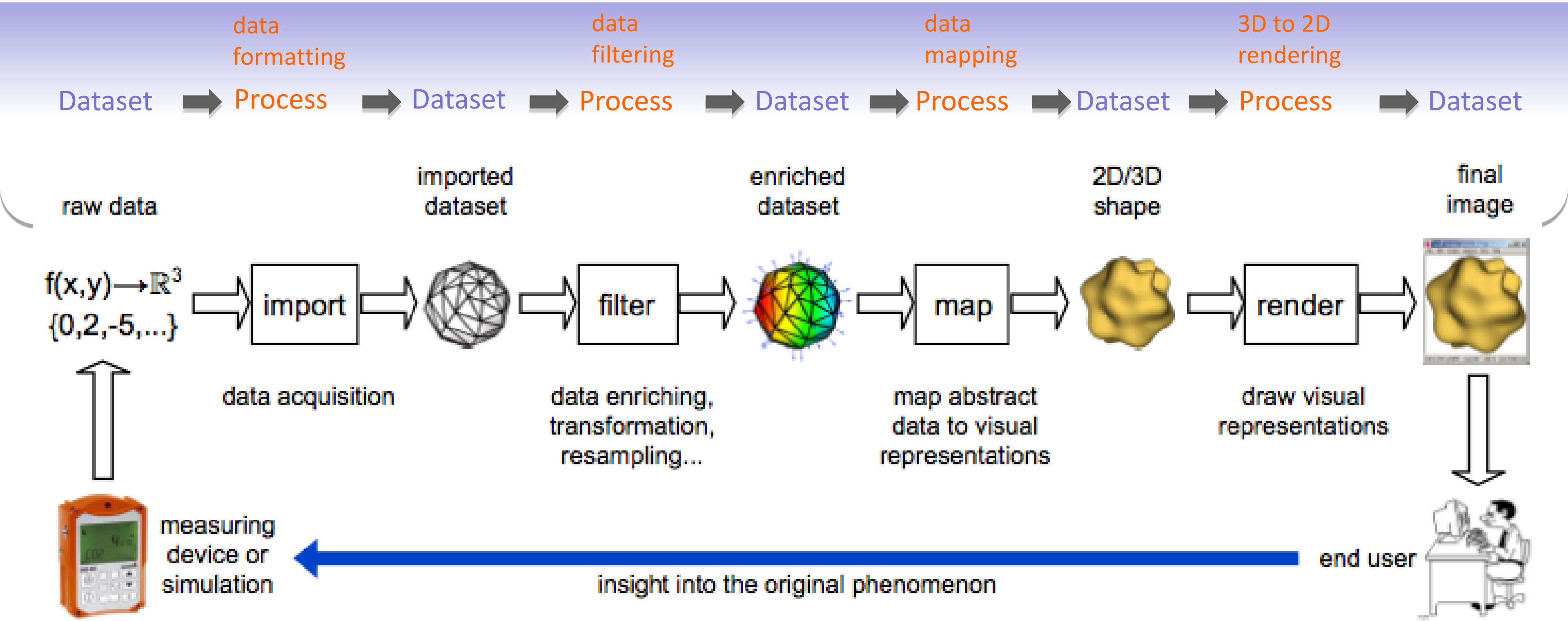
Scalar Visualization

The Visualization Pipeline



- **Filtering** : selection of data portions to be visualized -- usually user-centered.

  It involves processing raw data and includes operations such as resampling, compression, and other image processing algorithms such as feature-preserving noise suppression.

- M**apping** : focus data are mapped to geometric primitives (e.g., points, lines) and their attributes (e.g., color, position, size);

  This steps transforms the pre-processed filtered data into geometric primitives along with additional visual attributes, such as color or opacity, determining the visual representation of the data.

- **Rendering:** geometric data are transformed to image data

  This steps utilizes computer graphics techniques to generate the final image using the geometric primitives from the mapping process.

Data types

- **Scalar fields**

  scalar fields represent a quantity associated with a single (scalar) number, such as voltage, temperature, the magnitude of velocity, etc.

- **Vector fields**

  Vector fields are a fundamental quantity that describe the underlying continuous flow structures of physical processes.

  Ex: Electric fields, magnetic fields, as well the velocities and pressures of fluids.

Scalar Function

f: R → R

1-D, histogram

f : R^2 → R

2-D, color mapping, contouring, height plot

f : R^3 → R

3-D, isosurface, slicing, volume visualization

Popular scalar visualization techniques

- Color mapping
- Contouring
- Height plots

2D Scalar Field

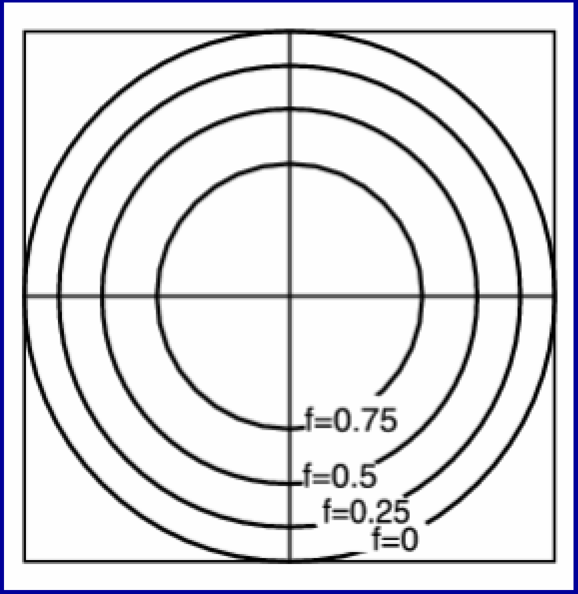
- z = f(x,y)

$$f(x,y) = \begin{cases} 1 - x^2 - y^2, & \text{if } x^2 + y^2 < 1 \\ 0 \end{cases}$$

How do you visualize this function?

- z = f(x,y)

$$f(x,y) = \begin{cases} 1 - x^2 - y^2, & \text{if } x^2 + y^2 < 1 \\ 0 \end{cases}$$

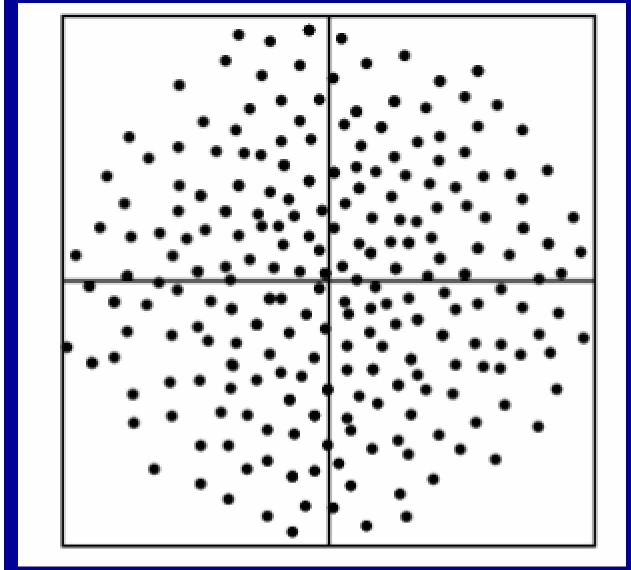


Contours

Topographical maps to indicate elevation

- z = f(x,y)

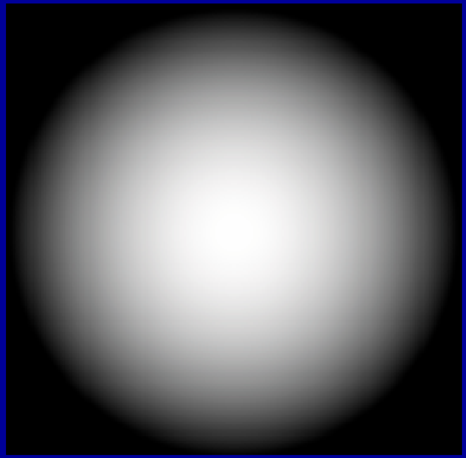
$$f(x,y) = \begin{cases} 1 - x^2 - y^2, & \text{if } x^2 + y^2 < 1 \\ 0 \end{cases}$$



Density plot
Density is proportional to the value of the function

- z = f(x,y)

$$f(x,y)=\begin{cases}1-x^2-y^2, & \text{if } x^2+y^2<1\\ 0\end{cases}$$



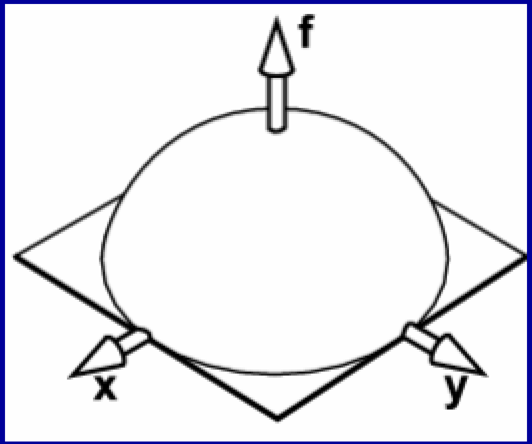
Gray scale density plot

z = 0 => (0,0,0)
z = 0.25 => (0,0,1)
z = 0.5 => (1,0,0)
z = 0.75 => (1,1,0)
z = 1.0 => (1,1,1)

2D Scalar Field

- z = f(x,y)

$$f(x,y) = \begin{cases} 1 - x^2 - y^2, & \text{if } x^2 + y^2 < 1 \\ 0 \end{cases}$$



Height plot
Shows shape of the function

- Scalar data : single value at each location
- Structure of data set may be 1D, 2D or 3D+

– we want to visualise the scalar within this structure

Two fundamental algorithms

– colour mapping (transformation : value → colour)

– contouring (transformation : value transition → contour)

Colour Mapping

- Map scalar value to colour range for display

  Color mapping maps scalar data to colors. The scalar mapping is implemented by indexing into a color lookup table. Scalar values then serve as indices into this lookup table

Color look-up table

– e.g. scalar value = height / max elevation

colour range

Colour Look-up Tables (LUT)

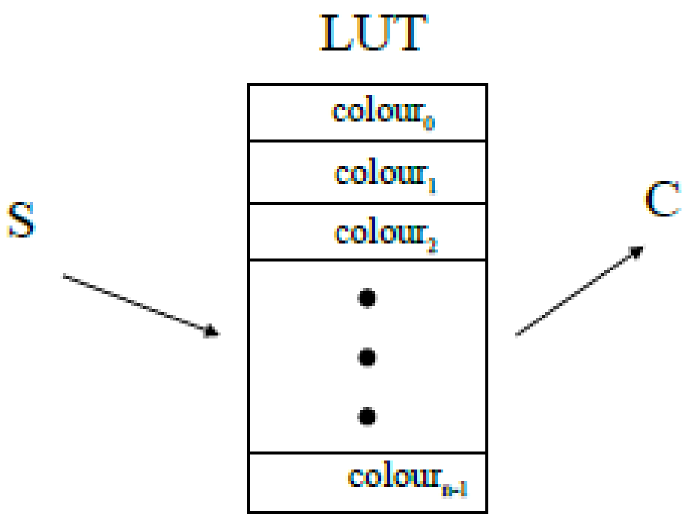
– provide scalar to colour conversion

– scalar values = indices into LUT = blue →red

Color LUT

- **Assume**
  - scalar values S_i in range {min→max}
  - n unique colours, {$colour_0$... $colour_{n-1}$} in LUT

- **Define mapped colour C:**
  - if S< min then C = $colour_{min}$
  - if S > max then C = $colour_{max}$
  - else
  - For (j = 0; j < n ; j++)
    - if (Cj min < S < Cj max) C = Cj



Color Transformation function

- More general form of colour LUT
  - scalar value S; colour value C
  - **colour transfer function** : f(S) = C
  - Any functional expression can map scalar value into intensity values for colour components

Color mapping

Basic idea

•Map each scalar value $f \in \mathbf{R}$ at a point to a color via a function $c : [0,1] \rightarrow [0,1]^3$

Color tables

•precompute (sample) c and save results into a table $\{c_i\}_{i=1..N}$

•index table by normalized scalar values

input data — scalar value f

determine input range — scalar value range $[f_{\min}, f_{\max}]$

normalize input to [0,N] — $$i = N \frac{f - f_{\min}}{f_{\max} - f_{\min}}$$

Color mapping ← $c_i = c\left(\frac{i}{N}\right)$ ← $c : [0,1] \rightarrow [0,1]^3$

color table

desired color transfer function

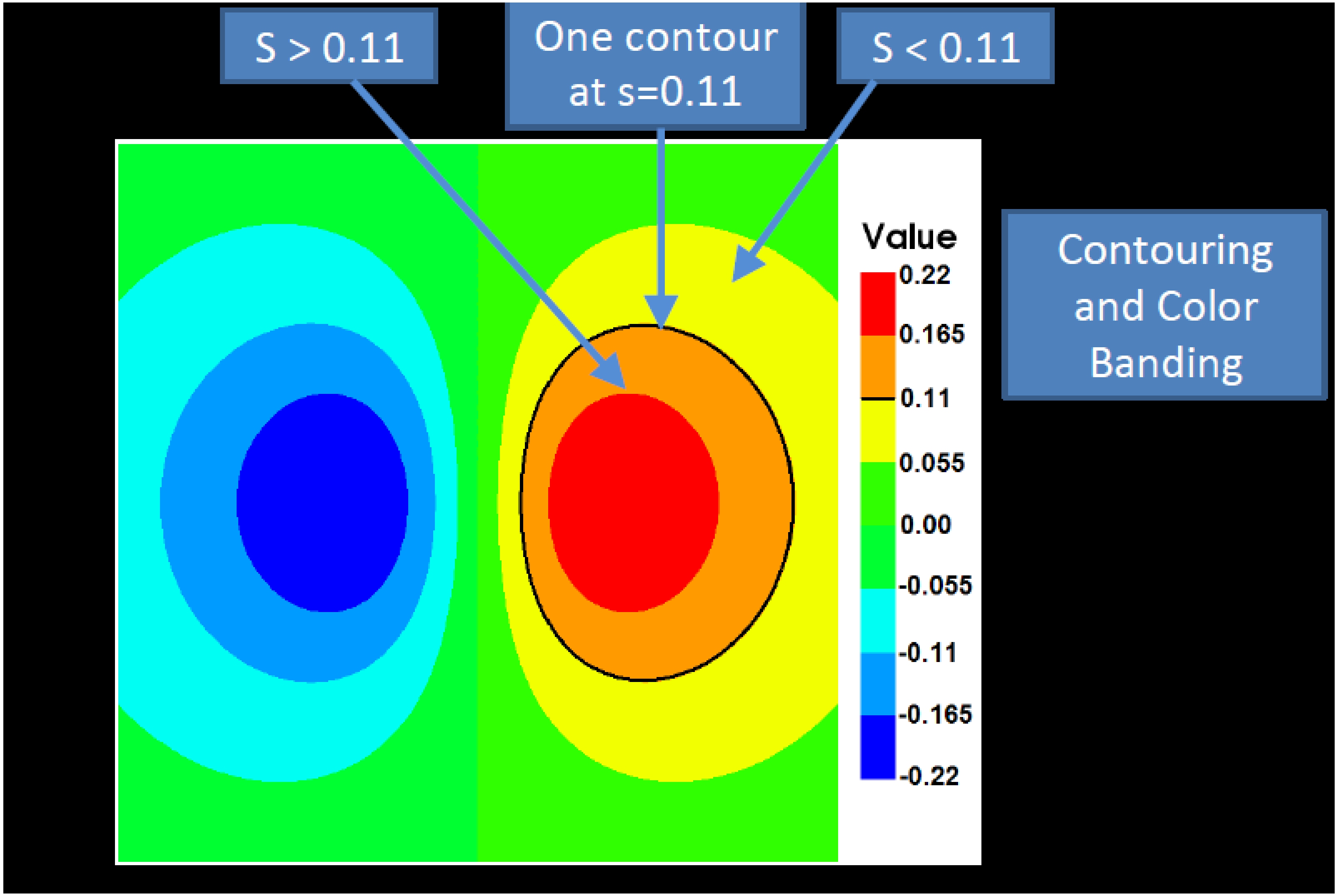
f is color-mapped to c_i

Contouring

- A contour line C is defined as all points p in a dataset D that have the same scalar value, or isovalue s(p)=x

$$C(x) = \{p \in D \mid s(p) = x\}$$

- For 2D dataset, a contour line is called an isoline
- For 3-D dataset, a contour is a 2-D surface, called isosurface


S > 0.11
One contour at s=0.11
S < 0.11
Value
0.22
0.165
0.11
0.055
0.00
-0.055
-0.11
-0.165
-0.22
Contouring and Color Banding

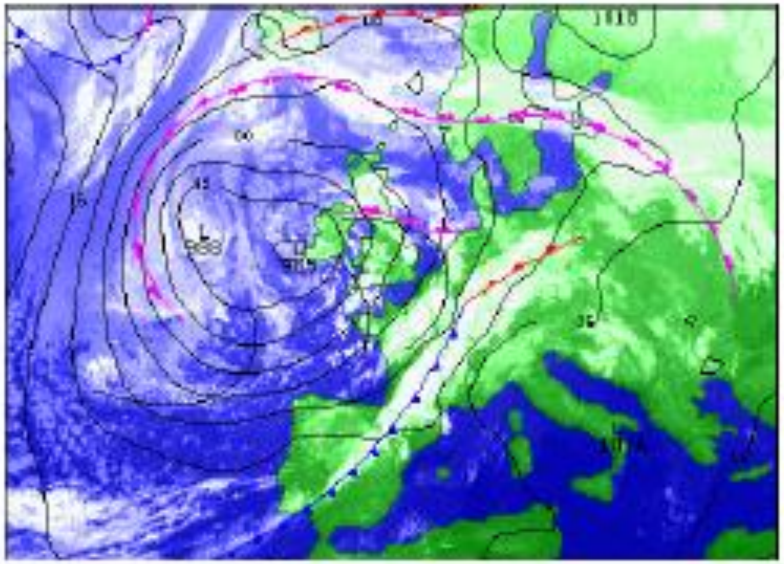
Contouring

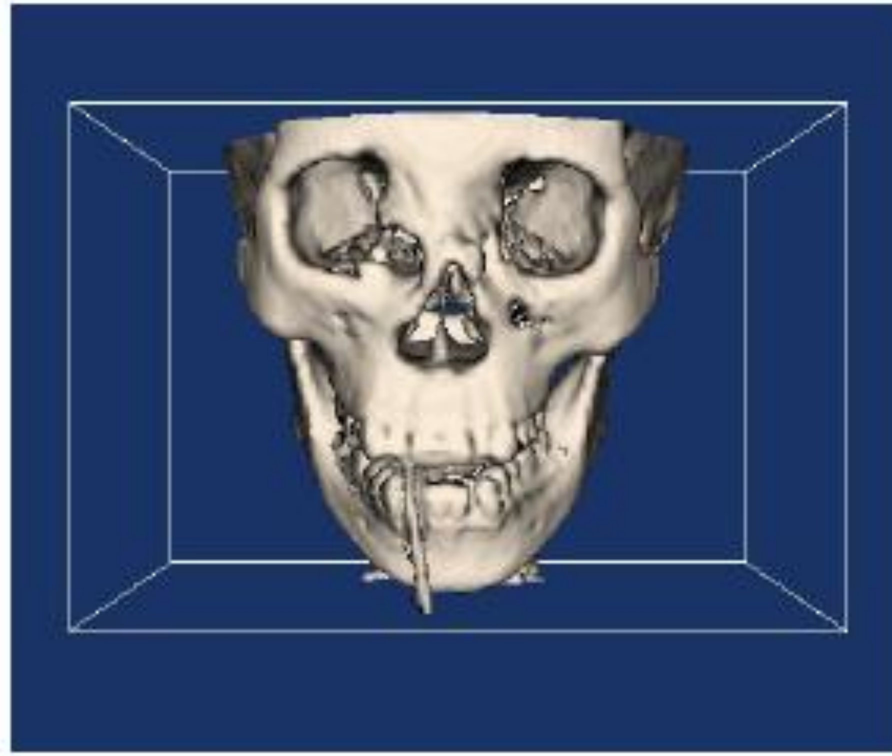
Contours explicitly construct the boundary between regions with values

Boundaries correspond to:

– lines in 2D

– surfaces in 3D (known as isosurfaces)





- lines of constant pressure on a weather map (isobars)
- surfaces of constant density in medical scan (isosurface)
  - “iso” roughly means equal / similar / same as

Vector Visualization

- A vector is an object with direction and length

  $\boldsymbol{v} = (\boldsymbol{v_x}, \boldsymbol{v_y}, \boldsymbol{v_z})$

Examples include

- Fluid flow, velocity $\boldsymbol{v}$
- Electromagnetic field: $\boldsymbol{E}, \boldsymbol{B}$
- Gradient of any scalar field: $\boldsymbol{A} = \boldsymbol{\Delta T}$

Vector Function

$$f\colon R^3 \to R^3$$

(usually in 3-D)

$$f\colon R^2 \to R^2$$

(simpler case: 2-D)

Vector versus Scalar

$$\text{Vector}: \vec{V}$$

$$\vec{V} = V_x \hat{i} + V_y \hat{j} + V_z \hat{k}$$

$$or$$

$$\vec{V} = (V_x, V_y, V_z)$$

$$or$$

$$\vec{V} = \begin{pmatrix} \mathrm{V_x} \\ \mathrm{V_y} \\ \mathrm{V_z} \end{pmatrix} = \begin{pmatrix} f_x(x, y, z) \\ f_y(x, y, z) \\ f_z(x, y, z) \end{pmatrix}$$

$$\text{Scalar}: \mathrm{s}$$

$$s$$

$$s = f(x, y, z)$$

Vector Color Coding

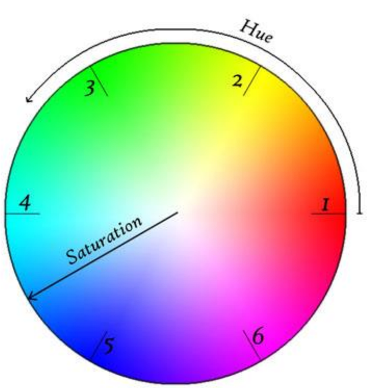
•Similar to scalar color mapping, vector color coding is to associate a color with every point in the data domain

•Typically, use HSV system (color wheel)

•Hue is used to encode the direction of the vector, e.g., angle arrangement in the color wheel

•Value of the color vector is used to encode the magnitude of the vector

•Saturation is set to one



In each cylinder, the angle around the central vertical axis corresponds to "hue",
the distance from the axis corresponds to "saturation",

and the distance along the axis corresponds to "lightness", "value" or "brightness".