Data and Attributes

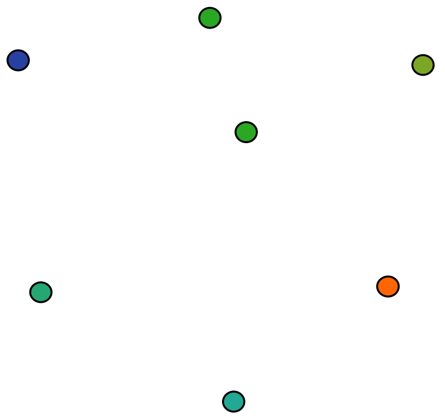
Visualisation – Lecture 5

Taku Komura

Institute for Perception, Action & Behaviour
School of Informatics

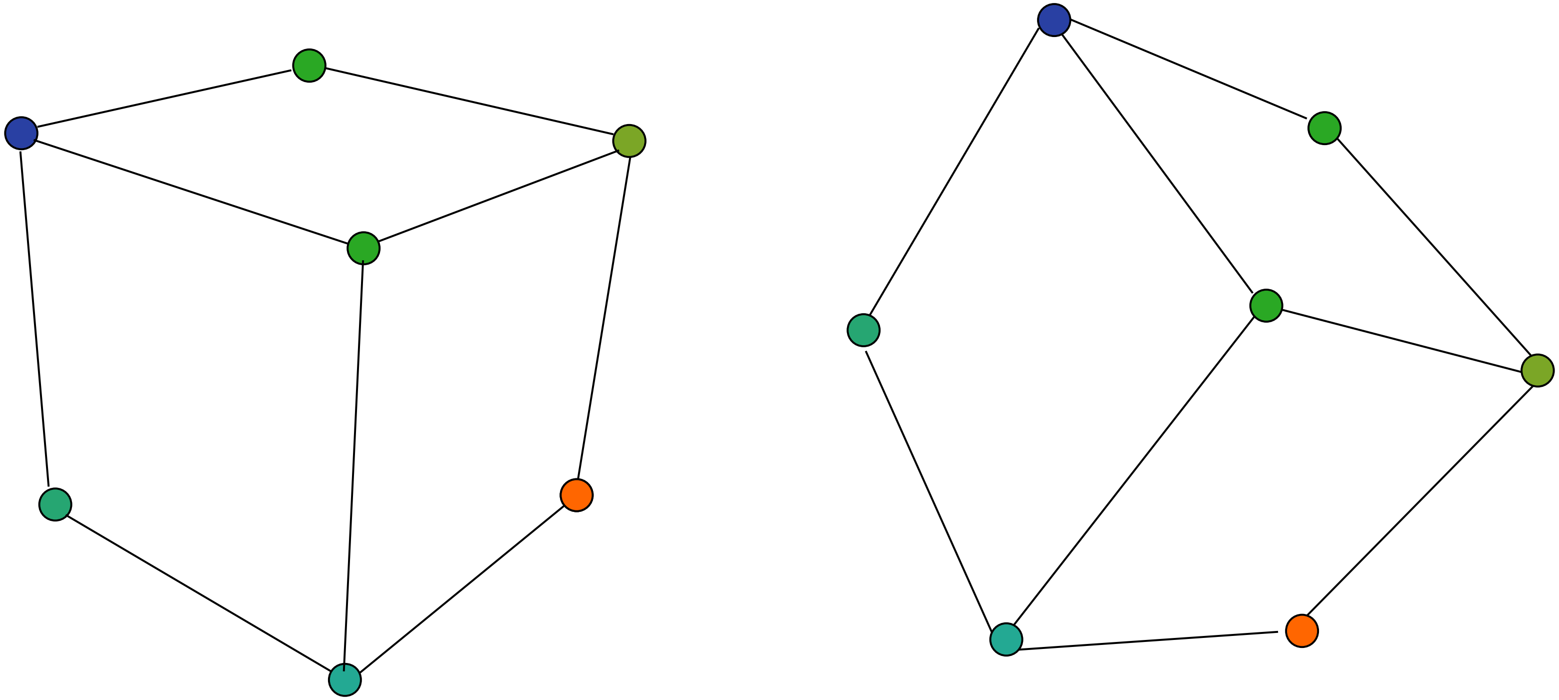
Discrete Vs. Continuous

- Real World is **continuous**
  - eyes designed towards the perception of continuous shape
- Data is **discrete**
  - finite resolution representation of a real world (of abstract) concept
- *Difficult to visualise continuous shape from raw discrete sampling*
  - *we need **topology** and **interpolation***



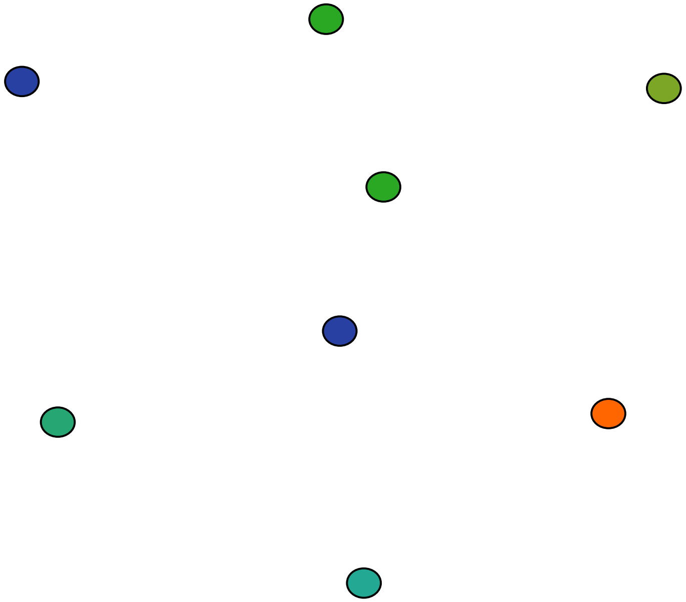
Topology

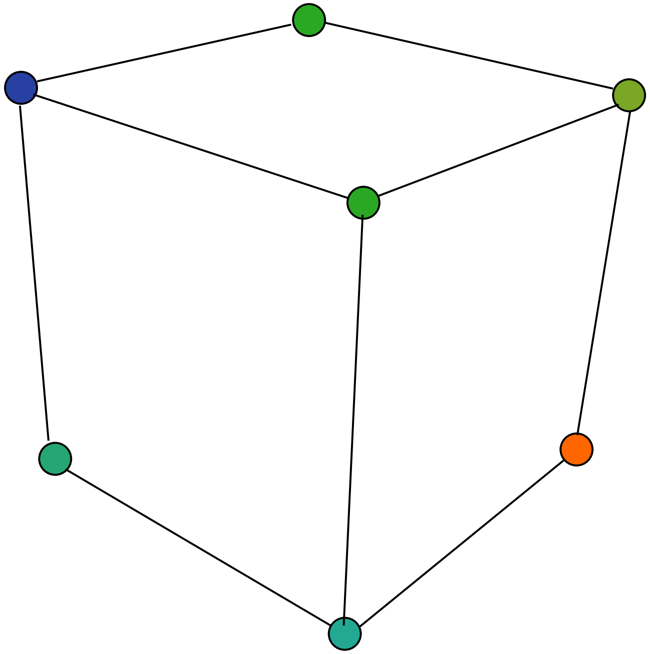
- **Topology :** relationships within the data invariant under geometric transformation
  - Which vertex is connected with which vertex by an edge?
  - Which area is surrounded by which edges?



Interpolation & Topology

- If we introduce **topology** our visualisation of discrete data improves

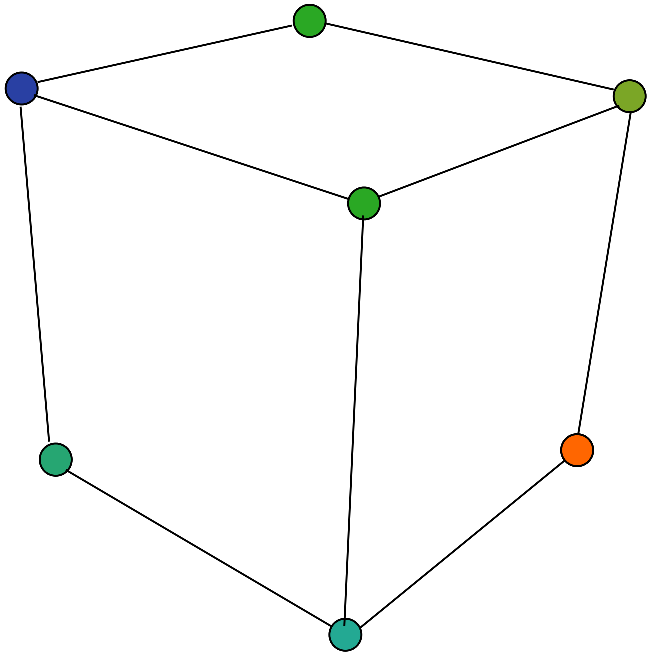


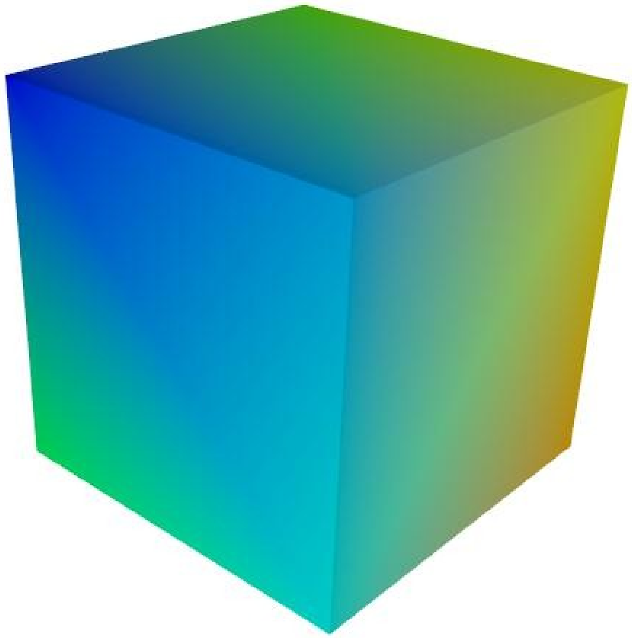


  - **why ?** : because then we can **interpolate** based on the topology

Interpolation & Topology

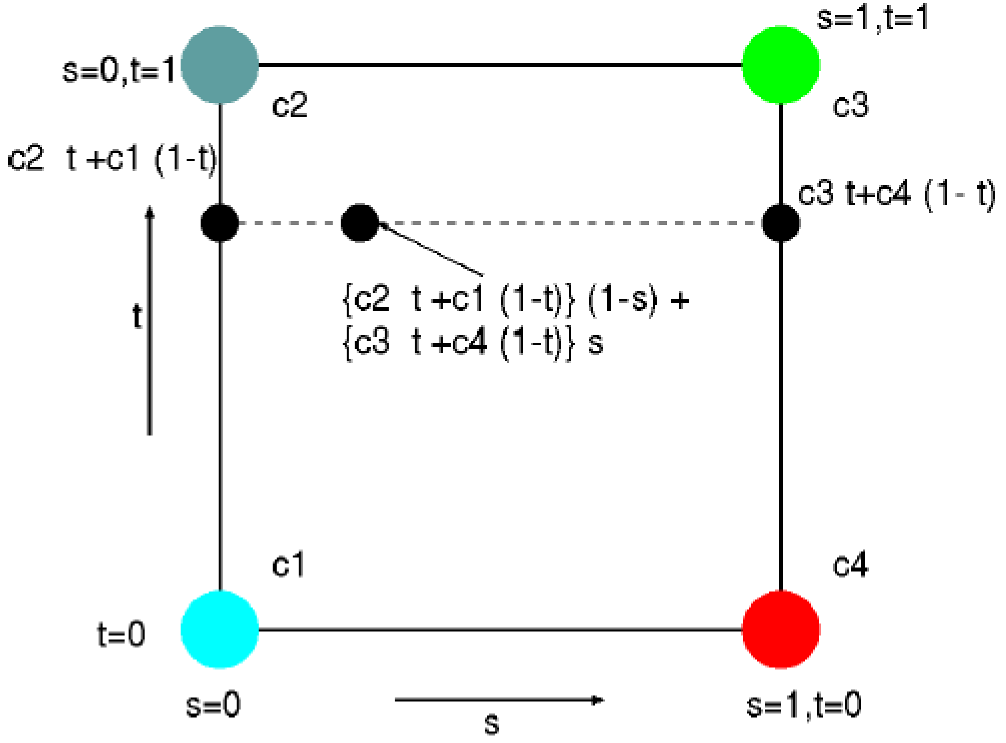
- Use interpolation to shade whole cube:





  - **Interpolation:** producing intermediate samples from a discrete representation

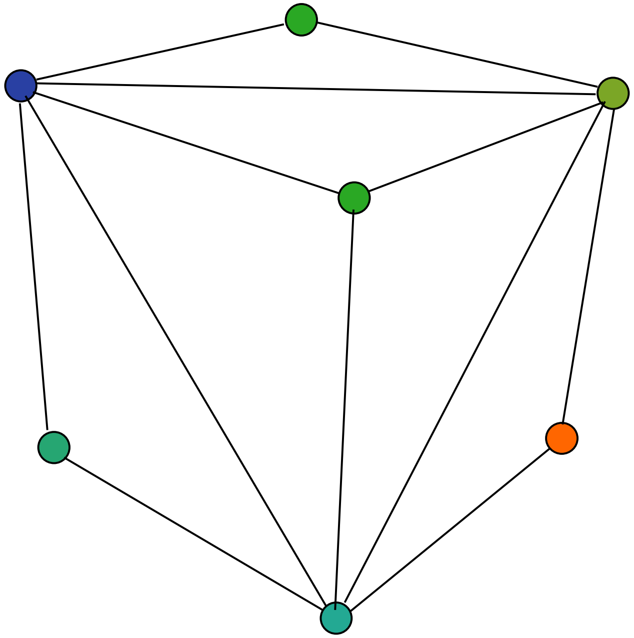
How to interpolate over a rectangle?



- Will the red color dot on the right lower corner affect the color of the point near the left top?

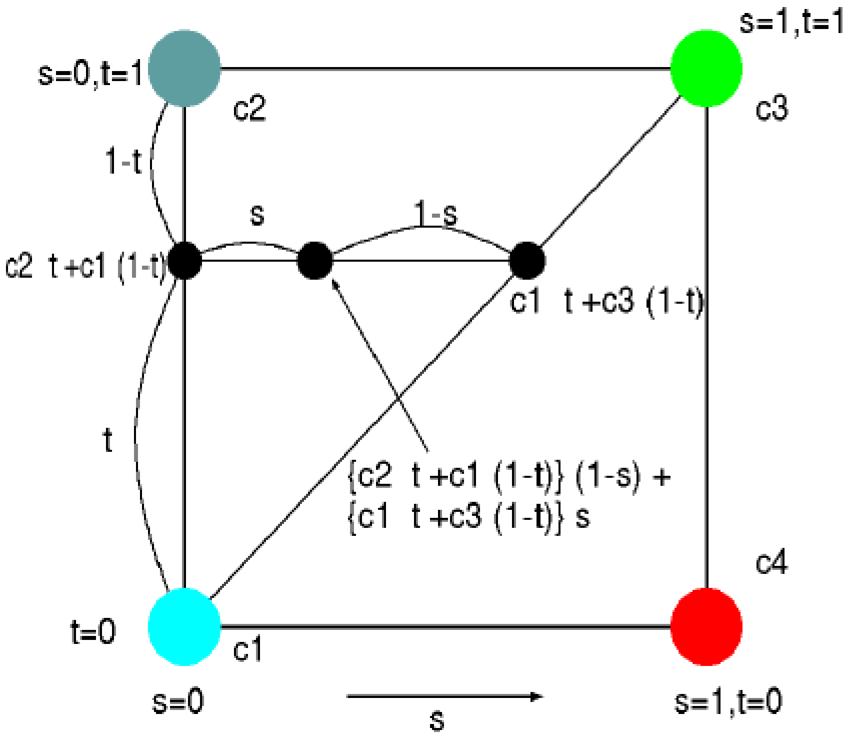
Importance of representation

- What happens if we change the representation?



- Discrete data samples remain the same
  - topology has changed ⇒ effects interpolation ⇒ effects visualisation

How to interpolate in this case?

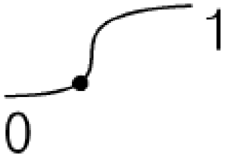


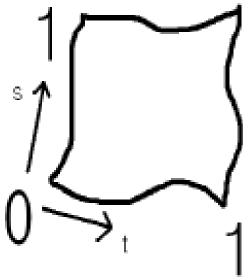
- Will the red color dot on the right lower corner affect the color of point near the left top?

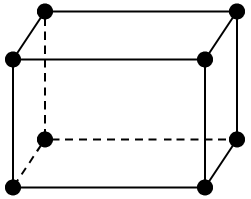
Topological Dimension

- **Topological Dimension:** number of independent continuous variables specifying a position within the topology of the data
  - different from **geometric dimension** (position within general space)

	Topological	**Geometric**	
point	0D	2D/3D	e.g. 2D (x,y)
curve	1D	2D/3D	e.g. 3D point on curve
surface	2D	3D (in general)	
volume	3D	3D	e.g. MRI or CT scan

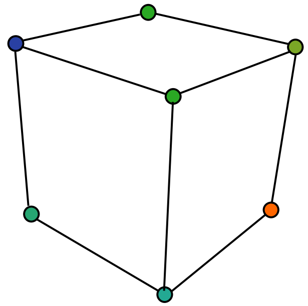


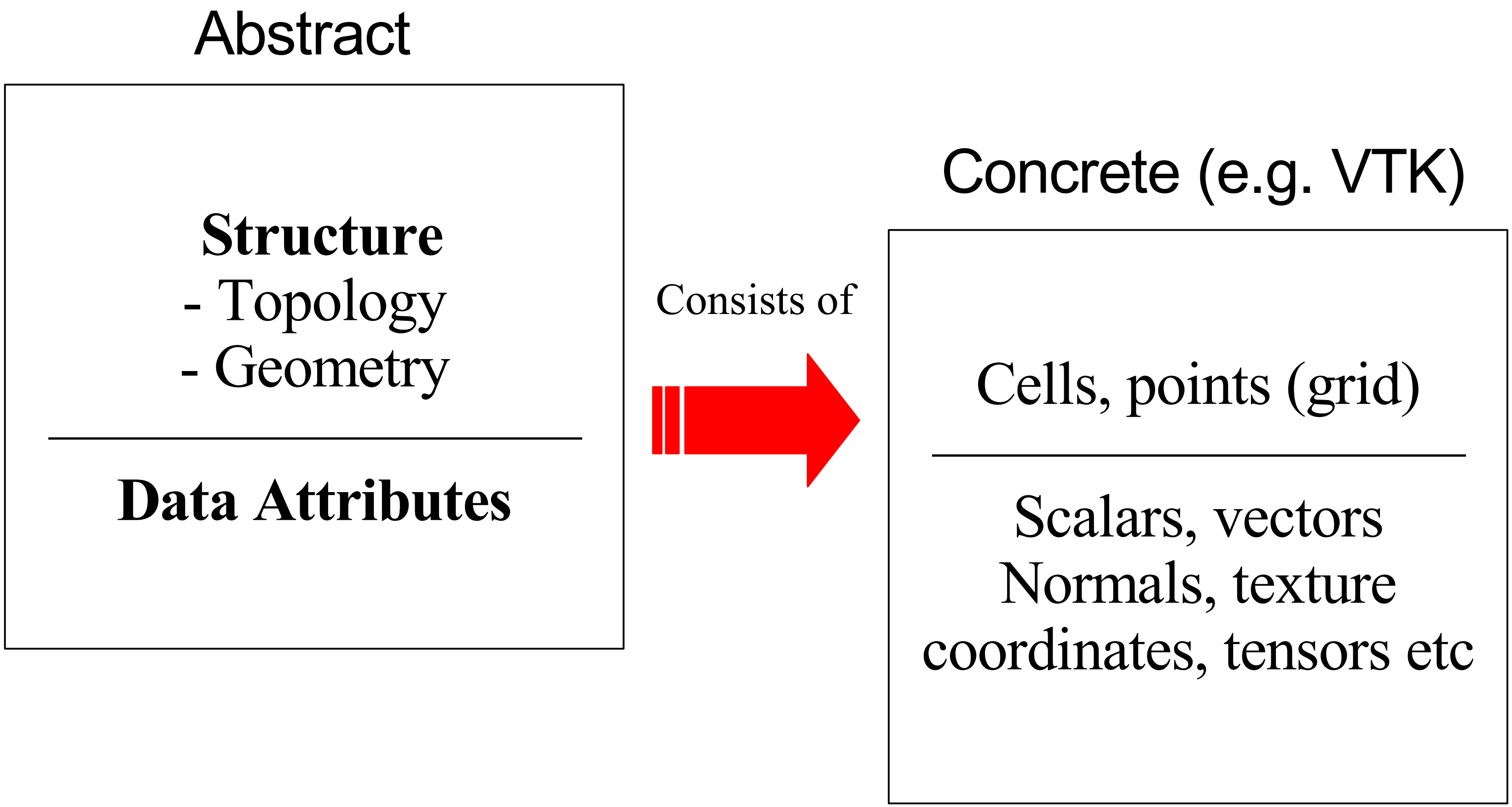




Data Representation

- **Data objects : structure + value**
  - referred to as ***datasets***





Abstract

Structure
- Topology
- Geometry

Data Attributes

Consists of

Concrete (e.g. VTK)

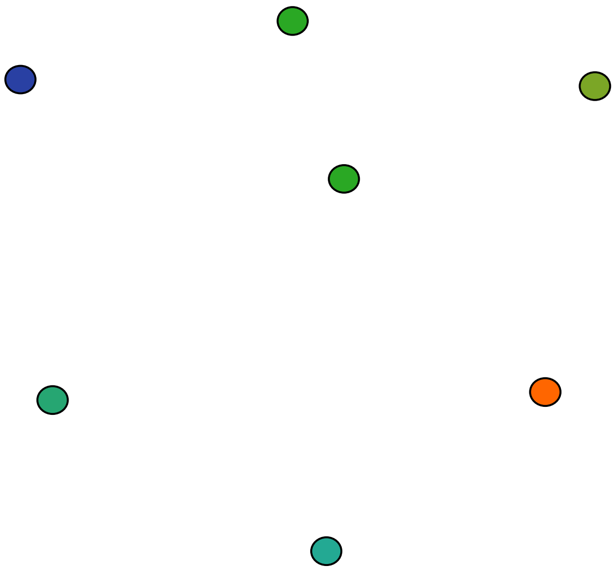
Cells, points (grid)

Scalars, vectors
Normals, texture
coordinates, tensors etc

What is a dataset?

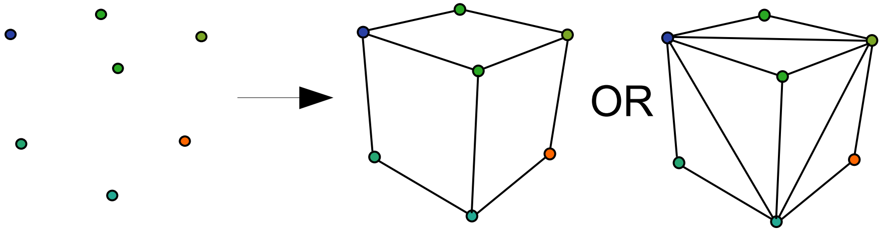
- **Dataset** consists of 2 main components
  - **structure** of the data
  - **value –** attributes associated to particular parts of the structure
  - **structure gives spatial meaning to the attributes**

```
values = {   blue, green, green,
             green, green, green,
             turquoise, red}
```

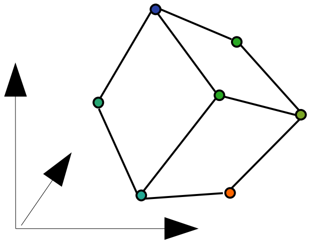


Structure of Data

- **Structure** has 2 main parts
  - **topology :** “set of properties invariant under certain geometric transformations”
    - determines interpolation required for visualisation
    - **“shape” of data**

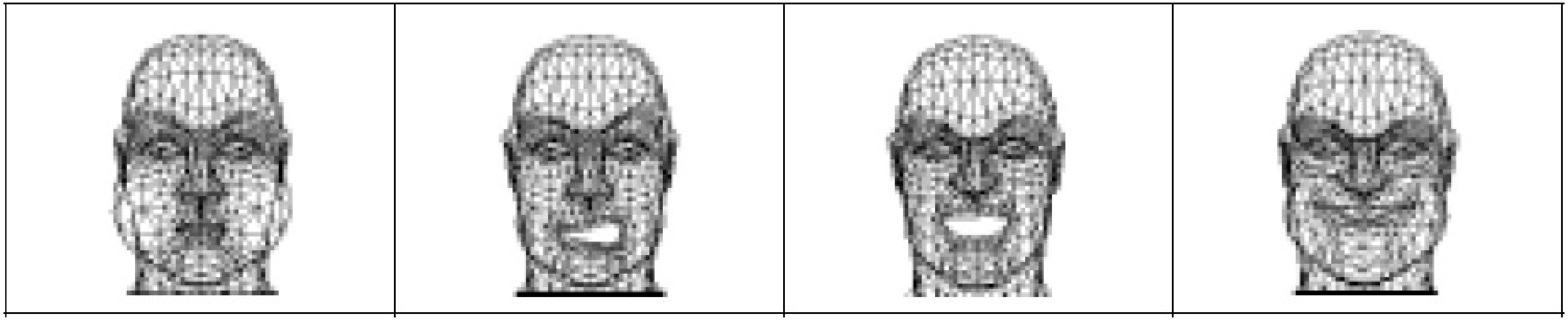


  - **geometry**
    - instantiation of the topology
    - specific position of points in geometric space



Example

- This is a generic face model
- The geometry are different, but the topology are the same

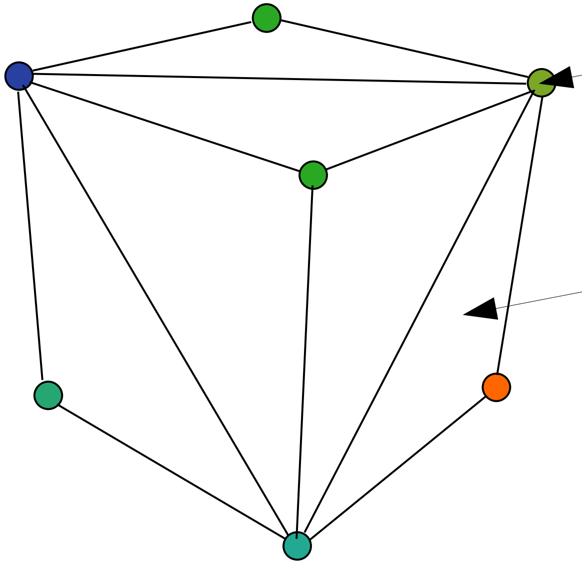


- Just the geometry of each vertex is changed

Representation Datasets the Visualisation Pipeline

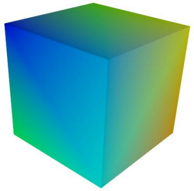
- **Points** specify where the data is known
- **Cells** allow us to interpolate between points
  - **specify topology** of points

- *Follows VTK model*



Point with known attribute data
(i.e. colour ≈ temperature)

Cell (i.e. the triangle) over which we can interpolate data.



Cells

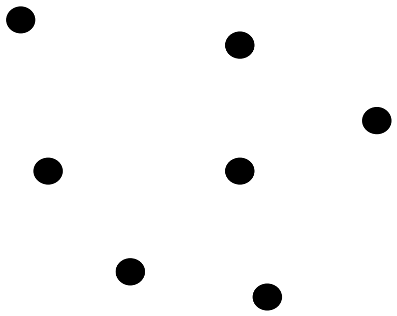
- **Fundamental building blocks of visualisation**
  - *our gateway from discrete to interpolated data*

- Various Cell Types
  - defined by **topological dimension**
  - specified as an ordered point list (connectivity list)
  - primary or composite cells
    - composite : consists of one or more primary cells

Zero-dimensional cell types

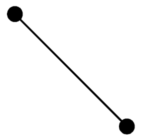
- **Vertex**
  - **Primary** zero-dimensional cell
  - Definition: single point

- **Polyvertex**
  - **Composite** zero-dimensional cell
    - composite : comprises of several vertex cells
  - Definition: arbitrarily ordered set of points

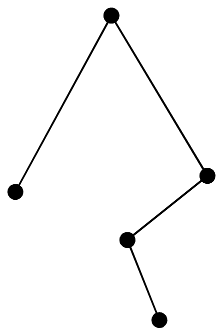


One-dimensional cell types

- **Line**
  - **Primary** one-dimensional cell type
  - Definition: 2 points, direction is from first to second point.

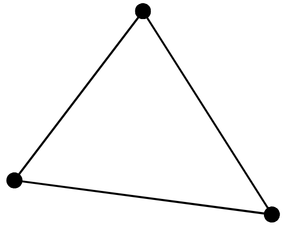


- **Polyline**
  - **Composite** one-dimensional cell type
  - Definition: an ordered set of *n+1* points, where *n* is the number of lines in the polyline

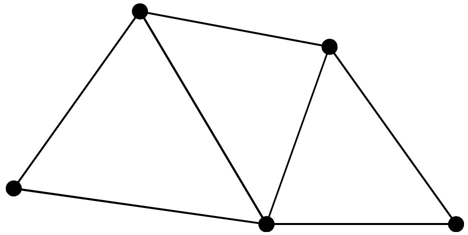


Two-dimensional cell types - 1

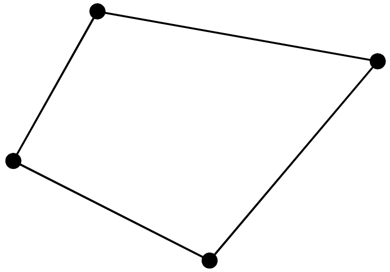
- **Triangle**
  - **Primary** 2D cell type
  - Definition: counter-clockwise ordering of 3 points
    - order of the points specifies the direction of the surface normal



- **Triangle strip**
  - **Composite** 2D cell consisting of a strip of triangles
  - Definition: ordered list of *n*+2 points
    - n is the number of triangles

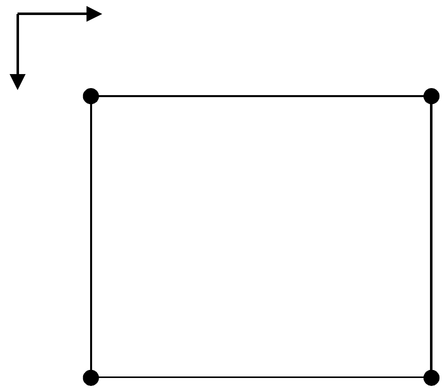


- **Quadrilateral**
  - **Primary** 2D cell type
  - Definition: ordered list of four points lying in a plane
    - constraints: convex + edges must not intersect
    - ordered counter-clockwise defining surface normal

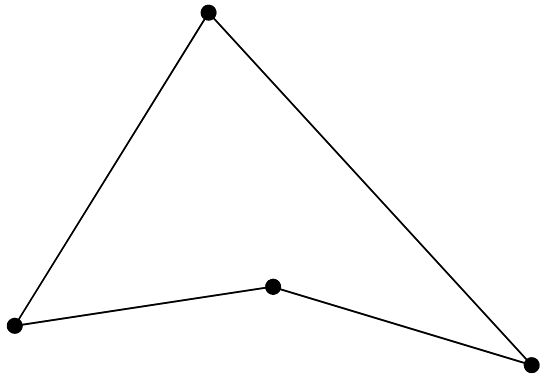


Two-dimensional cell types - 2

- **Pixel**
  - **Primary** 2D cell, consisting of 4 points
    - topologically equivalent to a quadrilateral
    - constraints: perpendicular edges; axis aligned
  - numbering is in increasing axis coordinates

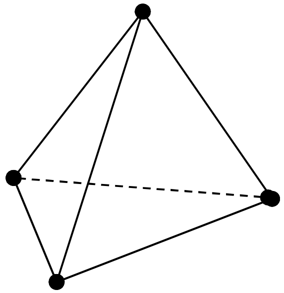


- **Polygon**
  - **Primary** 2D cell type
  - Definition: ordered list of 3 or more points
    - constraint: may not self-intersect

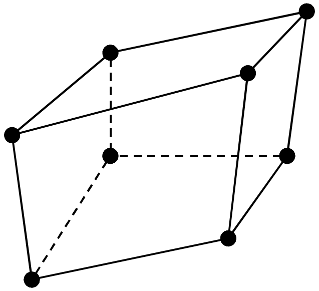


Three-dimensional cell types

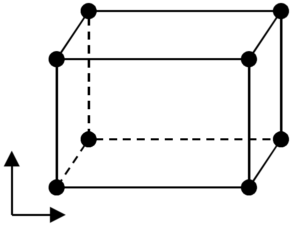
- **Tetrahedron**
  - Definition: list of 4 non-planar points
    - Six edges, four faces



- **Hexahedron**
  - Definition: ordered list of 8 points
    - six quadrilateral faces, 12 edges, 8 vertices
    - constraint: edges and faces must not intersect



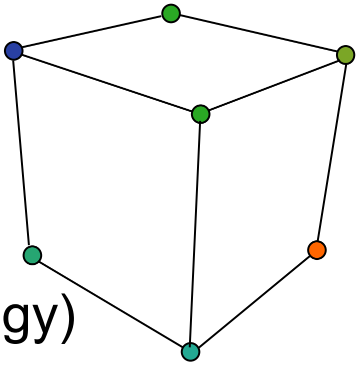
- **Voxel**
  - Topologically equivalent to Hexahedron
  - constraint: each face is perpendicular to a coordinate axis
  - “3D pixels”



Example Applications & Cell Types

- Lines
  - digitised contour data from maps
- Pixels
  - images, regular height map data
- Quadrilaterals
  - Finite element analysis
- Voxels : “3D pixels”
  - volume data : medical scanners

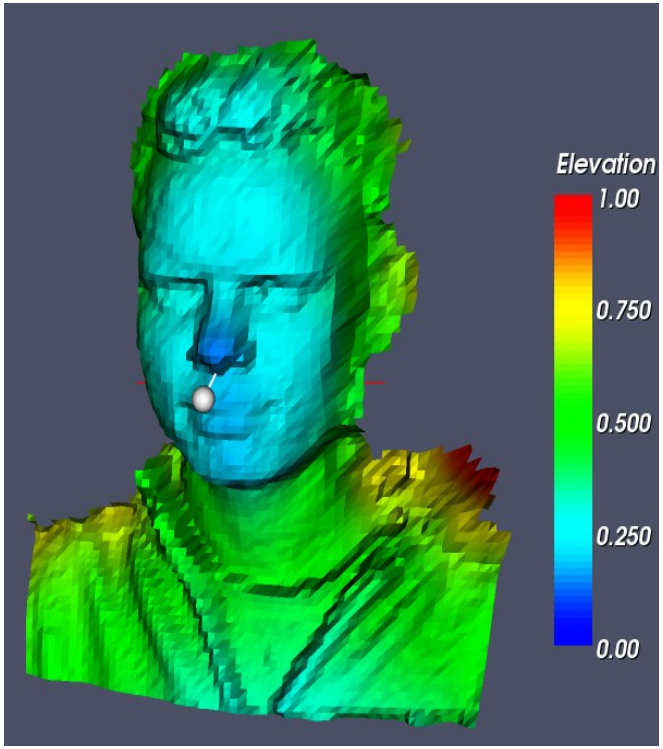
Attribute Data



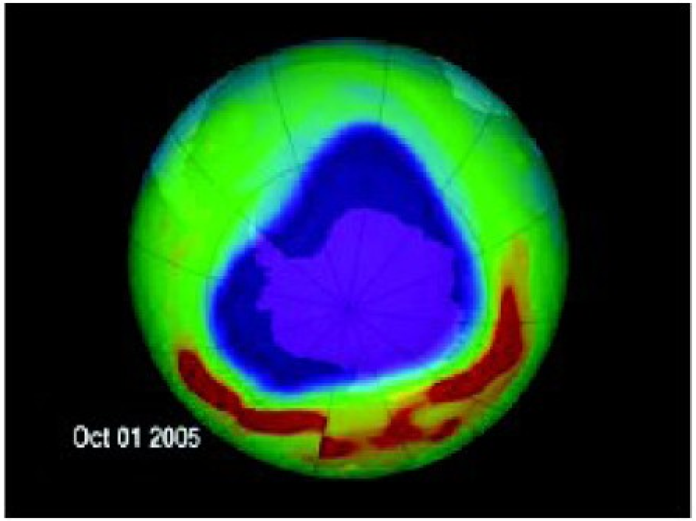
- **Information associated with data**
  - usually associated to points or cells
- Examples :
  - temperature, wind speed, humidity, rain fall (meteorology)
  - Heat flow, stress, vibration (engineering)
  - texture co-ordinates, surface normal, colour (computer graphics)
- Usually categorised into specific types:
  - **scalar** (1D)
  - **vector** (commonly 2D or 3D, or even more)
  - **tensor** (N dimensional array)

Attribute Data : Scalar

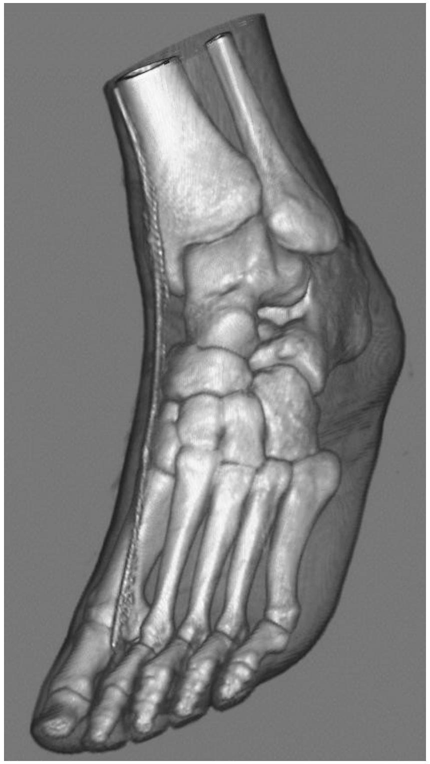
- **Single valued** data at each location



elevation from reference plane



ozone levels

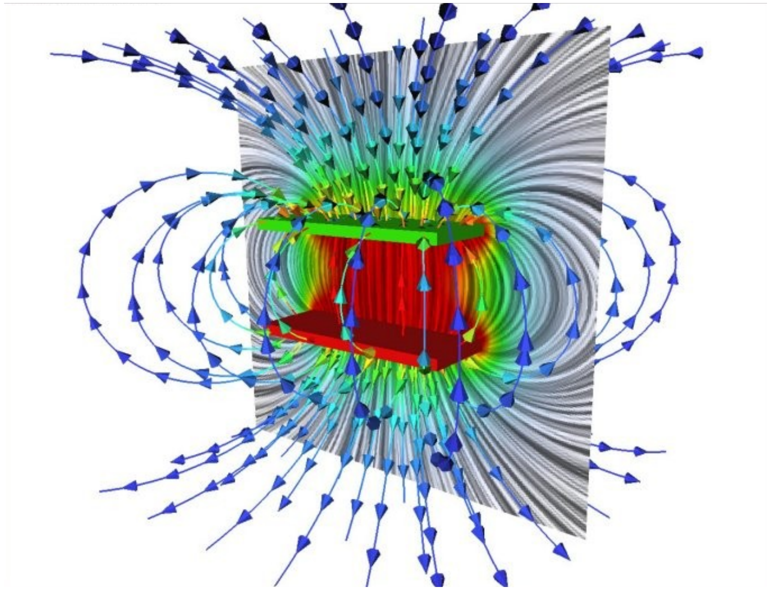


volume density (MRI)

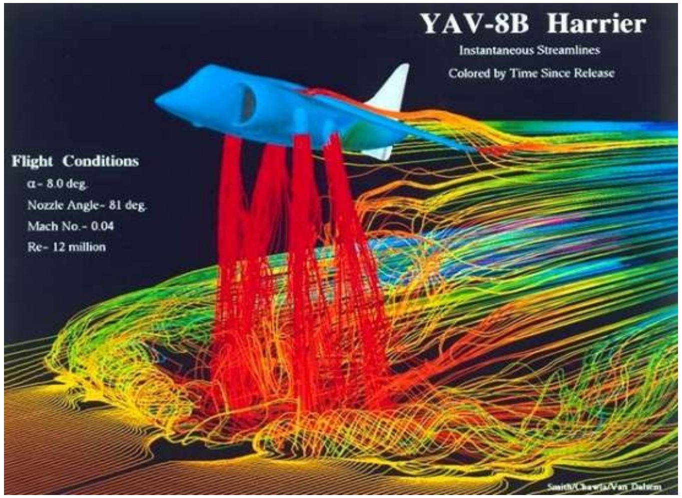
  - simplest and most common form of visualisation data

Attribute Data : Vector Data

- **Magnitude and direction** at each location
  - 3D triplet of values *(i, j, k)*



Magnetic Field



Wind flows

Attribute Data : Tensor Data

- **K-dimensional array** at each location
  - Generalisation of vectors and matrices
  - Tensor of rank *k* can be considered a *k*-dimensional table
    - Rank 0 is a scalar
    - Rank 1 is a vector
    - Rank 2 is a matrix
    - Rank 3 is a regular 3D array
- Tensor visualisation is a bit difficult
  - Need to show high dimensional data at every point
  - covered later in lectures

Visualisation Algorithms

- Generally, classified by attribute type
  - **scalar algorithms** (e.g. colour mapping)
  - vector algorithms (e.g. glyphs)
  - tensor algorithms (e.g. tensor ellipses)

Summary

- Need for **topology in data**
- **Datasets** : **structure + value**
  - structure **= topology** & **geometry**
  - value **= attribute data**
  - **cell types** in visualisation pipeline
- Types of **Attribute Data**
  - scaler, vector, tensor