

Report: Vector Field Visualization

1. Interface Description

The provided C++ code implements a vector field visualization interface that allows users to:

- Load a CSV file containing vector data (positions, directions, timestamps).
- Select a specific timestamp via console input to filter time-dependent vectors.
- Generate ensemble-based uncertainty glyphs using synthetic perturbations of each base vector.
- Perform Principal Component Analysis (PCA) and compute data depth metrics for visual encoding.
- Export the resulting visualizations as `.vtk` files for rendering in tools like ParaView or VisIt.

Key Features of the Interface:

- File Selection Dialog: Uses Windows API (`GetOpenFileName`) to allow users to select an input CSV file.
- Timestamp Filtering: Users enter a timestamp at runtime to extract only those vectors associated with that time step.
- Ensemble Generation: For each selected vector, the program generates an ensemble of 9 perturbed vector directions, with the last two being intentionally elongated (scaled by 2.0) to represent high-magnitude outliers.
- Glyph Construction: Custom glyph geometry is generated based on:
 - Principal component directions (orientation).
 - Angular spread (glyph width and depth).
 - Data depth (scaling factor).

Output Files:

- `raw_vectors.vtk`: Displays raw ensemble vectors as lines for validation.
 - `glyphs_timestamp_<X>.vtk`: Visualizes the final uncertainty-aware glyphs for a given timestamp.
-

2. Figure / Results Description

We provide a small-scale demo using synthetic vector data to validate the implementation and visualize differences in uncertainty representation.

- Two sample figures are generated:
 - Figure A : Raw ensemble vectors centered at positions
 - Figure B : Final glyphs built from the same ensembles, showing directional spread and magnitude scaling.



Figure A



Figure B