Basic Data Representations for Visualization
Data Representations

- There are many ways to represent datasets
- Points (e.g., 3D raster, point cloud)
- Lines
- Vectors
- These are all discrete data representations
- Data can be regular or irregular
- Regular = relationship exists between data points
- Compare: 3D raster vs. point cloud
- Data also has dimension: 1, 2, 3, ..., n, ....
Dataset = Structure + Attributes

- **Structure** = topology and geometry
- **Topology** refers to characteristics unchanged by transformations (holes, handles, branches)
- **Geometry** refers to \((x,y,z)\) positions of data points

- Here **cells** define topology, **points** define geometry
- There could be a large variety of different **cell types**
- Linear cell types and non-linear cell types
Cell Topology (Connectivity)

(a) Vertex

(b) Polyvertex

(c) Line

(d) Polyline (n lines)

(e) Triangle

(f) Triangle strip (n triangles)

(g) Quadrilateral

(h) Pixel

(i) Polygon (n points)
(j) Tetrahedron

(k) Hexahedron

(l) Voxel

(m) Wedge

(n) Pyramid
(a) Quadratic Edge

(b) Quadratic Triangle

(c) Quadratic Quadrilateral

(d) Quadratic Tetrahedron

(e) Quadratic Hexahedron
Cell Example: Hexahedron

- Vertices listed in special order define topology

Definition:
Type: hexahedron
Connectivity: (8,10,1,6,21,22,5,7)
Non-Linear Cell Decomposition

- Non-linear cells must be linearized for visualization
- Break non-linear cells into linear cells
Non-Linear Cell Decomposition

- Quadratic Edge
- Two lines
- Quadratic Triangle
- Four triangles
- Quadratic Quadrilateral
- Four quadrilaterals
Attribute Data

- Data values (attributes) usually assigned to vertices, as opposed to edges or faces
- Why?
- Interpolation concept easy to apply across edges and faces
- Common attributes include:
  - Temperature, density, velocity, pressure, heat flux, chemical concentration, others
- Scalars, vectors, tensors
Attribute Data

- **Scalar** data is data that is single-valued at all locations in a data-set
- **Examples**: temperature, stock price, elevation
- **Vector** data is data with magnitude and direction
- **Examples**: position, velocity, acceleration
- **Normals** (direction vectors) are vectors of magnitude 1
- **Texture coordinates** map a point from Cartesian space into a 1-D, 2-D or 3-D texture space
- **Textures** let us add color, transparency and other details to geometric shapes
Attribute Data

- **Tensors** are mathematical generalizations of vectors and scalars
- Usually written as matrices
- Tensor visualization is extremely difficult
Types of Data-sets

- Regular vs. irregular structure – refers to topology of data-set
- Data-sets with regular topology, we do not need to store connectivity information
- Points themselves can be regular or irregular
- If irregular, we need to store the positions
- Unstructured data must be explicitly represented
- High computational and storage costs usually
(a) Image Data

(b) Rectilinear Grid

(c) Structured Grid

(d) Unstructured Points

(e) Polygonal Data

(f) Unstructured Grid
Polygonal Data

• Vertices, edges, polygons, polylines, triangle strips, etc.

• Triangle strips can represent $n$ triangles using only $n+2$ points, vs. $3n$ points normally required
Image Data

• Collection of points and cells on a regular, rectangular grid
• Also called a “raster”
• (Book uses word “lattice” – avoid!)
• 2D grid → image
• 3D grid → volume
• $i$-$j$-$k$ coordinate system parallel to global $x$-$y$-$z$ coordinate system
• Simple representation, but “curse of dimensionality”
Rectilinear Grid

- Regular grid, but spacing along axes can vary
- Need to store 3 extra arrays of length $n_x$, $n_y$, $n_z$ — dimensions of the grid
- Each array stores spacing, basically
Structured Grid

• Regular topology, irregular geometry
• Curvilinear grids most common type
Unstructured Points

- No topology, irregular geometry
- Also called point clouds
Unstructured Grid

- Irregular topology and geometry
- Any combination of cells permitted
- Encountered in relatively few applications
- e.g., computational geometry
VTK Data Representations

- vtkFloat Array
- vtkImageData
- vtkRectilinearGrid
- vtkStructuredGrid
- vtkPolyData
  - vtkCellArray
- vtkUnstructuredGrid

Figure 5–13 The data structure of the class vtkUnstructuredGrid. (This is a subset of the complete structure. See Chapter 8 for complete details.)
VTK Data Representations

Figure 5–14 Dataset object diagram. The five datasets (shaded) are implemented in VTK.
Figure 5–15 Object diagram for twenty concrete cell types in VTK. vtkEmptyCell represents NULL cells. vtkGenericCell can represent any type of cell. Three-dimensional cells are subclasses of vtkCell3D. Higher order cells are subclasses of vtkNonLinearCell.
Example: Cube.cxx

```cpp
tvtkPolyData *cube = vtkPolyData::New();
tvtkPoints *points = vtkPoints::New();
tvtkCellArray *polys = vtkCellArray::New();
tvtkFloatArray *scalars = vtkFloatArray::New();

for (i=0; i<8; i++) points->InsertPoint(i, x[i]);
for (i=0; i<6; i++) polys->InsertNextCell(4, pts[i]);
for (i=0; i<8; i++) scalars->InsertTuple1(i, i);

cube->SetPoints(points);
points->Delete();
cube->SetPolys(polys);
polys->Delete();
cube->GetPointData()->SetScalars(scalars);
scalars->Delete();
```

Figure 5–17  Creation of polygonal cube (Cube.cxx).
Example: Vol.cxx

vtkImageData

vtkContourFilter

vtkPolyDataMapper
Example: SGrid.cxx
Example: RGrid.cxx

```plaintext
vtkRectilinearGrid
  └── vtkRectilinearGridGeometryFilter
    └── vtkPolyDataMapper
```
Example: UGrid.cxx

\[ \text{vtkUnstructuredGrid} \rightarrow \text{vtkDataSetMapper} \]
From 3D data clouds to surface meshes: Triangulation of data sets
Mesh Objects
Why Triangular Meshes are Needed?

• A simple piecewise linear approximation of 3D shapes of complex objects
• Appropriate for processing in graphics hardware
• Suitable for deformation and manipulation of the object surfaces
Main Topics

• Planar triangulation
  – Voronoi diagram
  – Delaunay triangulation

• 3D triangulation based on a physical model
  --- balloon inflation

• Marching cubes
Planar Triangulation

Diagram of a planar triangulation.
Voronoi Diagrams
Dual Graph of a Voronoi Diagram
Triangulation of Terrain Data
Edge Flipping

Figure 9.4
Flipping an edge
Incremental Triangulation Algorithm
Progressive Balloon Inflation

• Balloon inflation for surface fitting
Physical Model for Balloon Inflation

The Direction of the Inflation Force

The Tension Force between i and j

The Normal Vector of the Triangle
Subdivision of Triangular Faces

(a) Subdivision of a triangular face into smaller triangles.

(b) Further subdivision of the triangles from (a).

(c) Even more subdivisions.

(d) Final subdivision into smaller triangles.
Touching of Balloon at Data

The Ray of the Movement

Approximate Intersection Object Surface

Tangent Plane
Approximating Errors
Adaptation of Local Fitting
Subdivision for Fitting

New Triangles

The Center of Gravity
Hierarchy of Triangular Meshes

Meshes with triangles of different sizes
Contouring Using Marching Squares
Marching Squares Cases
Ambiguity in Connecting Edges

(a) Break contour

(b) Join contour
Marching Cubes
Marching Cubes Cases

Case 0  Case 1  Case 2  Case 3

Case 4  Case 5  Case 6  Case 7
Marching Cubes Cases

Case 8

Case 9

Case 10

Case 11

Case 12

Case 13

Case 14
Ambiguity in Connecting Edges

Case 3

Case 6c
An Example of Extracted Isosurfaces

(b) Isosurface of human skull
Marching Cubes for 3-D Data Clouds
An Example of Extracted Meshes