} = \boldsymbol{0}$$

This weighted formulation extends the tensor representation from a geometric spatial model into a structured data representation capable of supporting future analytical and optimization approaches.

The selection of a multiplicative formulation in Equation (17) over an additive or linear structure is mathematically justified through sensitivity and boundary state analysis (Saltelli et al., 2008) [20]. Because the vertical modifier operates on a closed interval $\boldsymbol{\gamma}(\boldsymbol{k}) \in [\boldsymbol{0}, \boldsymbol{1}]$, it functions as an operational accessibility discount factor that scales the base stock priority. Under boundary conditions, if a specific vertical tier is rendered entirely non-traversable or structurally compromised due to equipment or safety constraints ($\boldsymbol{\gamma}(\boldsymbol{k}) = \boldsymbol{0}$), the multiplicative structure correctly forces the entire local spatial value to a null state ($\boldsymbol{X_{ijk}} = \boldsymbol{0}$), accurately reflecting zero operational utility. Conversely, a conventional linear additive framework (e.g., $\boldsymbol{\omega} + \boldsymbol{\gamma}$) would erroneously preserve a positive priority index regardless of accessibility limits, causing critical modeling failures in downstream routing and location allocation algorithms (Borgonovo, 2017) [21]. This multiplicative coupling ensures that the global tensor maintains a strict non-linear sensitivity to coordinate boundaries, preserving absolute mathematical continuity between geometric constraints and inventory semantics.

IV. COMPUTATIONAL VERIFICATION AND FRAMEWORK VALIDATION

To validate the proposed mathematical framework, a computational implementation was developed to translate the theoretical formulation into a reproducible numerical workflow. The implementation was performed using Python 3 within a Jupyter Notebook environment, where each stage of the proposed framework was systematically executed, including spatial discretization, coordinate generation, matrix layer construction, tensor assembly, and weight assignment.

The computational notebook serves as a complementary research artifact that enables independent verification and reproduction of the numerical results presented in this section. While the manuscript focuses on the mathematical formulation and the underlying modeling principles, the accompanying implementation provides the complete executable procedure required to generate the computational domain and reproduce the presented visualizations and tensor representations.

The validation process is organized into sequential computational stages. First, the warehouse environment is transformed from a continuous physical space into a structured discrete domain. Subsequently, individual spatial layers are represented as matrices, which are then assembled into a higher-order tensor structure. Finally, weighting values are assigned to the spatial cells to demonstrate the capability of the framework to represent heterogeneous warehouse environments.

4.1 Experimental Setup

The experimental setup consists of defining a structured three-dimensional computational domain representing the spatial organization of a logistics warehouse. The warehouse environment is modeled as a finite Cartesian grid, where the continuous physical space is transformed into a discrete set of computational cells. Structured grid representations are widely used in numerical modeling because they provide regular spatial indexing, efficient data organization, and a direct correspondence between physical coordinates and computational elements (Ferziger & Perić, 2002) [22] and [14].

The warehouse domain is defined by its physical dimensions along the horizontal directions, represented by the length Lx and width Ly, while the vertical dimension is represented through a finite number of rack levels Nz. The spatial resolution of the computational grid is controlled by the discretization intervals Δx and Δy, which determine the size of each elementary spatial cell. Consequently, the warehouse space is represented as a collection of discrete volumetric elements, where each cell corresponds to a unique spatial position within the computational domain.

The number of cells generated along each horizontal axis is calculated using Eq. (11) where the ceiling operation ensures that the complete physical domain is covered even when the warehouse dimensions are not exact multiples of the selected spatial resolution. The total number of computational cells composing the three-dimensional domain is then calculated by Eq. (12) where N_{total} represents the number of discrete spatial elements that will later constitute the tensor representation.

For the computational verification, a warehouse domain of dimensions $L_x = 26m$ and $L_y = 29m$ was considered. The spatial resolution was fixed at $\Delta x = \Delta y = 1.2m$, resulting in a regular horizontal discretization, while the vertical structure was represented by four rack levels $Nz=4$. This configuration provides a sufficiently detailed spatial representation to demonstrate the proposed framework while maintaining a computationally manageable domain size.

The resulting computational domain establishes the fundamental spatial structure required for the following stages of the framework, with a total of 2200 computational cells while individual grid layers are represented as matrices and subsequently assembled into a higher-order tensor representation.

4.2 Grid Generation

After defining the structured computational domain, the next stage consists of generating the spatial coordinate system associated with each computational cell. This process establishes the mapping between the discrete representation of the warehouse environment and its corresponding three-dimensional physical coordinates.Each cell within the discretized domain is uniquely identified by its spatial position, formalized as the indexed coordinate triplet defined in Eq. (13).

Here, the operational indices along the longitudinal, transversal, and vertical directions are bounded as established in Section 3.

The spatial coordinates along the horizontal axes are mapped directly using the discretization intervals Δx and Δy, whereas the vertical coordinate maps directly to the discrete operational rack levels $z_k = k$ for $k \in \{0, 1, \dots, N_z - 1\}$. This spatial transformation materializes the complete coordinate domain configuration described by Eq. (11) and Eq. (13).

For linear algebra applications [16] and computational optimization loops, the complete coordinate representation can be unpacked and organized into a flattened global coordinate matrix: $\mathbf{C} \in \mathbb{R}^{N_{Total} \times 3}$ where each row index $\mathbf{n} \in \{0, 1, \dots, N_{total} - 1\}$ corresponds to a unique single computational cell containing its associated physical spatial coordinates:

$$\mathbf{C_n} = [\mathbf{x_n} \quad \mathbf{y_n} \quad \mathbf{z_n}] \ (18)$$

Here, the spatial entries are mapped systematically via a lexicographical (row-major) order from the underlying tensor structure.

For the computational verification, evaluating the domain through the criteria of Eq. (11) and Eq. (12) results in:

$$\mathbf{N_x = 22}, \qquad \mathbf{N_y = 25}, \qquad \mathbf{N_z = 4}$$

yielding a comprehensive mesh size of:

$$\mathbf{N_{total} = N_x N_y N_z = 22 \times 25 \times 4 = 2,200}$$

This coordinate generation stage provides the fundamental spatial indexing mechanism required for the subsequent matrix layer construction.

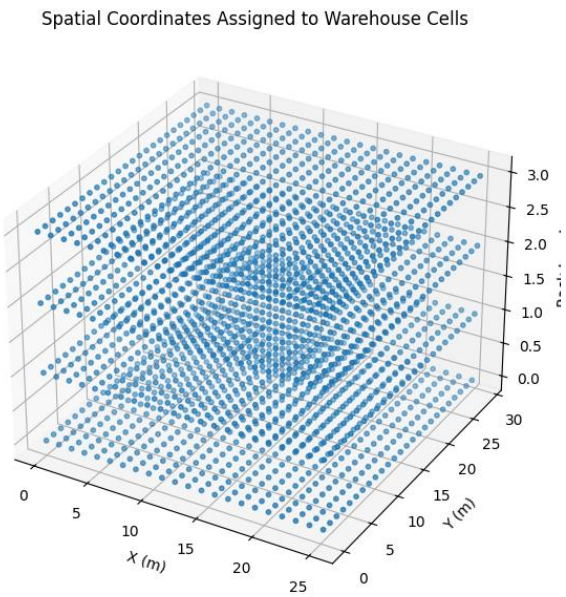

Fig. 6: Spatial coordinates assigned to warehouse cells across the generated three-dimensional computational domain.

4.3 Matrix Layer Construction

Following the generation of the spatial coordinate system, the three-dimensional computational domain is decomposed into a sequence of two-dimensional matrix layers. This transformation provides an intermediate representation between the individual cell coordinates and the final higher-order tensor structure.

For a fixed vertical level $\mathbf{z_k}$, all cells sharing the same rack-level index are grouped into a two-dimensional spatial matrix slice as defined in Eq. (14). Each element of the matrix maps directly to a unique spatial cell located at the position described by the coordinate triplet in Eq. (13): $\mathbf{M_k(i,j) \leftrightarrow C_{ijk}}$

Consequently, the matrix index pair $(\boldsymbol{i}, \boldsymbol{j})$ preserves the horizontal spatial relationships of the layout, while the layer index $\boldsymbol{k}$ represents the vertical dimension.

The complete sequence of matrix layers is organized into the finite set $\boldsymbol{\mathcal{M}}$, where each frontal matrix slice possesses identical horizontal dimensions: $\mathbf{M_k \in R^{N_x \times N_y}}, \qquad \mathbf{\forall k \in \{0, 1, \dots, N_z - 1\}}$ (19)

For the computational verification, evaluating the domain through the grid dimensions established in Section 4.2 yields a mesh layout of:

$$\boldsymbol{N_x = 22}, \qquad \boldsymbol{N_y = 25}, \qquad \boldsymbol{N_z = 4}$$

Therefore, each individual matrix layer contains exactly: $\mathbf{22 \times 25 = 550}$

discrete spatial cells. The complete warehouse domain is thus structurally partitioned into four separate matrix layers:

$$\boldsymbol{\mathcal{M}} = \{\mathbf{M_0, M_1, M_2, M_3}\}$$

This matrix representation preserves the spatial topology of each warehouse level by maintaining the neighborhood relationships between adjacent cells along the horizontal directions. Fig. 5 illustrates one matrix layer as a two-dimensional cell structure, where each element corresponds to one computational cell indexed by its spatial position.

The generated matrix layers provide the necessary intermediate structure for the subsequent tensor assembly stage, where the individual matrices are stacked systematically along the third mode to obtain the complete three-dimensional tensor representation of the warehouse environment.

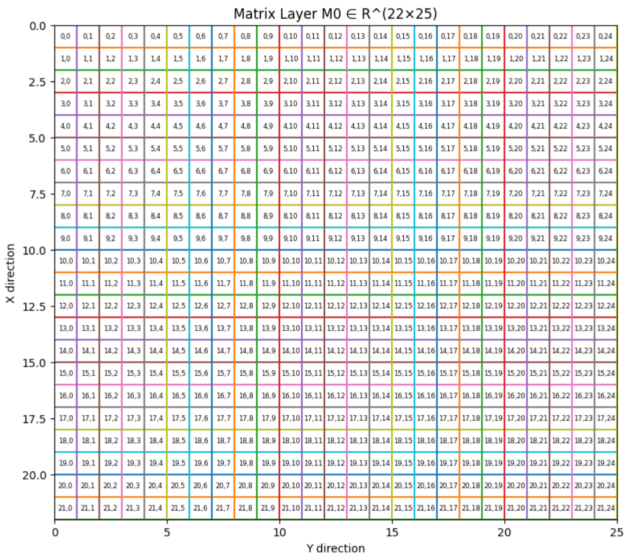

Fig. 7: Two-dimensional matrix representation of a single frontal grid layer ($M_0 \in \mathbb{R}^{22\times25}$). The lattice details the indexing grid scheme where each matrix element maps structurally to a specific warehouse spatial cell coordinate from Eq. (13) and Eq. (14).

4.4 Tensor Assembly

Following the construction of the two-dimensional matrix layers, the complete spatial representation of the warehouse is assembled into a third-order tensor. Each matrix layer corresponds to a fixed vertical level $\mathbf{z_k}$ and represents the horizontal distribution of discrete cells along the longitudinal and transversal directions.

The global tensor representation is defined as: $\mathbf{X} \in \mathbf{R}^{\mathbf{N_x} \times \mathbf{N_y} \times \mathbf{N_z}}$ where the first two modes represent the horizontal spatial dimensions of the warehouse, while the third mode represents the sequence of vertical rack levels.

The tensor is constructed by stacking the individual matrix layers systematically along the third mode (mode-3 frontal slices) in accordance with the structural framework established in Section 3: Eq. (19) where $\mathbf{X}_{(:,:,\mathbf{k})}$ denotes the frontal slice of the tensor corresponding to the $\boldsymbol{k-th}$ warehouse level. This 0-indexed formulation preserves direct alignment with the grid coordinate properties verified in **Eq. (13)** and **Eq. (14)**.

For the computational verification layout, stacking the four independents $(\mathbf{22} \ \times \mathbf{25})$ matrices yield a unified third-order tensor: $X \in R^{22 \times 25 \times 4}$.

This higher-order structure encompasses the complete set of $\boldsymbol{N_{total}} = \mathbf{2,200}$ computational cells. The numerical integration confirms that each generated matrix layer is preserved as an isolated tensor slice without altering the spatial ordering or neighborhood topography of the discrete cells.

The resulting tensor structure successfully materializes the transition from individual two-dimensional spatial layers into a global multidimensional representation, providing a consistent, unified computational realization of the mathematical framework.

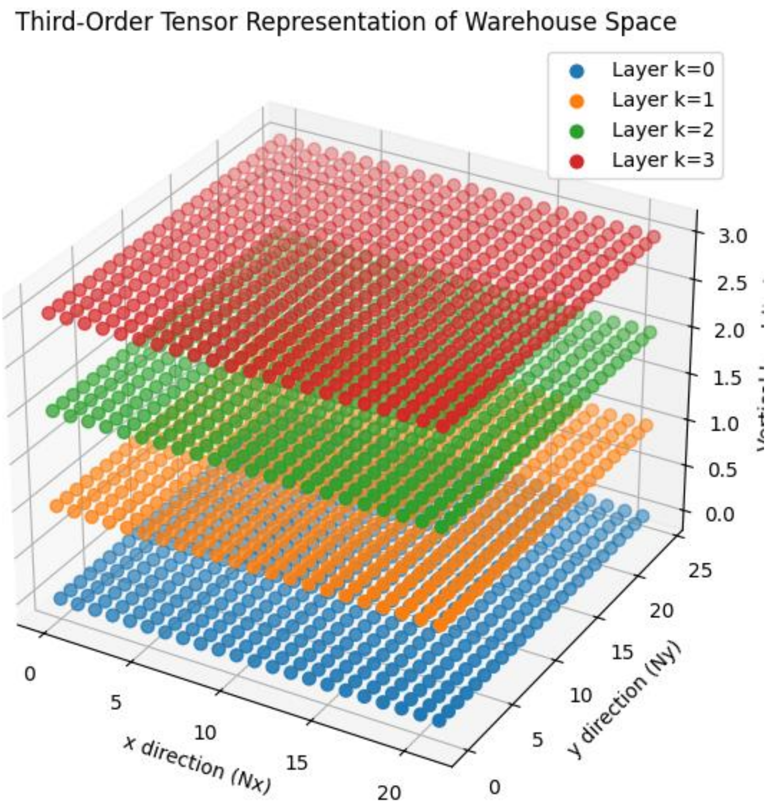

Fig. 8: Third-order tensor representation $X \in R^{22 \times 25 \times 4}$ of the warehouse computational space. The visualization illustrates the structural assembly of four distinct frontal matrix layers stacked vertically along mode-3.

4.5 Operational Semantics and Multiplicative Weighting

Following the integration of the structured components into a unified third-order tensor, the operational conditions of the warehouse domain are established. This stage defines the functional mapping that assigns an operational accessibility or cost metric to each discrete volumetric element, transforming the purely spatial geometric grid into a functionally weighted decision workspace. This mapping operation formalizes the complex physical and logistical constraints of industrial material handling environments within a strict multi-linear algebraic framework.

In accordance with the structural framework established in Section 3, the assignment of operational weights within the global tensor is governed by a multiplicative weighting function. For any given computational cell index triplet, the corresponding tensor element value is calculated as: $\boldsymbol{X_{i,j,k}} = \boldsymbol{\omega\left(C_{i,j,k}\right) \cdot \gamma(k)}$ where $\boldsymbol{\omega\left(C_{i,j,k}\right)}$ represents the horizontal spatial accessibility function derived from the cell's physical coordinate mapping established in Eq. (13), and $\boldsymbol{\gamma(k)}$ denotes the vertical rack-level operational discount factor. This mathematical decomposition decouples horizontal travel complexities such as corridor proximity, traffic congestion zones, and picking path distances from the structural and mechanical constraints associated with vertical lifting heights, hoisting speeds, and safety tiers at varying elevator or crane levels.

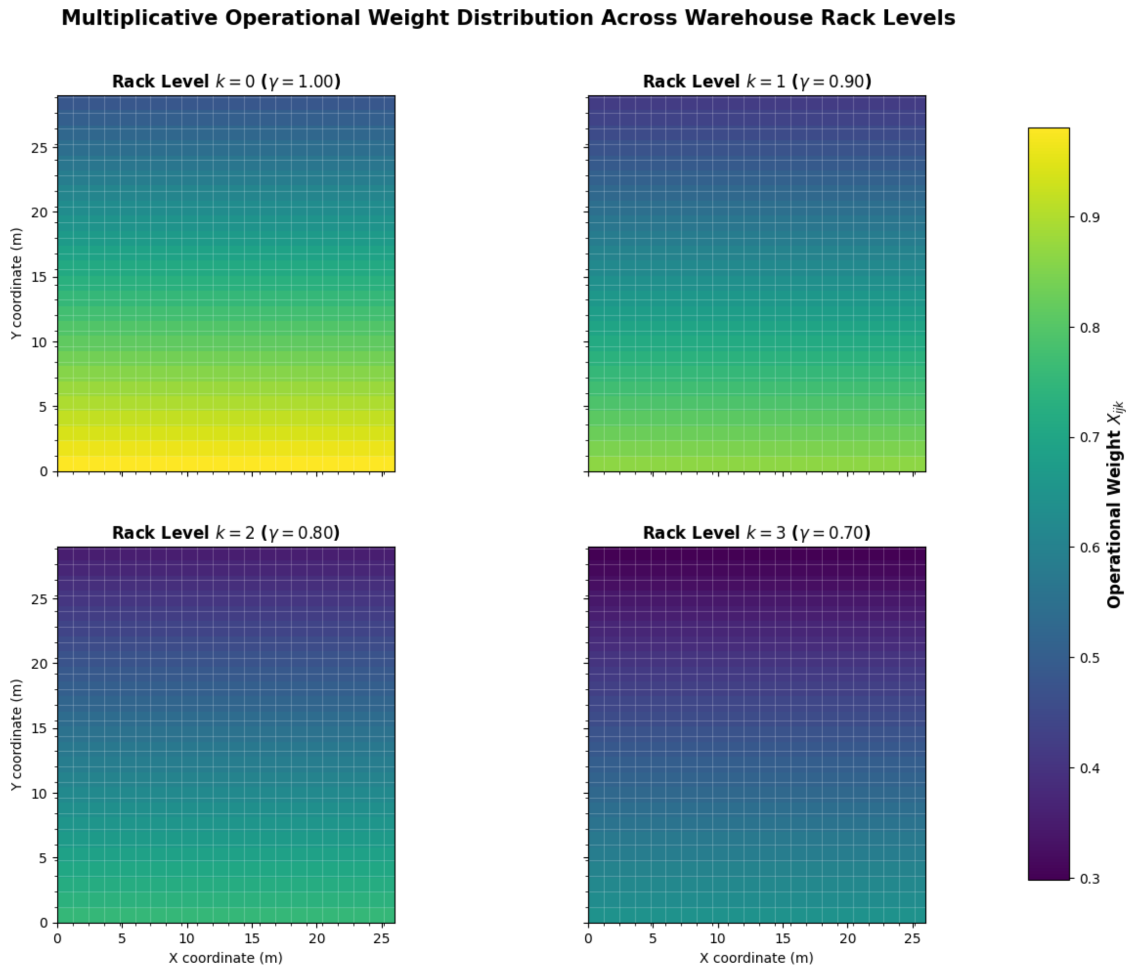

Fig. 9: Multiplicative operational weight distribution across the warehouse computational domain layers.

The product formulation ensures that both spatial dimensions and structural constraints scale dependently. If a particular cell location is completely inaccessible horizontally due to fixed warehouse infrastructure or structural columns, the function evaluates to

$\boldsymbol{\omega\left(C_{i,j,k}\right) = 0}$, completely nullifying the weight of the cell across all levels. Conversely, operational bottlenecks at extreme racking heights can be seamlessly modulated via $\boldsymbol{\gamma(k)}$ across an entire frontal matrix slice $\boldsymbol{M_k}$ without altering the underlying horizontal coordinate properties verified in **Eq. (14)**.

For the computational verification setup, the global weight tensor is instantiated over the previously validated domain dimensions: $\boldsymbol{X \in R^{22 \times 25 \times 4}}$. The entry-by-entry evaluations are processed utilizing vectorized array operations across the $\boldsymbol{N_{total} = 2,200}$ cells, guaranteeing absolute numeric synchronization between the physical dimensions and structural layer constraints. The continuous spatial variation of $\boldsymbol{\omega\left(C_{i,j,k}\right)}$ maps directly to the continuous coordinate matrix $\mathbf{C} \in \mathbf{R}^{2,200 \times 3}$. This ensures that the discrete cell lattices retain absolute geometric parity with the continuous physical boundaries of the logistics workspace.

This multi-linear weighting approach expands on established discrete numerical domain modeling principles [22], adapting them to tensor-based spatial representation structures. By converting spatial cells into numerical operational weights, the tensor X serves as a direct input for downstream optimization routines, including constrained path planning algorithms and tensor-based factorization architectures for spatial data reduction.

4.6 Computational Resolution and Spatial Discretization Trade-off

The accuracy and computational requirements of the proposed tensor-based spatial representation are directly influenced by the selected spatial resolution parameters $\boldsymbol{\Delta x}$ and $\boldsymbol{\Delta y}$. These parameters define the granularity of the computational domain by controlling the size of each discrete spatial cell. Consequently, the choice of spatial resolution introduces a fundamental trade-off between representation precision and computational cost.

In numerical modeling, reducing the discretization interval increases the number of computational elements required to represent a given physical domain. This relationship along the horizontal axes is governed by the ceiling operations formalized in **Eq. (11)**. Consequently, the total number of spatial elements composing the three-dimensional computational workspace grows according to the multi-linear accumulation product detailed in **Eq. (12)**.

A reduction in $\boldsymbol{\Delta x}$ and $\boldsymbol{\Delta y}$ results in a finer spatial discretization, expanding the dimensions of the resulting global tensor representation X. While finer resolutions improve the ability of the tensor to capture localized spatial variations and provide a more accurate representation of warehouse geometry, they also increase memory requirements and computational complexity. This behavior is a fundamental characteristic of grid-based numerical methods, where increased spatial resolution generally requires additional computational resources [14] and [22]

Conversely, larger discretization intervals reduce the number of computational cells and consequently decrease computational requirements. However, this simplification results in a lower-resolution spatial tensor, where multiple physical locations may be aggregated into a single computational cell. Such aggregation may reduce the ability of the representation to preserve local spatial characteristics and operational variations.

The relationship between spatial resolution and tensor precision can therefore be summarized as a resolution-complexity trade-off. A reduction in $\boldsymbol{\Delta x}$ and $\boldsymbol{\Delta y}$ yields a corresponding increase in the grid dimensions $\boldsymbol{Nx}$ and $\boldsymbol{Ny}$ via **Eq. (11)**, providing higher tensor precision. Conversely, an increase in $\boldsymbol{\Delta x}$ and $\boldsymbol{\Delta y}$ reduces the dimensions $\boldsymbol{Nx}$ and $\boldsymbol{Ny}$, leading to a lower overall computational cost for the framework.

The selection of appropriate discretization parameters is therefore dependent on the intended application of the spatial tensor. Applications requiring detailed spatial differentiation, such as precise storage allocation, accessibility analysis, or local optimization, benefit from finer resolutions. In contrast, applications requiring large-scale analysis may prioritize computational efficiency through coarser spatial representations.

This resolution-dependent behavior follows established principles in computational science, where numerical accuracy, memory consumption, and processing requirements must be balanced according to the objectives and constraints of the model [14] and (Quarteroni & Valli, 2008) [23].

V. CONCLUSION

This paper fulfilled its primary objective by developing a mathematical grid-based modeling framework that transforms a continuous physical warehouse into a finite three-dimensional Cartesian space, [7], [15]. Building upon spatial discretization, cellular automata lattice concepts, and multidimensional structures, the model establishes a direct indexed relationship between continuous coordinates and discrete cells. To achieve structured mapping, spatial coordinate fields were systematically generated, organized into matrix slices representing vertical rack tiers, and stacked to construct a global third-order tensor under rigorous linear algebra principles. Computational verification on a twenty-six meter by twenty-nine-meter facility with four rack levels validated the framework's capacity to handle non-integer grid parameters, generating exactly two thousand two hundred unique elements. The objective of assigning localized semantic indicators was met using a non-linear multiplicative weighting function, embedding operational attributes while maintaining strict topology and coordinate parity. The resolution-complexity analysis highlighted the fundamental trade-off between geometric precision and the computational costs governed by resolution parameters, [14], [22], [23], [18], [19]. This robust structure successfully preserves multidimensional spatial features to support advanced optimization pipelines in smart logistics, [5], [12]. Future research objectives will expand this framework across three areas: Dimensionality Reduction: Applying CP and Tucker tensor decompositions to uncover latent spatial patterns within large logistics datasets. Autonomous Automation: Integrating the weighted tensor as direct operational inputs for path planning and fleet routing algorithms governing automated guided vehicles, automated storage and retrieval systems, and autonomous mobile robots, [3], [4]. Spatio-Temporal Extensions: Scaling the model into a fourth-order tensor to dynamically capture real-time inventory alterations and structural corridor traffic evolution over time, [5].

REFERENCES

[1] Winkelhaus, S., & Grosse, E. H. (2020). Logistics 4.0: A systematic review towards a new logistics system. International Journal of Production Research, 58(1), 18–43. 10.1080/00207543.2019.1612964.

[2] Baker, P., & Halim, Z. (2007). An exploration of warehouse automation implementations: Cost, service and flexibility issues. Supply Chain Management: An International Journal, 12(2), 129–138. 10.1108/13598540710737316.

[3] Keith, R., & La, H. M. (2024). Review of autonomous mobile robots for the warehouse environment. arXiv preprint arXiv:2406.08333.

[4] Roodbergen, K. J., & Vis, I. F. A. (2009). A survey of literature on automated storage and retrieval systems. European Journal of Operational Research, 194(2), 343–362. https://doi.org/10.1016/j.ejor.2008.01.038

[5] Zhen, L., & Li, H. (2022). A literature review of smart warehouse operations management. Frontiers of Engineering Management, 9(1), 31–55. https://doi.org/10.1007/s42524-021-0178-9

[6] Dray, S., Legendre, P., & Peres-Neto, P. R. (2006). Spatial modelling: A comprehensive framework for principal coordinate analysis of neighbour matrices (PCNM). Ecological Modelling, 196(3–4), 483–493. https://doi.org/10.1016/j.ecolmodel.2006.02.015

[7] von Neumann, J. (1966). Theory of self-reproducing automata (A. W. Burks, Ed.). University of Illinois Press.

[8] Kaufman, A., & Shimony, E. (1987). 3D scan-conversion algorithms for voxel-based graphics. In Proceedings of the 1987 workshop on Interactive 3D graphics (pp. 45–75). Association for Computing Machinery. https://dl.acm.org/doi/10.1145/319120.319126

[9] Rouwenhorst, B., Reuter, B., Stockrahm, V., van Houtum, G. J., Mantel, R. J., & Zijm, W. H. M. (2000). Warehouse design and control: Framework and literature review. European Journal of Operational Research, 122(3), 515–533. https://doi.org/10.1016/S0377-2217(99)00020-X

[10] Albert, P.-W., Rönnqvist, M., & Lehoux, N. (2023). Trends and new practical applications for warehouse allocation and layout design: A literature review. SN Applied Sciences, 5, Article 378. https://doi.org/10.1007/s42452-023-05608-0

[11] Mohamud, I. H., Kafi, M. A., Shahron, S. A., Zainuddin, N., & Musa, S. (2023). The role of warehouse layout and operations in warehouse efficiency: A literature review. Journal Européen des Systèmes Automatisés, 56(1), 61–68. https://doi.org/10.18280/jesa.560109

[12] ten Hompel, M., & Schmidt, T. (2007). Warehouse management: Automation and organisation of warehouse and order picking systems. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-35220-4

[13] Zienkiewicz, O. C., Taylor, R. L., & Zhu, J. Z. (2013). The finite element method: Its basis and fundamentals (7th ed.). Elsevier.

[14] LeVeque, R. J. (2007). Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems. Society for Industrial and Applied Mathematics.

[15] Samet, H. (2006). Foundations of multidimensional and metric data structures. Morgan Kaufmann.

[16] Strang, G. (2016). Introduction to linear algebra (5th ed.). Wellesley-Cambridge Press.

[17] Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. SIAM Review, 51(3), 455–500.

[18] Accorsi, R., Manzini, R., & Maranesi, F. (2014). A decision-support system for the design and management of warehousing systems. International Journal of Production Research, 52(6), 1754–1773.

[19] Derhami, S., Smith, J. S., & Gue, K. R. (2020). Space utilization models for warehouses with multiple rack designs. International Journal of Production Economics, 219, 129–139.

[20] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., & Tarantola, S. (2008). Global sensitivity analysis: The primer. John Wiley & Sons.

[21] Borgonovo, E. (2017). Sensitivity analysis: An introduction for data analytics, computing, and decision making. Springer.

[22] Ferziger, J. H., & Perić, M. (2002). Computational methods for fluid dynamics (3rd ed.). Springer.

[23] Quarteroni, A., & Valli, A. (2008). Numerical Approximation of Partial Differential Equations. Springer.