Unit-V

Visualization of volumetric data: vector fields, processes and simulations, Visualization of maps, Evaluating visualizations.

Recent trends in various perception techniques, various visualization techniques.

1. Visualization of Volumetric Data

Volumetric data involves **3D scalar or vector fields** — values defined within a 3D space (x, y, z).

Formally:

$$f(x,y,z) \rightarrow v$$

where (x,y,z) are spatial coordinates

and v is the **value** at that point (e.g., temperature, pressure, intensity)

Common in:

- Medical imaging (CT, MRI)
- Fluid dynamics simulations
- Weather and environmental models
- 3D heat maps or density model

Visualization Goals

- Show **internal structures** (not just surface)
- Reveal **patterns, gradients, and anomalies**
- Allow **interactive exploration** of 3D data
- Support **quantitative analysis** (measuring regions, densities)

Techniques for Visualizing Volumetric Data

Ward, Grinstein, and Keim classify these into **two broad approaches:**

- **Indirect Volume Rendering**
- **Direct Volume Rendering (DVR)**

Indirect Volume Rendering

Extract surfaces or geometric structures from volume data first.

1. Isosurface Extraction

•Represents a surface of **constant scalar value** (e.g., intensity threshold).

•Uses **Marching Cubes algorithm**.

•Output: polygonal mesh.

Example:

In a CT scan, isosurfaces at different density thresholds show bone, muscle, and tissue boundaries.

2. Slicing / Sectional Views

•Extract 2D slices from 3D data (XY, XZ, YZ planes).

•Enables exploration of internal layers.

3. Surface Shading

•Apply lighting and color to extracted surfaces for realism.

Direct Volume Rendering (DVR)

No explicit geometry; instead, render directly using **opacity, color, and transfer functions**.

1. Ray Casting

•Rays are sent through the volume; intensity accumulates along each ray.

•Produces semi-transparent, depth-aware visualization.

2. Texture-Based Rendering

•Uses 2D slices of 3D textures for hardware-accelerated rendering.

3. Transfer Functions

•Map scalar values to **color + opacity**.

Example:

- Low density = transparent
- Medium density = gray
- High density = bright white (bones)

4. Maximum Intensity Projection (MIP)

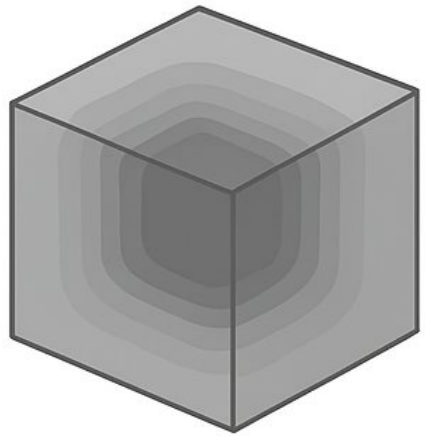
•Projects only the **maximum voxel value** along a ray.

•Common in medical imaging for highlighting bright features (e.g., blood vessels).

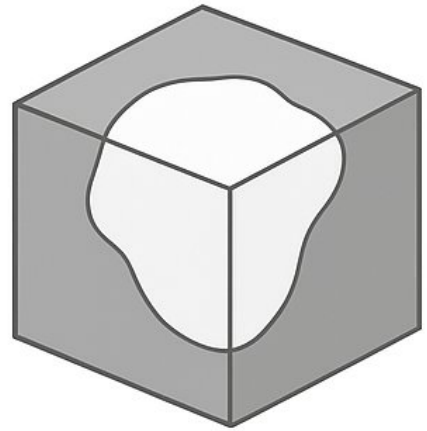
Techniques:

Technique	Description	Example Visualization
Isosurfaces	Extracts surfaces from 3D scalar fields (constant value contours).	MRI slice rendering
Volume Rendering	Directly maps volume data to opacity and color.	CT scan visualization
Slice-based Views	Displays cross-sections in orthogonal planes.	Temperature slices in simulation
Glyphs / Arrows	For representing direction and magnitude of flow.	Wind or current visualization

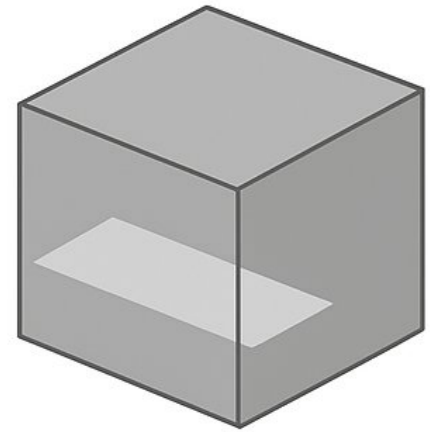
VOLUMETRIC VISUALIZATION TECHNIQUES



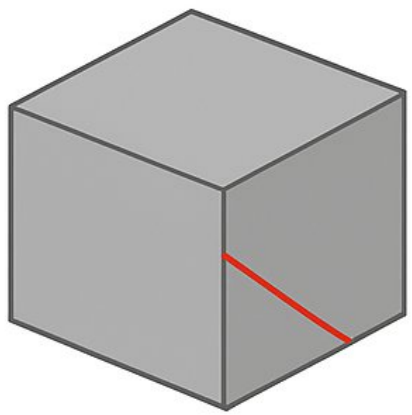
VOLUME RENDERING



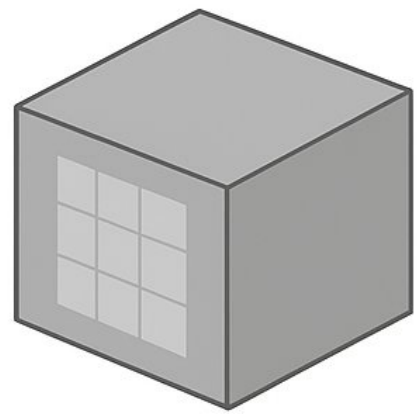
SURFACE RENDERING



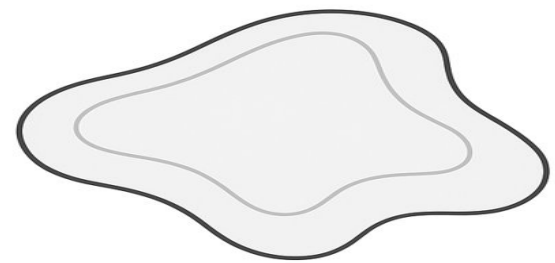
SLICE-BASED



MIP



3D TEXTURE



Isosurface

Types of Volumetric Data

Volumetric data (also called **volume data**) represents values in a **three-dimensional (3D) space**.
Each data point (called a **voxel** — volume + pixel) contains information about a specific location inside the 3D region.
Volumetric data can come from **scientific measurements, simulations, or imaging** — such as CT scans, fluid dynamics, or weather models

Type	Description	Example
Scalar field	One value per 3D location	Temperature distribution
Vector field	3D direction & magnitude	Wind flow vectors
Tensor field	Matrix per point (directional magnitude)	Stress/strain field in materials

1. Scalar Volumetric Data

•Each voxel stores **a single scalar value** (e.g., temperature, density, pressure).

•Example:

- Medical imaging (CT intensity values).
- Atmospheric temperature field.

•**Visualization techniques:**

- Volume rendering
- Isosurfaces
- Slicing planes

Example Explanation: CT Scan

•A CT scanner captures multiple X-ray images from different angles.

•Reconstruction algorithms create a **3D voxel grid**.

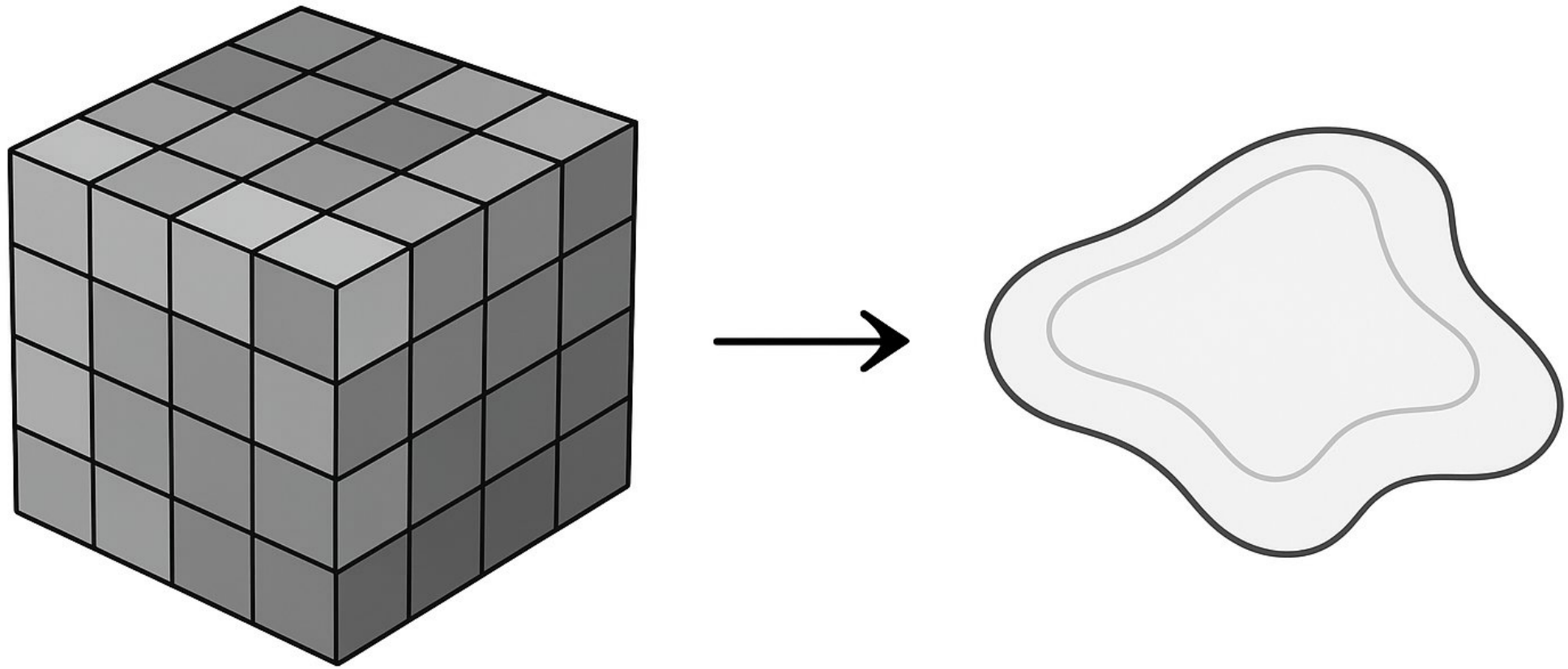
•Each voxel stores one **scalar intensity value** (Hounsfield unit).

•Visualization:

- High values → bones (white)
- Medium → soft tissue (gray)
- Low → air (black)

This is one of the most widely used real-time examples of scalar volumetric data visualization.

SCALAR VOLUMETRIC DATA



3D voxel
volume dataset

Isosurface

Real-Time Examples of Scalar Volumetric Data

Domain	Example	Description / Data Source	Visualization Technique
Medical Imaging	**CT (Computed Tomography) Scan**	Each voxel stores **X-ray attenuation coefficient** — showing tissue density (bone, muscle, organ).	Volume rendering or isosurfaces (e.g., for bone surfaces)
MRI (Magnetic Resonance Imaging)	MRI voxels store **signal intensity** representing different tissue types in the brain or body.	3D slicing, transparency rendering, color mapping	
Weather / Climate Simulation	Atmospheric **temperature or pressure fields** represented as scalar values across 3D grids.	3D contour maps, isosurfaces (constant-temperature surfaces)	
Fluid Dynamics	Simulation of **temperature or density** in a combustion chamber or airflow region.	Volume rendering or isovolume extraction	
Geology / Seismic Data	Voxel data represents **rock density or seismic amplitude** across underground layers.	3D volumetric rendering used in oil exploration	
Oceanography	Ocean **salinity or temperature distribution** at various depths.	3D layered visualization, color-coded slices	

CT Scan Visualization in Power BI (Scalar Volumetric Data Implementation)

- While Power BI isn't built for full 3D voxel rendering (like ParaView or MATLAB), you *can simulate volumetric visualization* using layered 2D slices or 3D maps.
- To simulate a **CT scan visualization** by representing **scalar intensity values** (density) from a 3D dataset as layered heatmaps or stacked slices in Power BI.

•Each **Slice** = a layer in the CT scan
•Each (X, Y) pair = pixel location
•**Intensity** = scalar value (grayscale brightness)

Step-by-Step Implementation
Step 1 – Prepare the Dataset
•Create or download a **CT dataset** (e.g., sample lung or brain CT).
•Structure your data in CSV format as:

Slice	X	Y	Intensity
1	1	1	45
1	1	2	56
1	2	1	70
2	1	1	50
2	1	2	62
…	…	…	…

Step 2 – Load Data into Power BI

1.Open Power BI → *Get Data* → *CSV/Excel* → *Load*.
2.Import the table containing `Slice`, `X`, `Y`, and `Intensity`.
3.Ensure all numeric fields are correctly recognized as numbers.

Step 3 – Create a Heatmap for Each Slice

You can use:
•The **Matrix visual** (native) or
•The **Heatmap custom visual** from AppSource (recommended).

Using the Heatmap visual:

1.Go to **Visualizations → Get more visuals → AppSource**.
2.Search for **"Heatmap"** (e.g., "Synoptic Panel" or "Heatmap by Weiwei Cui").
3.Add it to your report.
4.Configure:

1. **X-Axis:** X
2. **Y-Axis:** Y
3. **Values:** Intensity (Average)
4. **Legend/Filters:** Slice (so you can view one slice at a time)

Step 4 – Add a Slice Slider

•Add a **Slicer visual** for `Slice`.

•When you move the slider, the heatmap changes → simulating moving through CT slices.

•This mimics **scrolling through the 3D volume** layer-by-layer.

Step 5 – Color Map

•In "Data colors", choose a **diverging grayscale** palette (black → white).

•Dark = low intensity (air), Light = high intensity (bone).

•This simulates CT density values.

Step 6 – Add Interactivity

•Add tooltips showing `Intensity` on hover.

•Optionally add a **Play Axis visual** (custom visual) to animate slices like a real CT scan scroll.

Step 7 – Dashboard Enhancement

•Title: *"CT Scan Visualization – Scalar Volumetric Data"*

•Add explanation cards for:

- Voxel definition
- Scalar intensity meaning
- Slice navigation

Output

Interactive dashboard where the user can:

•Scroll through CT slices

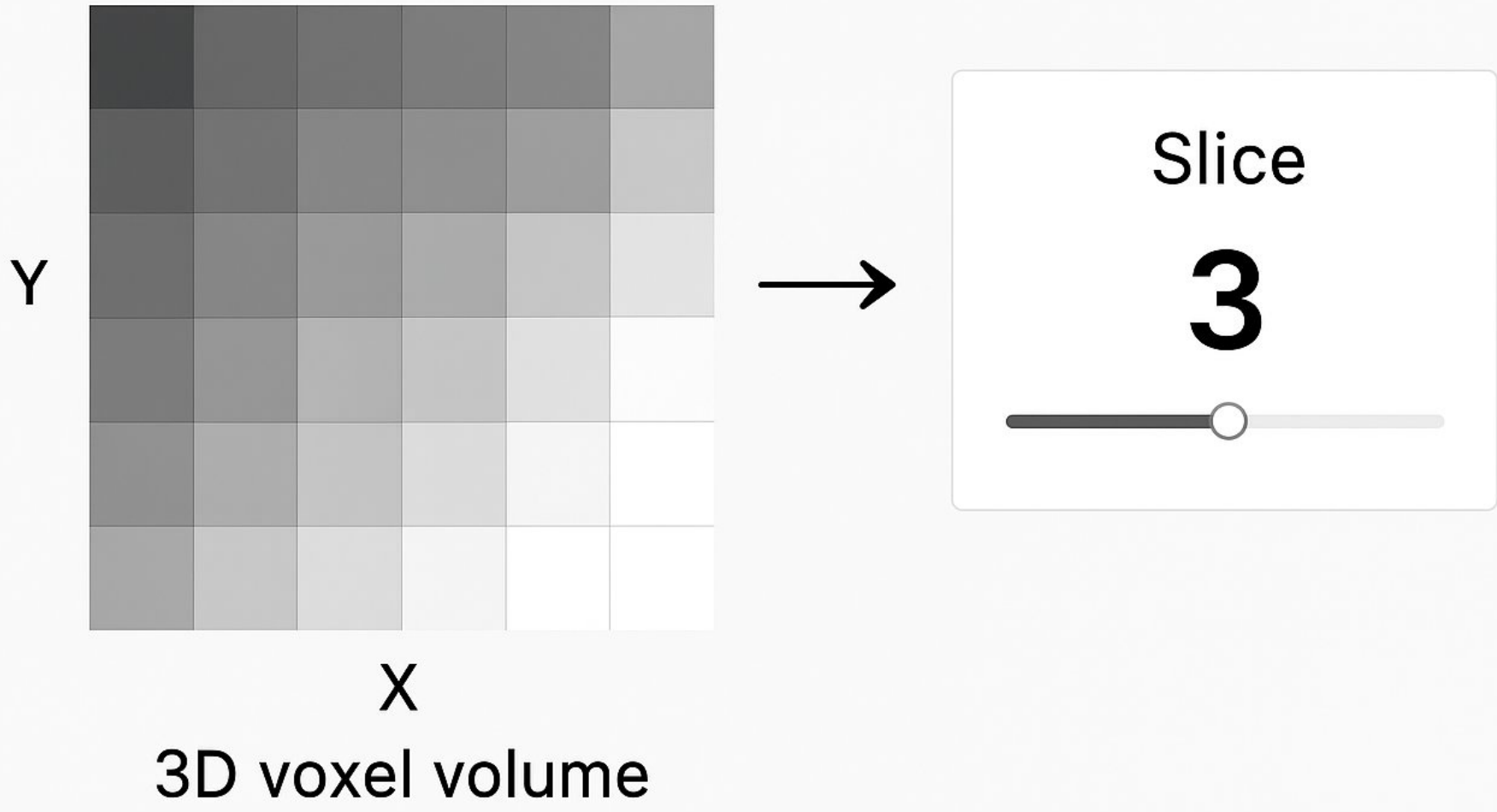
•Observe varying intensity patterns

•Interpret dense (bright) vs. hollow (dark) regions

This effectively demonstrates **scalar volumetric data visualization** using **2.5Drepresentation**.

CT SCAN VISUALIZATION

SCALAR VOLUMETRIC DATA



Vector Field Visualization

A **vector field** is a dataset where each point in space is associated with a **vector** — a quantity having both **magnitude** and **direction**.
Formally:

$$V(x,y,z)=(Vx,Vy,Vz)$$

Each vector describes how something (like wind, fluid, or magnetic field) moves through space.

Purpose

- To visualize **flow**, **direction**, and **strength** of physical quantities.
- Common in scientific, meteorological, and engineering simulations.

Real-Time Examples

Domain	Example	Vector Quantity
Meteorology	Wind flow simulation	Wind speed and direction
Oceanography	Water current flow	Velocity of ocean currents
Physics	Magnetic or electric field	Field strength and direction
Engineering	Airflow over an airplane wing	Velocity and turbulence
Medicine	Blood flow visualization	Flow vectors in arteries

Vector Field Data Representation

•Each grid point (voxel) contains:

- **Position:** (x, y, z)
- **Vector components:** (Vx, Vy, Vz)

X	Y	Z	Vx	Vy	Vz
1	1	1	0.2	0.4	0.1
1	2	1	0.1	0.3	0.0
…	…	…	…	…	…

Vector Field Visualization Techniques

1. Glyph-Based Visualization

•Draws arrows, cones, or lines at each data point.

•Arrow direction = vector direction, Arrow length = magnitude.

• *Use Case:* Simple 2D/3D flow direction display.

2. Streamlines

•Curves that follow the instantaneous direction of the flow field.

•Tangent to velocity vector at every point.

Use Case: Visualizing steady fluid flow.

3. Pathlines

•Actual trajectory of a particle over time in a time-varying flow.

Use Case: Particle motion simulation (e.g., smoke trails).

4. Streaklines

•Shows the locus of all particles passing through a fixed point over time.

Use Case: Wind tunnel visualization, flow release points.

5. Line Integral Convolution (LIC)

•Texture-based technique that creates streak patterns showing flow direction and density.

•Smooth, continuous representation.

Use Case: Visualizing complex turbulent flow fields.

6. Stream Surfaces / Stream Volumes

•Extends streamlines into surfaces or 3D regions to represent flow layers.

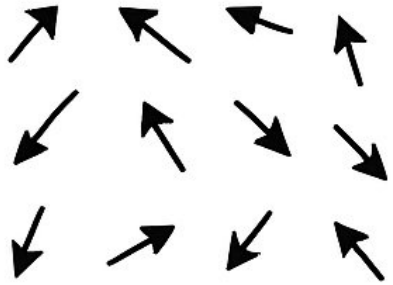
Use Case: 3D simulations (weather, aerodynamics).

Technique	Description	Application
Glyphs	Arrows or cones show direction and magnitude	Simple field maps, Weather forecasting dashboards. Wind turbine site analysis.
Streamlines	Curved lines following flow direction	Steady flow fields, Aerodynamic design, car wind tunnel simulation.
Pathlines	Particle trajectory over time	Dynamic flow, Air quality and pollutant spread studies.
Streaklines	Paths from a fixed point	Wind tunnel studies, Flow visualization in aerospace or automotive testing.
Line Integral Convolution(LIC)	Texture-based continuous visualization	Dense vector fields, Global climate and current modeling (NOAA, NASA visualizations).
Stream Surfaces	3D sheets following flow	Complex simulations, Aerospace and fluid system optimization.

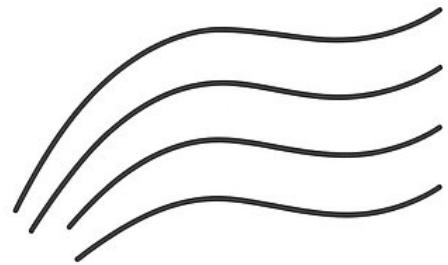
Tools for Vector Field Visualization

- **ParaView**
- **Matplotlib / Mayavi / Plotly (Python)**
- **Power BI 3D Maps (for simplified flow arrows)**
- **VTK (Visualization Toolkit)**
- **Unity / Unreal Engine (for simulation-based visualization)**

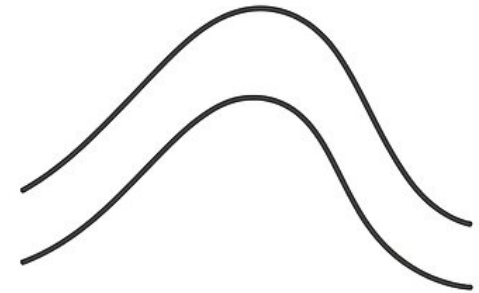
VECTOR FIELD VISUALIZATION



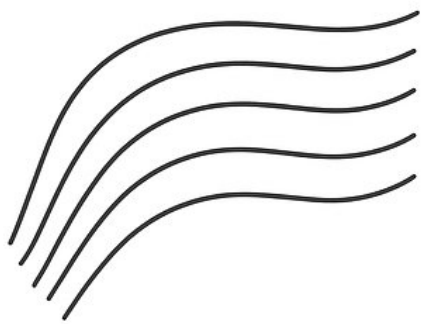
GLYPHS



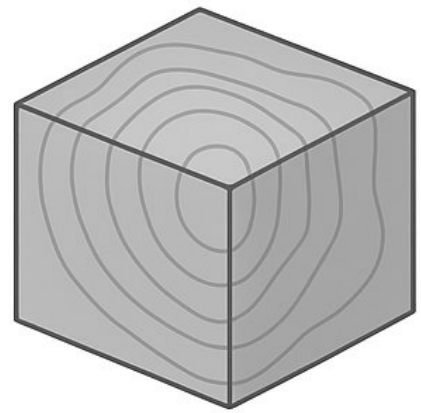
STREAMLINES



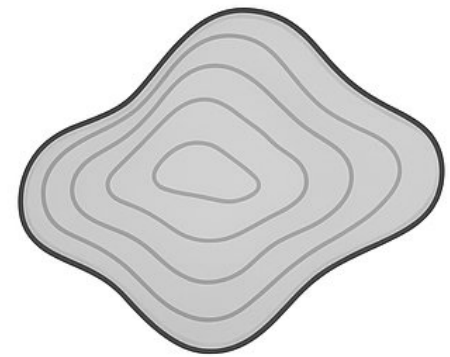
PATHLINES



STREAKLINES



LIC



STREAM SURFACES

Wind Turbine Site Analysis using Glyphs in Power BI

To visualize **wind direction** and **wind speed** across multiple locations using **arrow glyphs** — allowing analysts to identify optimal wind turbine placement sites.

- **Glyphs** represent **vectors** (magnitude + direction).
- In this context:
  - **Wind Speed** → magnitude (arrow length or color intensity).
  - **Wind Direction** → arrow rotation or bearing.
  - **Location** → latitude & longitude.

Station	Latitude	Longitude	WindSpeed (m/s)	WindDirection (°)	Elevation (m)
Site_1	14.21	78.32	5.6	180	320
Site_2	14.45	78.50	8.3	220	350
Site_3	14.58	78.61	12.1	270	390
Site_4	14.70	78.85	6.5	160	300

Interpretation:

•**WindDirection** measured in degrees (0° = North, 90° = East, etc.).

•**WindSpeed** defines arrow size or color.

3. Implementation Steps in Power BI

- **Step 1 – Load the Dataset**

1.Open **Power BI Desktop**.

2.Click **Get Data → Excel/CSV → Load.**

3.Import your wind dataset.

- **Step 2 – Add a Map Visual**

1.From the **Visualizations pane**, choose **Map** or **Filled Map**.

2.Drag the following:

1. **Latitude** → Latitude field
2. **Longitude** → Longitude field
3. **Station** → Legend or Tooltip
4. **WindSpeed** → Size (bubble radius)
5. **Elevation** → Color saturation (optional)

At this stage, you'll see location markers sized by speed.

Step 3 – Represent Wind Direction (Glyph Concept)

Power BI doesn't natively support **rotated arrows**, but we can **simulate glyphs** in two ways:

Option A: Use a Custom Visual ("Flow Map" or "Wind Rose")

1.Go to **Visualizations → Get More Visuals → AppSource.**

2.Search for and install **Flow Map** or **WindRose by DEVAR**.

3.Add it to your report.

4.Configure:

1. **Origin Latitude/Longitude:** same as your station
2. **Destination Latitude/Longitude:** calculate based on wind direction using a DAX formula (see below)
3. **Weight:** WindSpeed

This draws arrows in the direction of the wind — simulating glyphs.

Step 4 – DAX Formula to Calculate Arrow Destination

To simulate glyphs (direction vectors), create new calculated columns:

```
Dest_Latitude = [Latitude] + ([WindSpeed] * SIN(RADIANS([WindDirection])) / 100)
Dest_Longitude = [Longitude] + ([WindSpeed] * COS(RADIANS([WindDirection])) / 100)
```

Now assign:

•**Start (Latitude, Longitude)** = Origin

•**End (Dest_Latitude, Dest_Longitude)** = Destination

Step 5 – Customize the Visualization

•**Color by Speed:** Gradient color (blue → red for low → high speed).

•**Arrow Thickness:** Proportional to wind speed.

•**Tooltip Fields:**

- `Station`, `WindSpeed`, `WindDirection`, `Elevation`.

◆ **Step 6 – Add Interactivity**

•Add a **Slicer** for Wind Speed or Direction range.

•Add a **Card visual** to show average wind speed of the selected area.

•Use **Animation controls** (custom visual *Play Axis*) to animate wind direction changes over time.

Output Dashboard (Concept)

Title: "Wind Turbine Site Analysis – Glyph Visualization"

Dashboard Components:

• **Map visual:** Arrows showing wind direction and magnitude at sites.

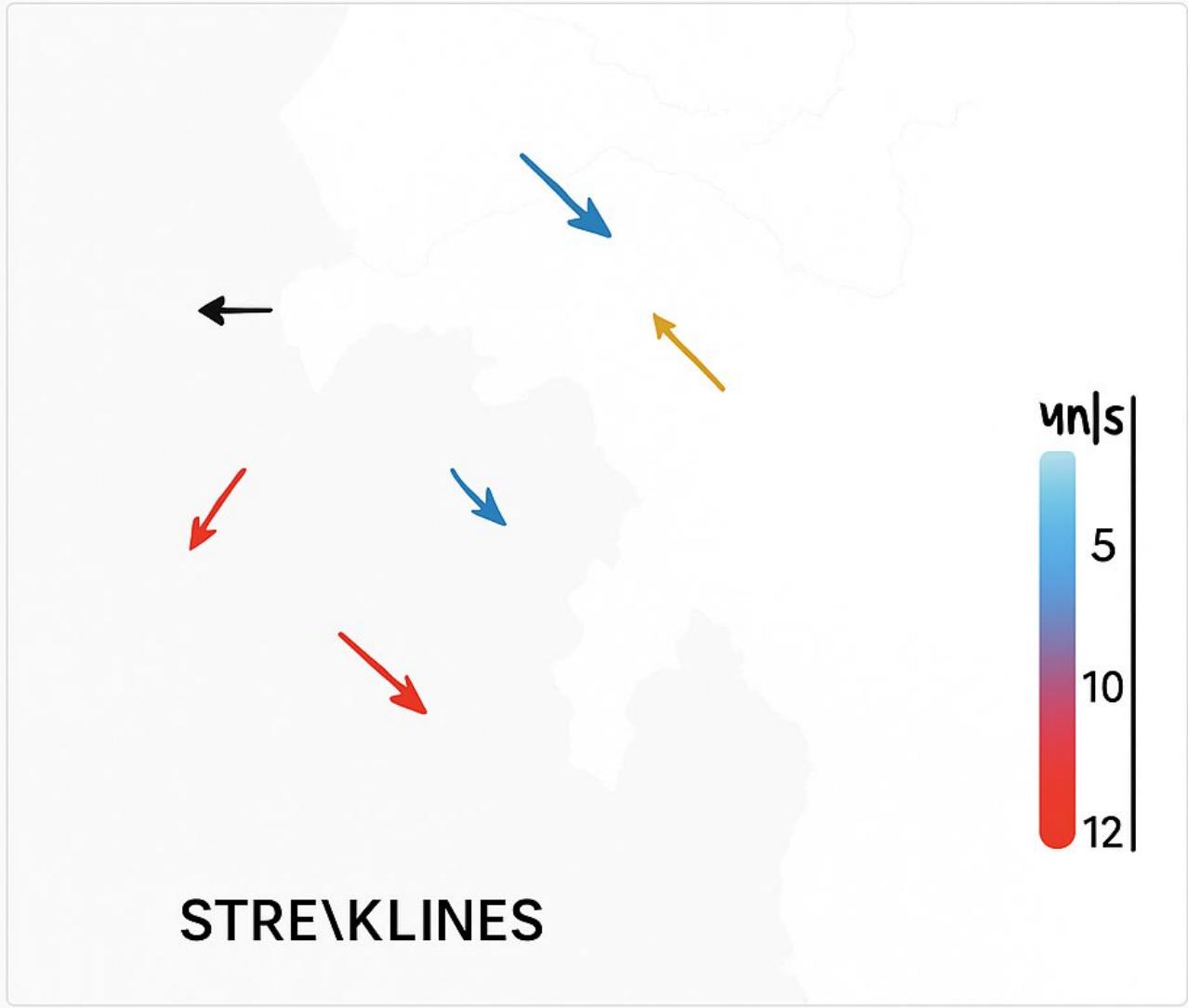
Slicers: Time, Speed range, Region filter.

•**Cards:** Average wind speed, Optimal direction.

• **Legend:** Color scale (blue = low speed, red = high).

Feature	Purpose
Glyphs	Represent wind vectors (direction + magnitude)
Flow Map Visual	Creates directional arrows
DAX Offset	Calculates destination coordinates
Tooltip Info	Displays wind metrics per station
Color & Size	Encodes wind speed visually

Wind Turbine Site Analysis – Glyph Visualization



Average Wind Speed

9.3

m/s

Wonl Speed

Optimal Direction

210

Tensor Field Visualization

Definition
A **tensor field** is a dataset in which each point in space is associated with a **tensor** — a multi-dimensional quantity describing magnitude and direction in multiple axes.
•A **scalar field** → 1 value per point (e.g., temperature).
•A **vector field** → 3 values per point (x, y, z).
•A **tensor field** → a *matrix* of values per point (multiple directions & magnitudes).
Mathematically:

$$T(x,y,z) = \begin{bmatrix} T_{xx} & T_{xy} & T_{xz} \\ T_{yx} & T_{yy} & T_{yz} \\ T_{zx} & T_{zy} & T_{zz} \end{bmatrix}$$

Tensor fields describe **anisotropic** (direction-dependent) behavior — they show how something *changes differently along different directions*.
This makes them vital for understanding **stress, strain, and diffusion** phenomena.

Real-Time Examples

Domain	Tensor Type	What It Represents	Use Case
Medical Imaging	Diffusion Tensor	How water molecules diffuse in tissues	Brain white matter visualization (DTI)
Mechanical Engineering	Stress/Strain Tensor	Deformation & internal force directions	Structural analysis of bridges, cars
Geophysics	Seismic Tensor	Direction of earth stress & wave propagation	Earthquake stress mapping
Fluid Dynamics	Viscosity/Shear Tensor	Resistance of fluid to deformation	Turbulent flow analysis
Material Science	Anisotropic Conductivity Tensor	Heat or electrical flow directionality	Crystals, composites, or alloys

Visualization Techniques (Tensor filed)

1. Tensor Glyphs

•Each tensor is visualized as an **ellipsoid** (or superquadric).

•Shape and orientation represent the principal directions and magnitudes (eigenvalues and eigenvectors).

- Elongated ellipsoid → strong directionality.
- Sphere → isotropic region.

Example: Diffusion Tensor Imaging (DTI) in the brain.

2. Hyperstreamlines

•Extends the concept of streamlines to tensor fields.

•The line follows one eigenvector direction, and the cross-section of the line varies according to the other eigenvalues.

Example: Tracking fiber pathways in the brain.

3. Tensor Slices

•2D cross-sections through a 3D tensor field.

•Each pixel shows tensor orientation (using color coding or small glyphs).

Example: Stress distribution slice through a mechanical part.

4. Tensor Volume Rendering

•Direct rendering of 3D tensor data using opacity and color to encode eigenvalues.

•Shows full volume context instead of single surfaces.

Example: Visualizing varying stress intensity within a solid.

5. Anisotropy Maps

•Scalar representation derived from tensors (e.g., fractional anisotropy in DTI).

•Shows how directional the data is (0 = uniform, 1 = fully directional).

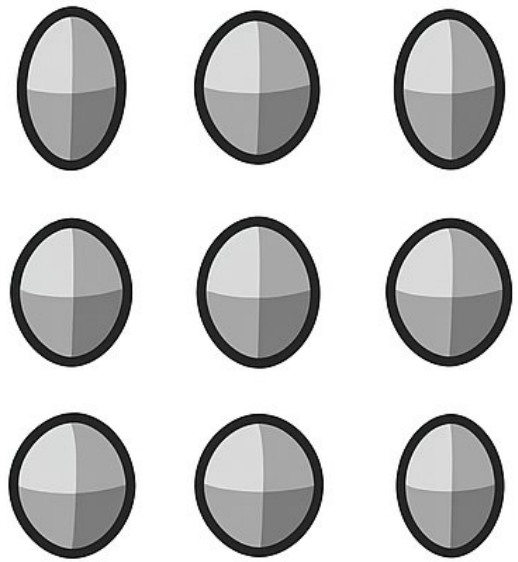
Example: FA maps in brain MRI.

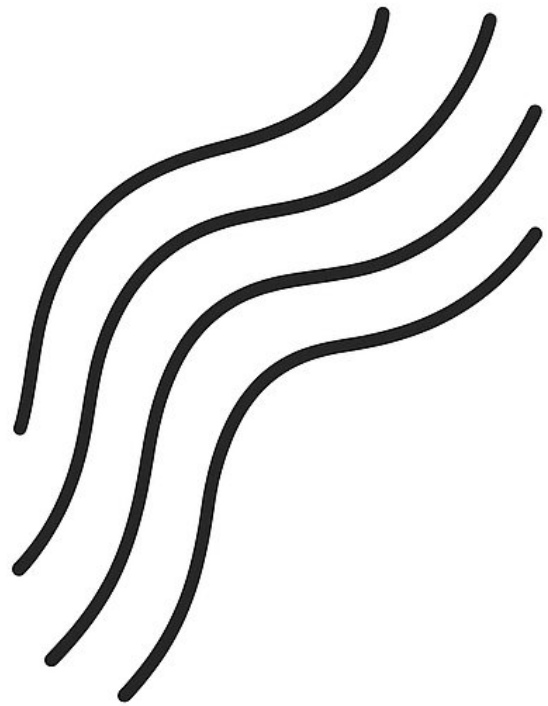
Mathematical Insight (Simplified)
Each tensor has:
•**Eigenvalues ($\lambda_1, \lambda_2, \lambda_3$)** → magnitude in each principal direction
•**Eigenvectors (v_1, v_2, v_3)** → orientation of those directions
These help visualize **directional dependence** of the property being studied.

Technique	Visualization Type	Key Representation	Application
Tensor Glyphs	3D Ellipsoids	Local magnitude & orientation	DTI, stress analysis
Hyperstreamlines	Curved tubes	Flow of principal directions	Brain connectivity
Tensor Slices	2D colored maps	Cross-sectional tensor variation	Engineering & medical
Volume Rendering	3D semi-transparent volume	Full volumetric context	Stress distribution
Anisotropy Maps	Scalar visualization	Degree of directional dependence	DTI MRI

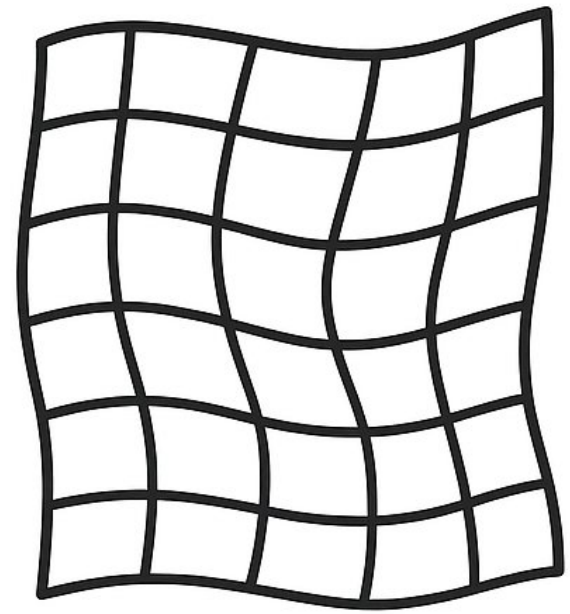
Tensor Field Visualization Techniques

Ellipsoid Glyphs





Hyperstreamlines



Deformation Grid

Type	Stored Value per Voxel	Examples	Visualization Methods
Scalar	Single numeric value	CT/MRI scans, temperature	Isosurfaces, volume rendering
Vector	Magnitude + direction	Wind, magnetic fields	Streamlines, glyphs
Tensor	Multi-directional matrix	DTI, material stress	Tensor glyphs, hyperstreamlines
Time-varying	Values change over time	Weather, fluid flow	Animation, temporal rendering

Advanced Techniques

Technique	Description	Purpose
Volume Segmentation	Divides volume into meaningful subregions	Organ separation in medical scans
Volume Simplification	Reduces voxel count	Faster rendering
Multi-Resolution Rendering	Varying detail by zoom level	Interactive performance
Hybrid Rendering	Combine isosurface + direct rendering	Best of both methods

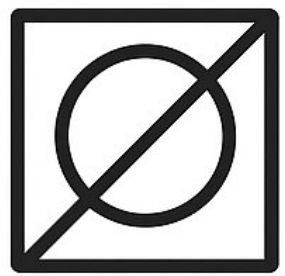
Software and Tools

Tool	Use Case	Notes
ParaView / VTK	Scientific & engineering volumetric rendering	Supports ray casting, slicing
MATLAB Volume Viewer	Simple 3D scalar visualization	Great for research
Python (Plotly / Mayavi / PyVista)	Interactive 3D visualization	Lightweight and programmable
Power BI (3D Map / ArcGIS)	Limited to 3D business data	Uses coordinates + metrics
Unity / Unreal Engine	Real-time volumetric visualization	Used in simulations and VR

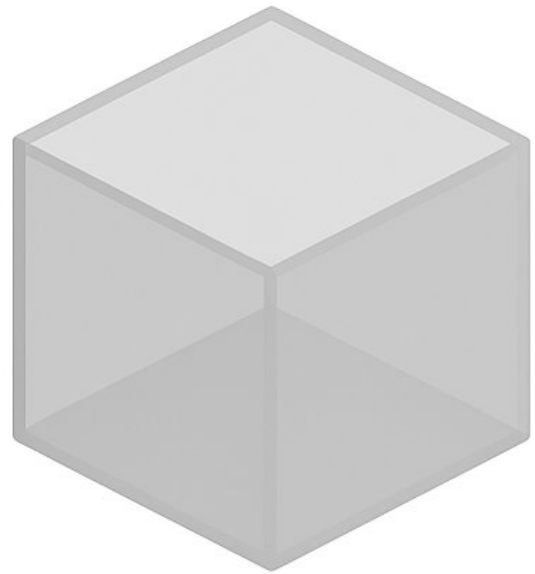
Challenges in Volumetric Visualization

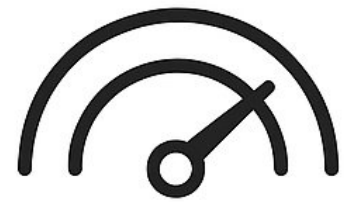
Challenge	Description
Occlusion	Internal structures hidden behind others
Performance	Rendering large volumes (e.g., 512^3 voxels) is computationally heavy
Perceptual overload	Users may misinterpret 3D density perception
Transfer function design	Requires domain knowledge for correct mapping

Challenges in Volumetric Visualization



Occlusion

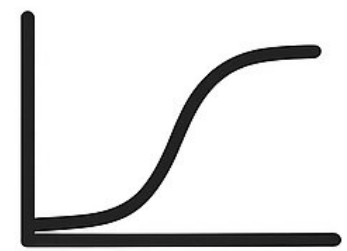




Performance



Perceptual Overload



Transfer Function Design

Example: 3D MRI Brain Tumor Visualization System

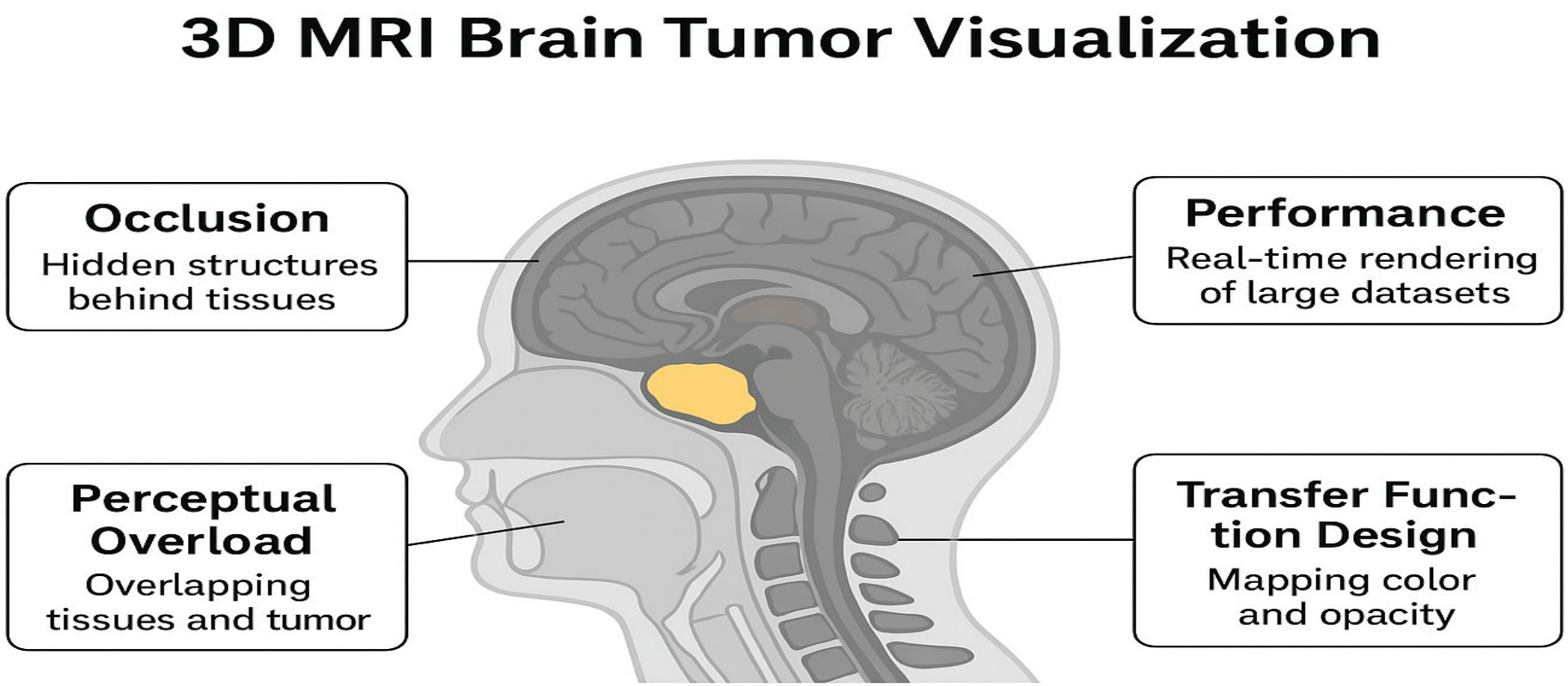
Scenario

A medical imaging team develops an interactive 3D visualization tool to examine **MRI brain scans** for tumor detection, segmentation, and surgical planning.

How Each Challenge Appears

Challenge	Explanation in This Example
1. Occlusion	The **tumor** or blood vessels may be hidden behind layers of tissue, making it hard to view internal structures clearly. 👉 Doctors must use **clipping planes** or transparency controls to reveal hidden regions.
2. Performance	MRI data involves **millions of voxels**, requiring real-time rendering while the doctor rotates or slices through the 3D volume. 👉 The system must optimize **GPU rendering** and use **level-of-detail techniques** to maintain smooth performance.
3. Perceptual Overload	The doctor views multiple overlapping tissues (gray matter, white matter, tumor, fluid). Too much information causes confusion. 👉 The interface must simplify visuals using **layer control** and color differentiation.
4. Transfer Function Design	The radiologist adjusts how tissues are mapped to **color and opacity** — e.g., transparent soft tissue, bright opaque tumor — to highlight key regions. 👉 Designing these mappings is complex because small changes can hide or distort critical details.

In a **3D MRI brain visualization**:
•**Occlusion** makes internal structures hard to see,
•**Performance** limits interactivity with large datasets,
•**Perceptual overload** affects how doctors interpret the volume,
•**Transfer function design** determines how effectively the tumor and tissues are distinguished.
Together, these challenges define the **core problems in volumetric visualization design** for real-time, high-precision applications like **medical diagnostics**.

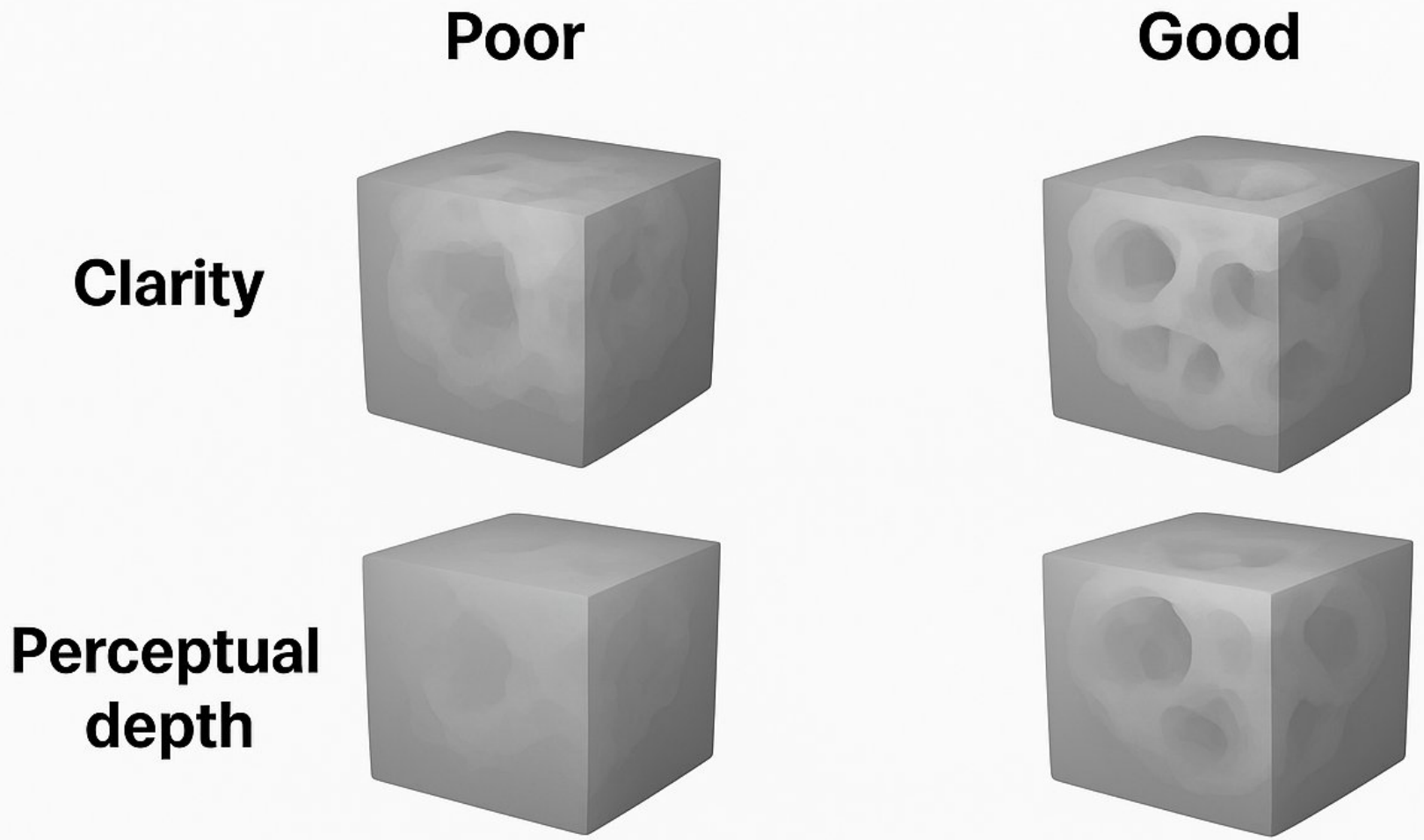


Evaluation Criteria (from Ward et al.)

To judge the effectiveness of a volumetric visualization:

1. **Clarity:** Can internal features be identified easily?
2. **Accuracy:** Are scalar values represented truthfully?
3. **Interactivity:** Can users slice, rotate, zoom fluidly?
4. **Performance:** Is rendering responsive for large datasets?
5. **Perceptual depth:** Does it preserve spatial relationships correctly?

Criterion	Description	Purpose
1. Clarity	The visualization should clearly reveal internal structures and features within the volume.	Ensures users can interpret internal patterns, boundaries, and relationships without ambiguity.
2. Accuracy	The mapping from scalar (or vector/tensor) values to visual properties (color, opacity, etc.) must be truthful.	Prevents misinterpretation of data due to distortion or misleading visual encoding.
3. Interactivity	Users should be able to fluidly manipulate the view — slice, rotate, zoom, or adjust parameters in real time.	Supports exploration and understanding of complex 3D data through hands-on engagement.
4. Performance	Rendering must remain responsive even for large or high-resolution datasets.	Maintains usability and prevents lag during analysis of dense volumetric information.
5. Perceptual Depth	Depth cues (shading, occlusion, transparency) should preserve correct spatial relationships.	Enhances spatial perception, helping users understand how features are distributed in 3D space.


Poor
Good
Clarity
Perceptual depth

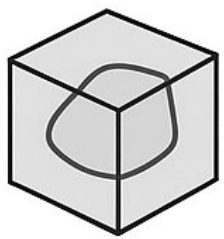
Real-World Applications

Domain	Visualization Example	Insight
Medical Imaging	MRI volume rendering	Detect internal abnormalities
Oil & Gas	3D seismic data visualization	Identify underground resources
Meteorology	Volumetric temperature maps	Understand weather fronts
Engineering CFD	Pressure/velocity fields	Optimize fluid dynamics
Astrophysics	Gas density clouds	Study stellar formation

VOLUMETRIC DATA VISUALIZATION

Scalar Fields

Isosurface Extraction



Volume Rendering

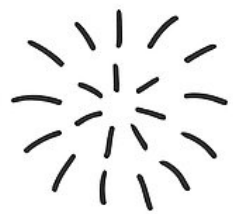


Vector Fields

Glyphs



Particle Animation



Tools and Evaluation

- ParaView, VTK, Plotly
- Evaluate: Clarity, Depth, Accuracy, Speed

A **real-time application of volumetric data visualization** is in **medical imaging — particularly MRI and CT scans**.

Example: 3D Medical Imaging (MRI/CT)

•**What it does:** Visualizes 3D structures inside the human body — organs, bones, or tissues — using volumetric rendering.

•**How it works:** Each voxel (3D pixel) represents tissue density or intensity. Volume rendering combines all these voxels to create a semi-transparent, realistic 3D view.

•**Use case:**

- Surgeons analyze brain tumors or internal injuries.
- Radiologists examine organ deformations.
- Enables non-invasive diagnosis and surgical planning.

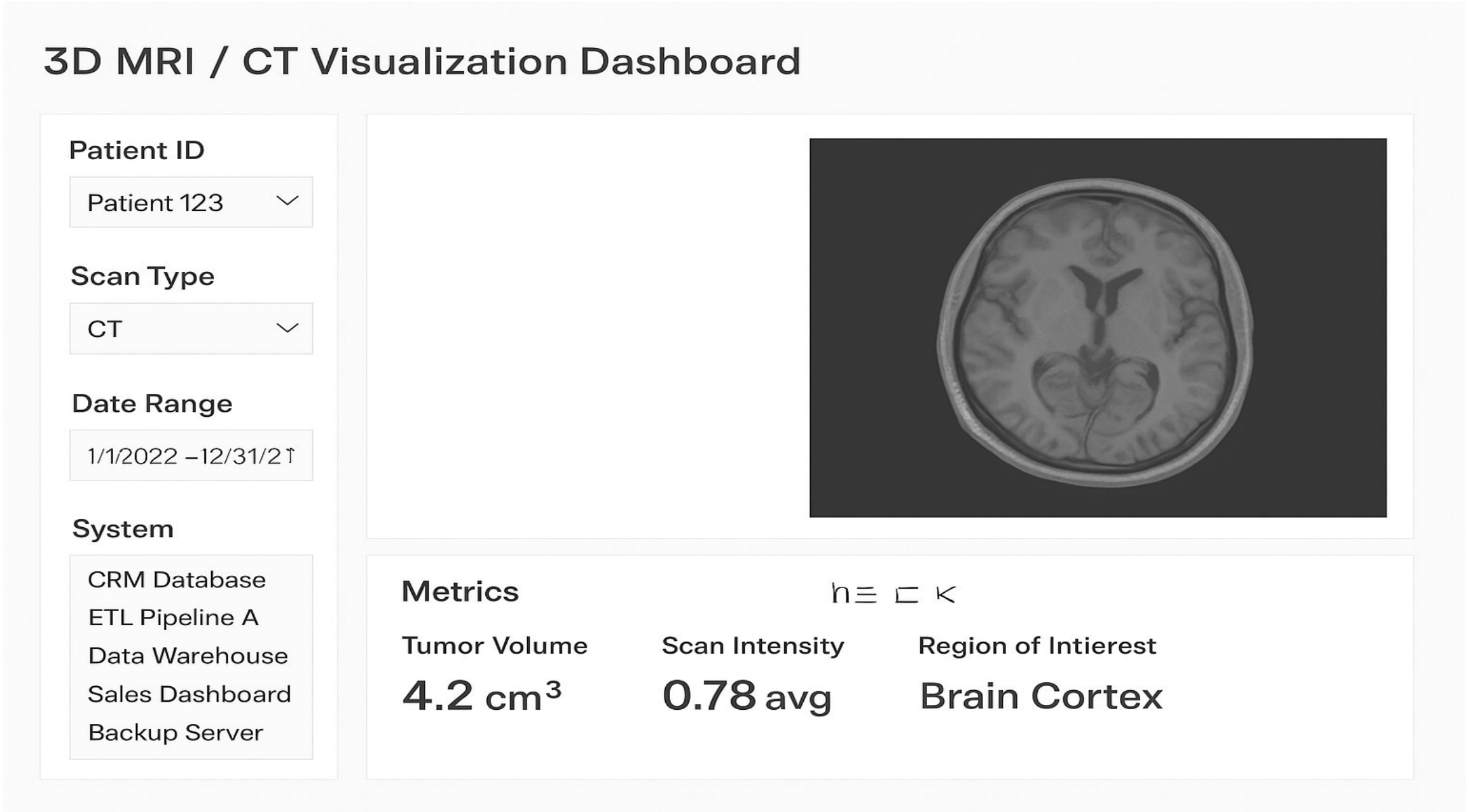
Visualization Tools Used:

•**ParaView** or **VTK** for 3D rendering.

•**Python + Plotly Volume Plots** for interactive web visualizations.

•**Power BI + 3D Map plug-ins** for integrating medical data dashboards.

Visualizes 3D structures inside the human body — organs, bones, or tissues — using volumetric rendering.



Processes and simulations

Dynamic systems evolve over time — visualizing *processes* (continuous changes) and *simulations* (computer-modeled phenomena).

Processes

- Represent **continuous changes** in a system over **time and space**.
- Example: Temperature diffusion, chemical reactions, or traffic movement.

- **Simulations**

- Are **computer-generated models** of real-world systems.
- They help visualize how systems behave under different conditions.
- Example: Weather prediction, fluid dynamics, airflow, biological processes.

Visualization of processes and simulations helps:

- Understand **time-varying (dynamic)** behavior.
- Detect **patterns, anomalies, or critical states**.
- Communicate **complex spatiotemporal relationships** easily.
- Aid in **scientific decision-making and engineering design**.

3. Types of Processes and Simulations

Type	Description	Example	Visualization Techniques
Physical Simulations	Modeled from real-world physics equations	Weather, fluid flow, heat diffusion	Volume rendering, streamlines, animation
Biological Processes	Life and health-related changes	Blood flow, neuron activity, cell growth	Particle visualization, diffusion maps
Mechanical / Industrial Simulations	Engineering or machinery modeling	Stress analysis, combustion, robot motion	Tensor fields, isosurfaces, vector fields
Environmental Simulations	Ecosystem or atmosphere modeling	Pollution spread, rainfall, tsunami	Geo-mapping, contour plots, spatiotemporal 3D plots
Computational / Algorithmic Processes	Data-driven or system simulations	Machine learning processes, network spread	Flowcharts, animated data graphs

4. Key Visualization Techniques Used

1. Time-Varying Volume Rendering

•Displays how volumetric data changes across time steps.
•Each frame = snapshot of the process at time *t*.
•Example: MRI scans of a beating heart, changing weather fronts.

2. Particle System Visualization

•Represents entities (air particles, molecules, vehicles) moving through space.
•Each particle's motion reflects simulation parameters like speed or force.
•Example: Smoke, rainfall, airflow.

3. Isosurface Animation

•Tracks changes in a constant value surface (like temperature = 100°C) over time.
•Example: Expanding heat zone or melting object.

4. Streamlines, Pathlines, Streaklines

•Used for **flow processes** (air, water, blood).
•Shows how particles or fluid elements move.
•Example: Wind pattern evolution during a cyclone simulation.

5. Spatiotemporal Visualization

•Combines **space + time dimensions** into one visual.
•Uses color, transparency, or animation to encode time.
•Example: Traffic flow through a city during the day.

6. Interactive Dashboards

•Used in business or environmental simulations.
•Combine 2D/3D charts, sliders, and maps to manipulate simulation parameters live.
•Example: Power BI dashboard simulating carbon emission reduction by year.

5. Real-Time Examples

Domain	Process/Simulation	Visualization Example
Medical	Blood flow in arteries	Animated streamline visualization in ParaView
Weather	Cloud and rainfall simulation	Volumetric time-lapse with 3D maps
Automotive	Airflow around a car	CFD simulation using isosurfaces
Geoscience	Lava or tsunami spread	3D terrain animation
Industrial	Combustion chamber process	Heat map and isovolume rendering
Environmental	Pollution dispersion	Animated contour maps in Power BI
Biological	Protein folding	Molecular simulation with volume rendering

6. Example Implementation in Power BI

You can simulate simple process visualizations using:

- **Animated scatter plots** (change over time)
- **3D Maps** for environmental or geographic simulations
- **Play Axis Visual** to animate data frames
- **Heatmap Custom Visuals** for diffusion or density visualization

Example:

“Pollution Diffusion over City (2015–2025)”

- X/Y = Latitude/Longitude
- Color = Pollution level
- Time = Year (animated)

Technique	Used For	Example Application
Volume Rendering	3D field changes	Temperature diffusion
Particle System	Moving entities	Smoke, rainfall
Streamlines / Pathlines	Flow visualization	Air or fluid flow
Isosurfaces	Boundary evolution	Pressure threshold tracking
Spatiotemporal Maps	Changes over time & space	Traffic or environmental analysis
Interactive Dashboards	Simulation control	Real-time monitoring

Visualization of maps

Map Visualization (or **Geospatial Visualization**) is the graphical representation of **data that has a geographic or spatial component** — using maps to reveal patterns, relationships, and trends.
It transforms **location-based data** into **interactive and meaningful visual forms** such as choropleth maps, heat maps, or flow maps.

Types of Map Visualizations

Type	Description	Example Use
1. Choropleth Map	Uses color shades to represent values within geographic boundaries.	Population density by state, COVID-19 cases by country
2. Heat Map	Uses color intensity to show concentration or density of events.	Traffic accident hotspots, customer activity
3. Symbol Map	Uses circles, icons, or proportional symbols to represent data magnitude.	Earthquake magnitudes, sales by city
4. Flow Map	Shows movement between locations using arrows or lines.	Migration, airline routes, trade flows
5. Dot Density Map	Each dot represents a quantity or event in a location.	Disease spread, voter distribution
6. 3D Terrain Map	Displays elevation or terrain using 3D surfaces or shading.	Geological surveys, flood risk analysis
7. Cartogram	Distorts geographic areas to emphasize a variable.	GDP or population-based country resizing
8. Interactive Map (GIS-based)	Allows zooming, filtering, and data interaction (e.g., Power BI, Tableau, Mapbox).	Real-time weather, delivery tracking, or demographic dashboards

Key Visualization Techniques

Technique	Purpose
Color Encoding	To represent magnitude, density, or category
Layering	Combine multiple datasets (roads, elevation, population)
Legends and Scales	Help interpret colors or symbol sizes
Interactivity	Allows users to zoom, pan, filter, or query data
Projection Mapping	Converts 3D Earth surface to 2D maps (Mercator, Robinson, etc.)

4. Real-Time Example: COVID-19 Spread Dashboard
Scenario:
A dashboard shows global COVID-19 data using multiple map visualizations.

Visualization	Data Insight
Choropleth Map	Shows case intensity by country using color gradient.
Dot Map	Each dot shows active outbreak locations.
Flow Map	Displays infection spread between countries.
Heat Map	Highlights regional hotspots of infection.

Evaluating visualizations.

Evaluating a visualization means assessing how effectively it helps users understand and interpret data. Ward et al. outline **key principles and criteria** for evaluating visualization systems, focusing on both **technical quality** and **human perception**.

1. Effectiveness

•**Definition:** Measures how well the visualization communicates information.

•**Aspects:**

- Can users accurately interpret the data?
- Are patterns, trends, and outliers visible?
- Does it reduce cognitive load or cause confusion?

✓ *Example:* In a heatmap, color gradients should clearly show intensity differences — not mislead the viewer with poor color choice.

⚙ **2. Efficiency**

•**Definition:** Evaluates how quickly and easily users can extract insights.

•**Aspects:**

- Is interaction smooth (zoom, filter, pan)?
- Does the design minimize unnecessary visual clutter?
- Can users reach conclusions faster compared to raw data?

✓ *Example:* Interactive dashboards with drill-down features improve efficiency by letting users explore subsets of data instantly.

👁 3. Clarity and Accuracy

•**Definition:** The visualization must present data truthfully without distortion.

•**Aspects:**

- Are visual encodings (color, size, position) accurate?
- Is the mapping consistent and intuitive?
- Are scales and legends properly labeled?

✓ *Example:* A bar chart must start its y-axis at zero to avoid exaggerating small differences.

4. Interactivity

•**Definition:** Evaluates how well the visualization supports user control and exploration.

•**Aspects:**

- Can users modify views, filter data, and explore relationships?
- Are interactions responsive and meaningful?

✓ *Example:* A 3D volumetric medical visualization that allows rotation, slicing, and zooming enhances understanding of internal structures.

🧠 5. Cognitive and Perceptual Factors

•**Definition:** Considers how human perception and cognition affect interpretation.

•**Aspects:**

- Does color choice, shape, and layout align with perceptual principles?
- Are depth cues, contrasts, and groupings perceptually intuitive?

✓ *Example:* Using consistent colors for categories helps users track groups easily.

⚡ 6. Performance

- **Definition:** Measures computational responsiveness and scalability.
- **Aspects:**
  - Can large datasets be rendered efficiently?
  - Does interaction remain smooth with complex data?

✔ *Example:* Volume rendering should remain interactive even for large MRI or terrain datasets.

📈 7. Insight and Utility

- **Definition:** Determines whether users gain valuable, actionable insights.
- **Aspects:**
  - Does it help in decision-making or discovery?
  - Are new patterns revealed that were not obvious before?

✔ *Example:* A network visualization revealing hidden community structures.

Criterion	Focus	Example
Effectiveness	Communication quality	Clear trend visibility
Efficiency	Speed and ease of understanding	Interactive dashboard
Clarity & Accuracy	Truthful representation	Proper scaling, labeling
Interactivity	User engagement	Zoom, filter, rotate
Cognitive & Perceptual	Human factors	Good color contrast
Performance	Responsiveness	Fast rendering
Insight & Utility	Knowledge gained	Pattern discovery

Recent trends in various perception techniques

Recent advancements in perception techniques span various domains, including robotics, neuroscience, human-computer interaction, and data visualization. Here's an overview of the latest trends:

🤖 **Robotics & Autonomous Systems**

•**Sensor Fusion & AI Integration**: Combining data from multiple sensors like LiDAR, cameras, and radar enhances environmental perception. Integrating AI and machine learning enables real-time adaptation and improved decision-making in dynamic environments TechNexion.

•**Miniaturization of Sensors**: Advancements in sensor technology have led to the development of smaller, cost-effective sensors, allowing for more compact and efficient robotic systems TechNexion.

•**5G and Cloud Integration**: The rollout of 5G networks facilitates faster data transmission, enabling real-time processing and remote control of robotic systems through cloud-based platforms TechNexion.

Neuroscience & Cognitive Perception

•**Unconscious Perception Studies**: Emerging methodologies like regression-based Bayesian modeling and sensitivity vs. awareness curves are being employed to explore unconscious perception, emphasizing the need for brain-based approaches

•**Neural Imaging in Social Interactions**: Techniques such as functional near-infrared spectroscopy (fNIRS) are being used to study neural responses during natural conversations, providing insights into the coordination of movements and speech W

🖥 **Human-Computer Interaction & Visualization**

•**Immersive Visualization**: Advancements in immersive visualization focus on multimodal perception and interaction, enhancing user experience in virtual environments ScienceDirect.

•**AI-Driven Data Visualization**: AI tools are increasingly used to predict real-time opportunities and risks, improving decision-making processes through enhanced data visualization mokkup.ai.

Autonomous Driving & Perception Systems

•**Vision-Based Perception Solutions**: AI algorithm upgrades have led to the adoption of vision-based perception systems in autonomous vehicles, reducing hardware costs while maintaining performance Omdia.

•**Superhuman Perception**: Research is focused on developing sensor concepts that mimic human-like perception strategies, augmented by additional sensors like RADAR or LiDAR, to navigate environments shared with humans Bosch Global.

📊 Data Compression & Perceptual Coding

•**Perceptual Coding**: This technique exploits the limitations of human sensory systems to reduce data size without significantly affecting perceived quality. It's widely applied in audio, image, and video compression standards