# 4. 1D and 2D Scalar Fields

# Introduction

## Basic mapping techniques

- Mapping
  - from (filtered) data to renderable representation (visualization pipeline)
- Most important part of visualization
- Possible visual representations
  - Position
  - Size
  - Orientation
  - Shape
  - Brightness
  - Color (hue, saturation)
  - …

ohm

# Introduction

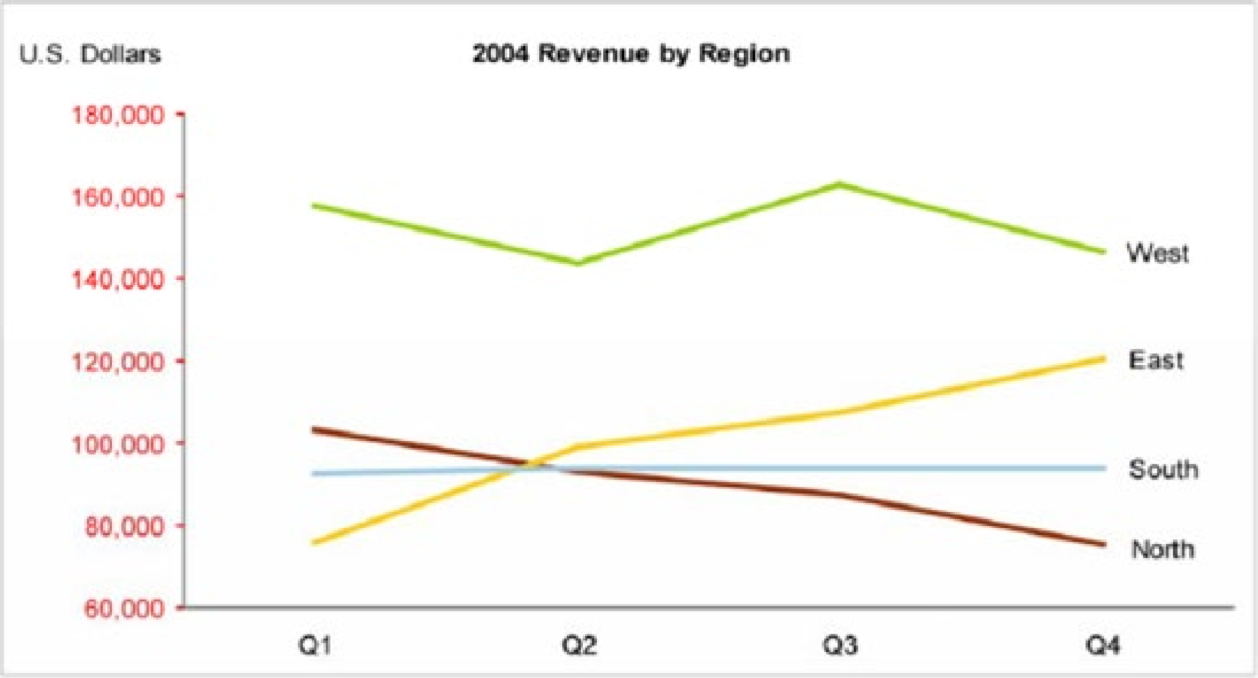
## 1D Functions - Typical strategy when creating a graph

- Stepping through a series of choices, including
  - Different types of graphs
  - Several aspects of appearance
- Choices are often made as if one is sleepwalking
  - Neglected issue: Why one choice is better than another?
- What we have to know
  - Nature of the data
  - Graphing conventions
  - Psychological investigations
    - A bit about visual perception
    - Evaluate appropriateness of mapping approaches

# Data and Scales

## Quantitative information consists of both

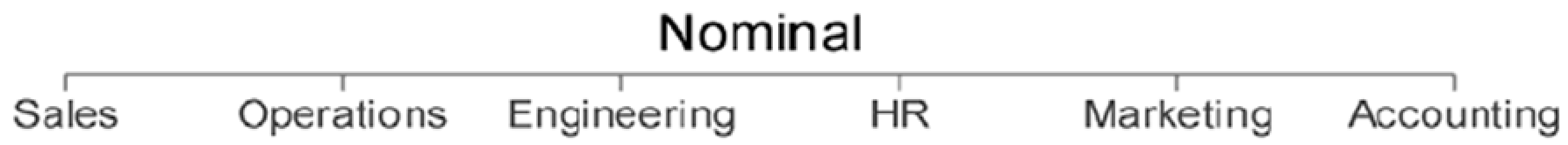
- Quantitative data: Numbers
- Categorical data: Labels what numbers measure
- Example: Categorical (horiz. axis) and quantitative data (vert. axis)



# Data and Scales

## Three types of categorical scales

- **Nominal scales**
  - Consists of discrete items
    - Belong to a common category
    - Do not relate to one another in any particular way
    - Differ in name only (nominally!)
    - Have no particular order and don't represent quantitative values




- Further examples
  - Continents: the Americas, Asia, Europe, ...
  - Departments (e.g. sales, marketing and finance)

# Data and Scales

## Three types of categorical scales

- **Ordinal scales**
  - Consists of items with intrinsic *order*
  - Items do not represent quantitative values


- Further examples involve rankings
  - "A, B and C"
  - "small, medium and large"
  - "poor, below average, average, above average and excellent"

ohm

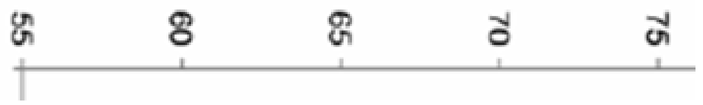
# Data and Scales

## Three types of categorical scales

- **Interval scales**
  - Consists of items with *intrinsic order*
  - Represent quantitative values as well


- Example
  - Start as quantitative then convert into categorical scale

Consider quantitative range along an axis




Conversion into a categorical scale according to

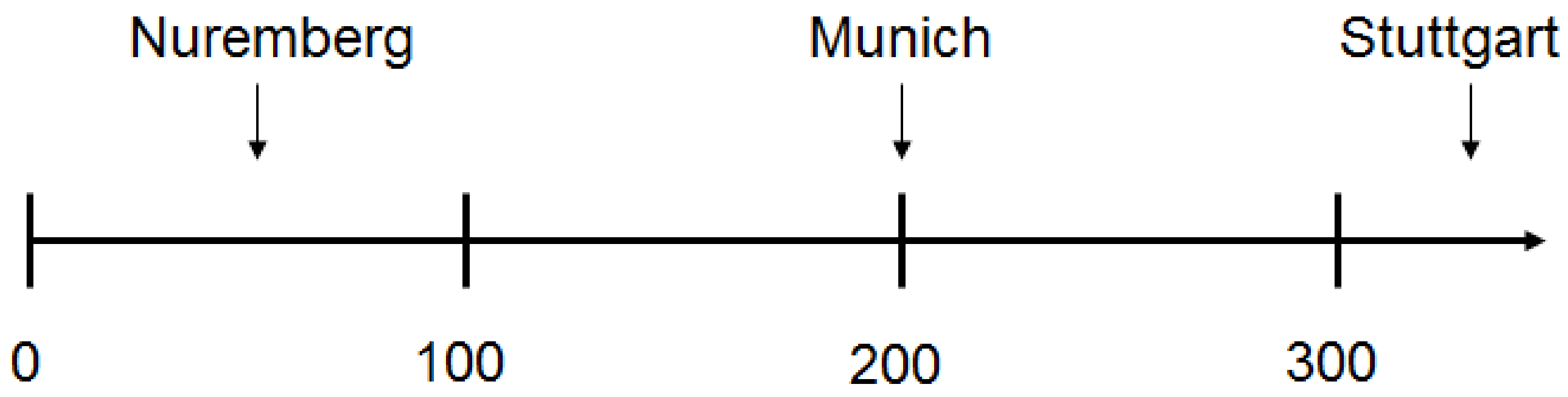
1. > 55 and ≤ 60
2. > 60 and ≤ 65
3. > 65 and ≤ 70
4. > 70 and ≤ 75
5. > 75 and ≤ 80

# 4.1 Diagram Techniques

# Diagram Techniques

- Discrete range of definition
  - 1D drawing / 2D drawing

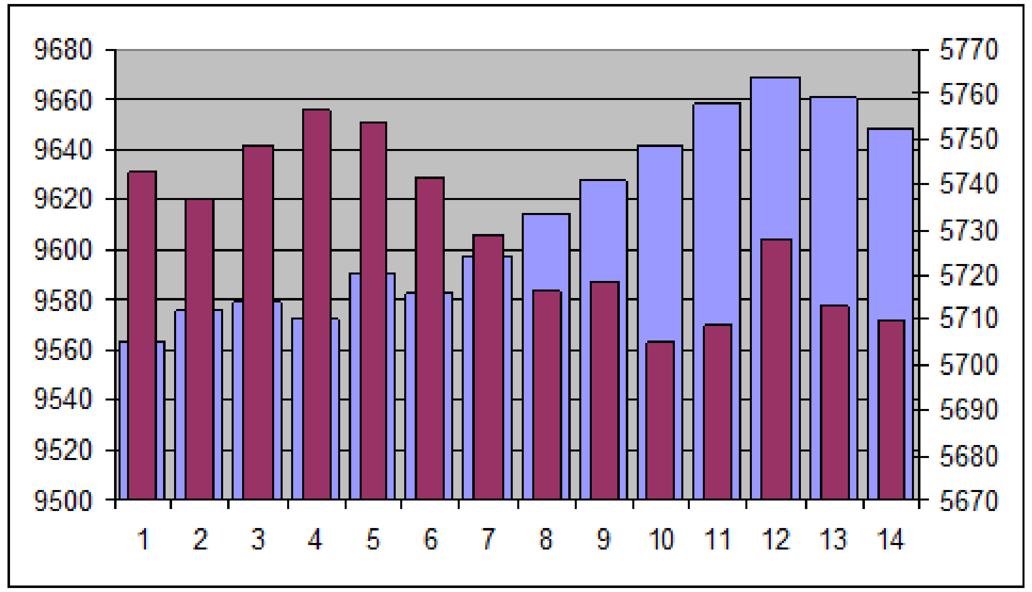
- Example
  - Distance from reference location (e.g. Erlangen)

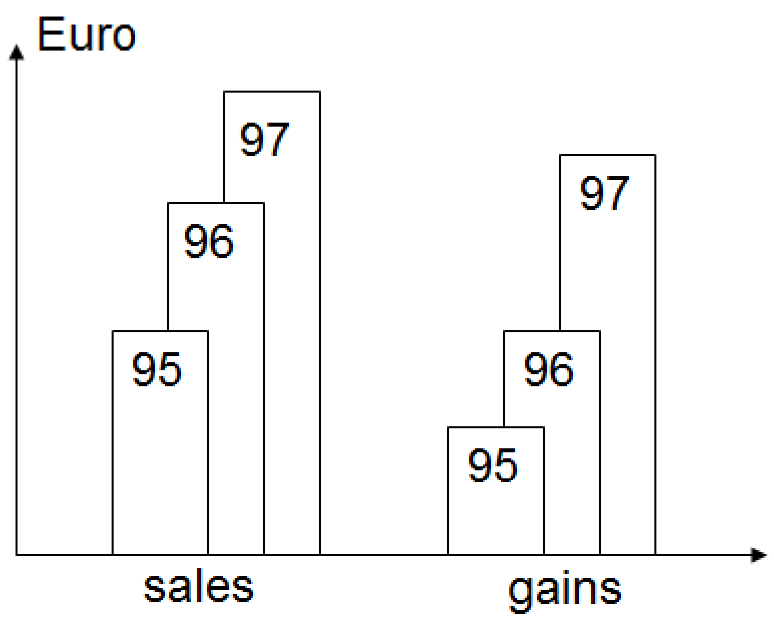


# Diagram Techniques

## Bar charts

- Appropriate for **nominal and ordinal scales**
  - Individual items *not related closely enough* for link with lines



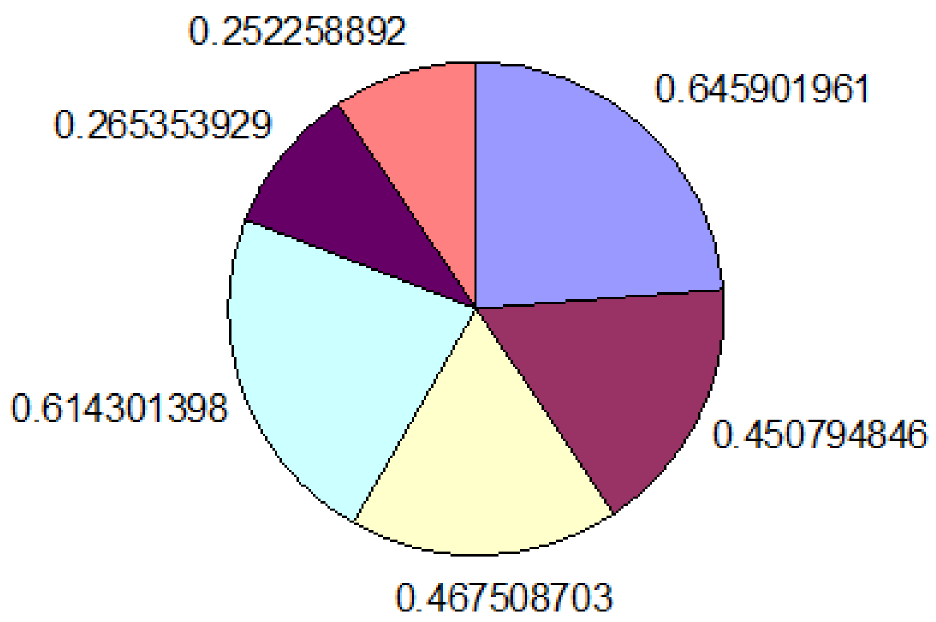



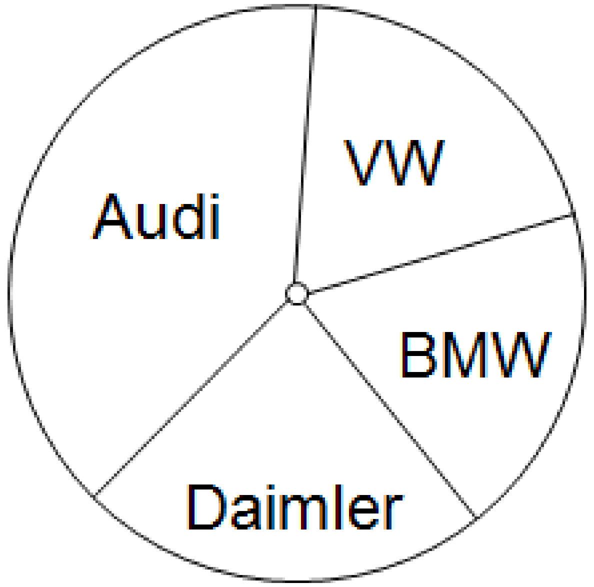

- Example
  - Transition from one sales region to the next
    - Use bars - inappropriate to display with lines

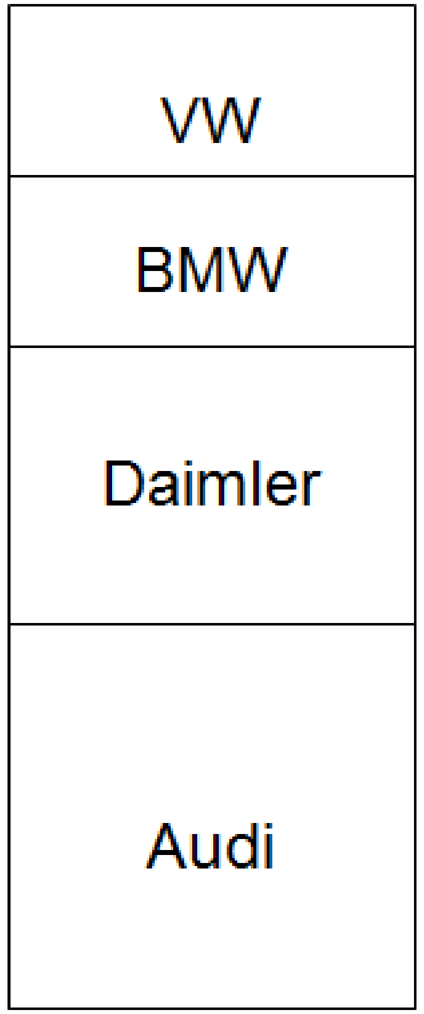
# Diagram Techniques

## Pie charts

- Quantitative data that adds up to a (fixed?) number



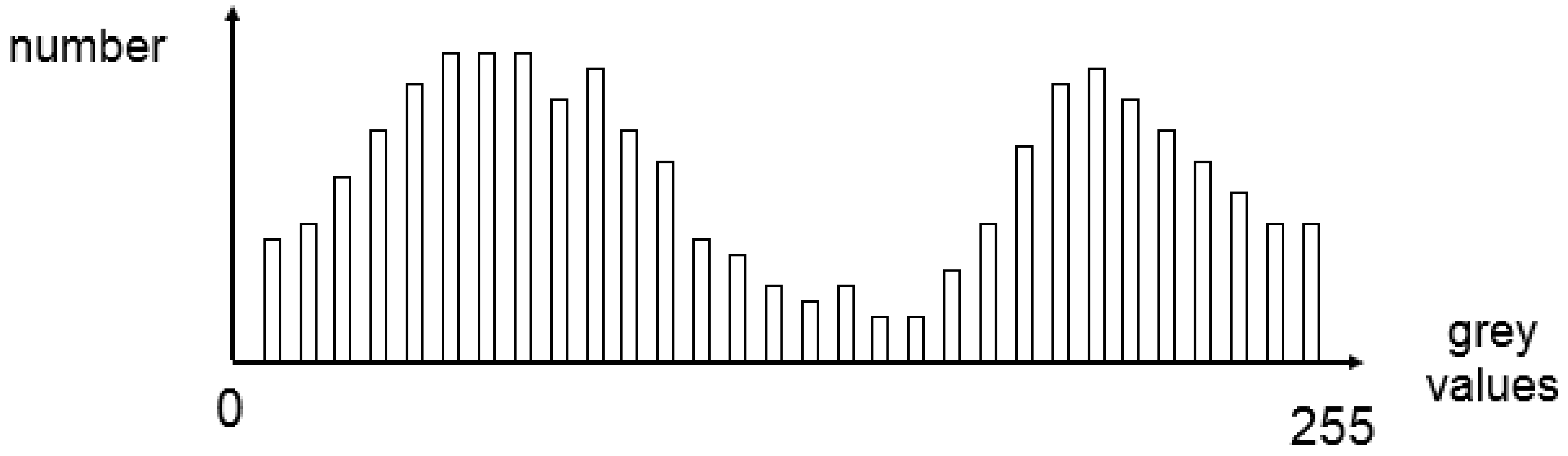



# Diagram Techniques

## Histogram

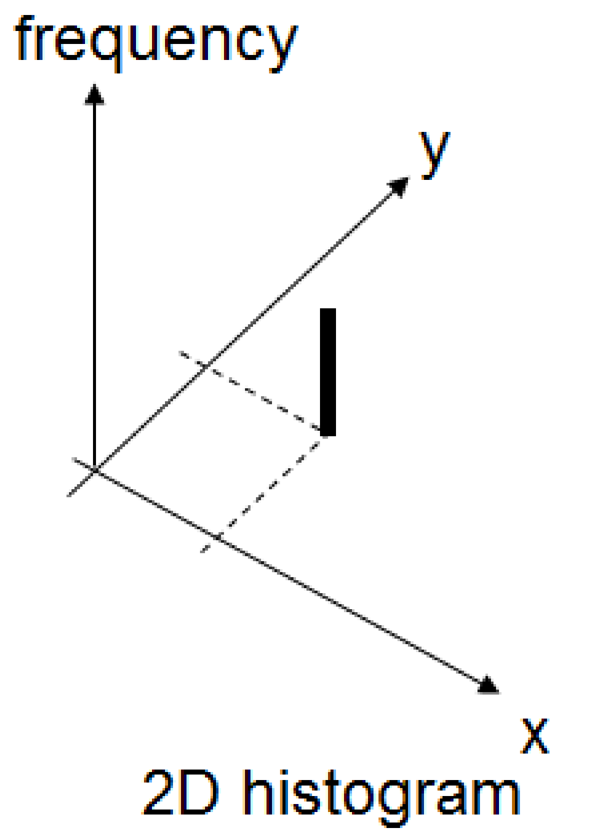
- Measure the frequency (number) of items
  - Very important in scientific analysis
- Example
  - Grey values in an 8-bit image

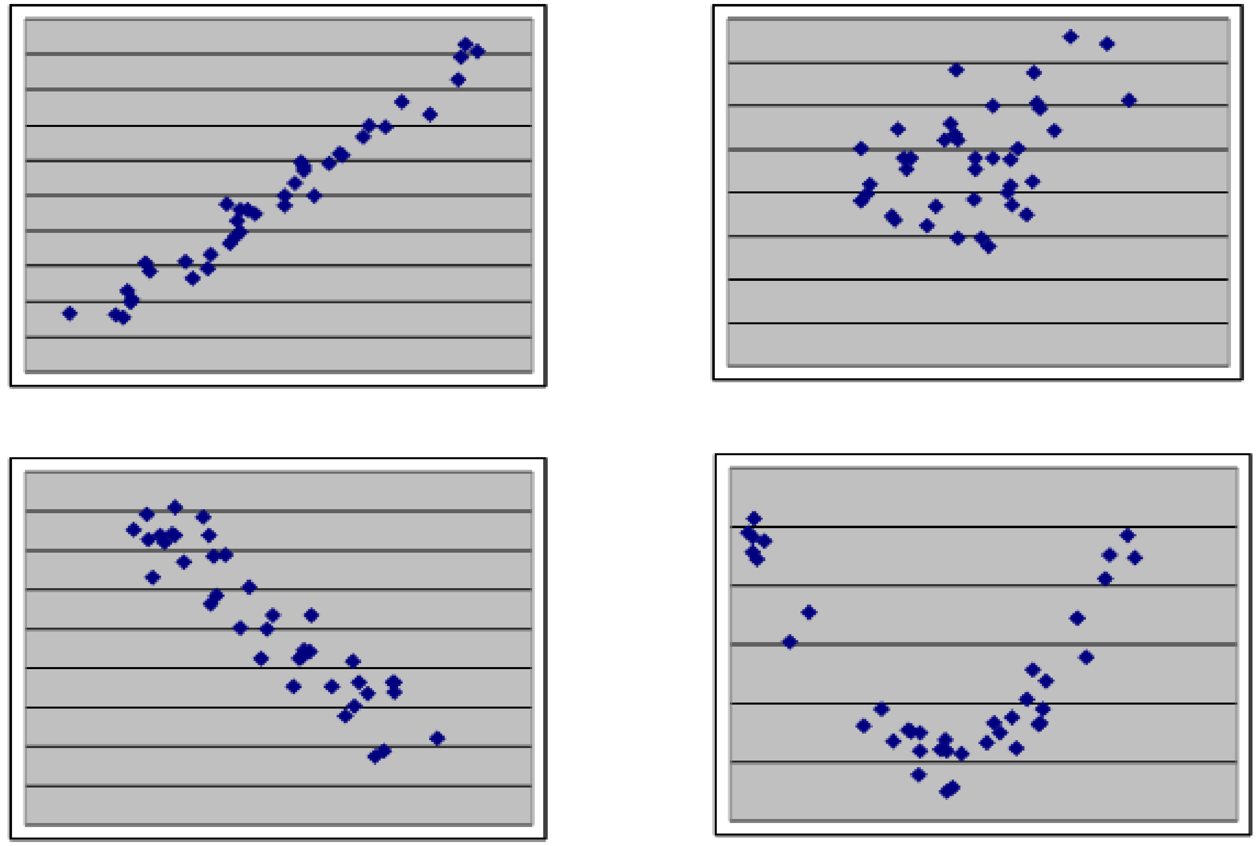


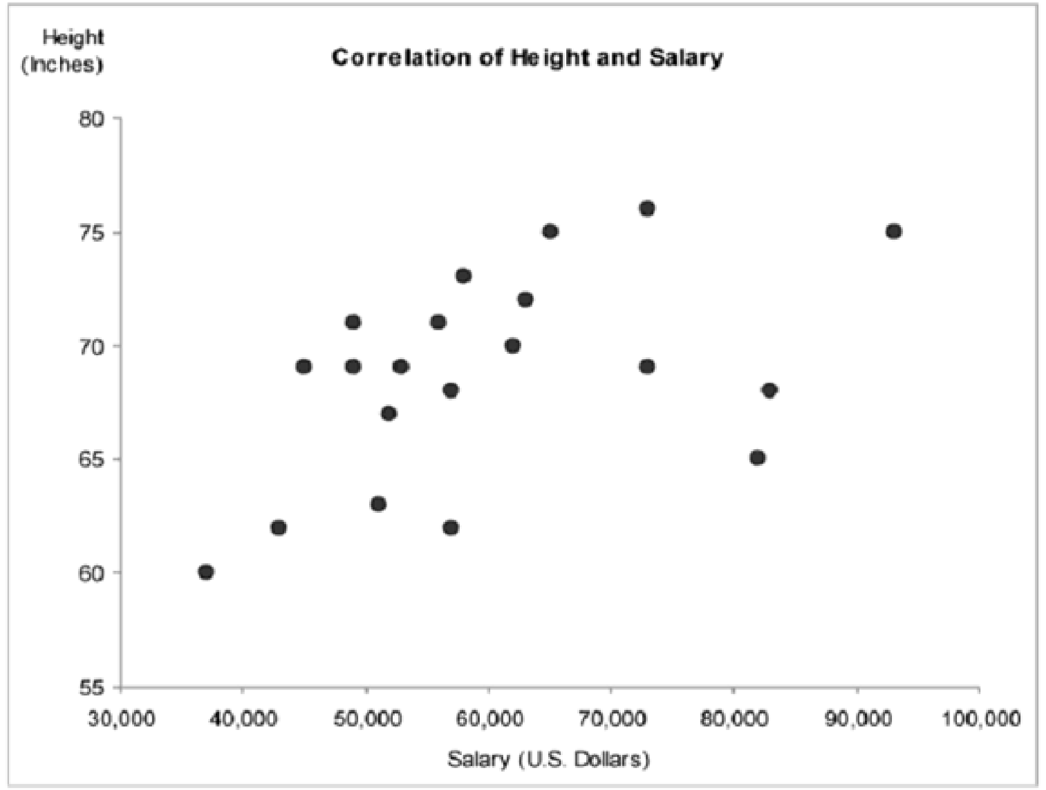
# Diagram Techniques

## Scatter plots

- Quantitative scales along both axes
- Visual recognition of correlations



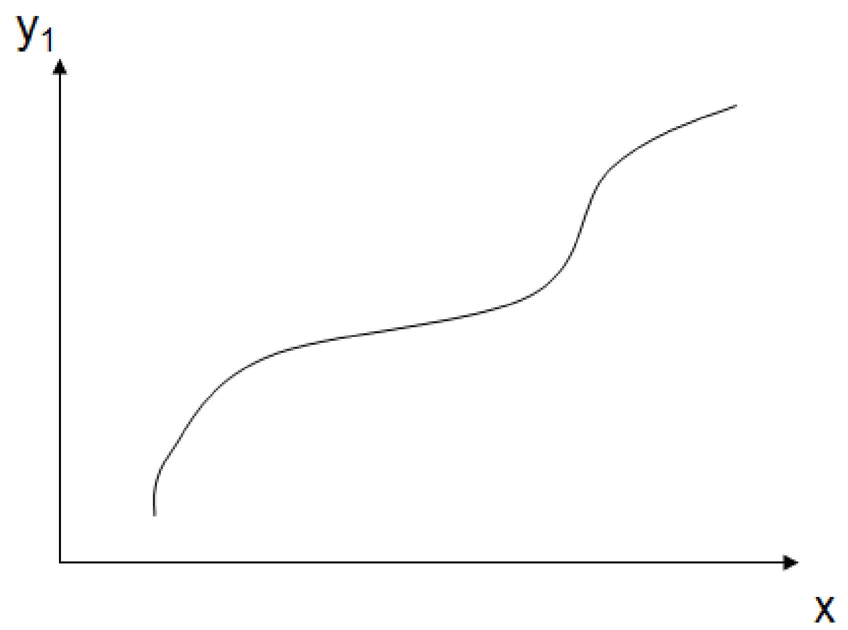



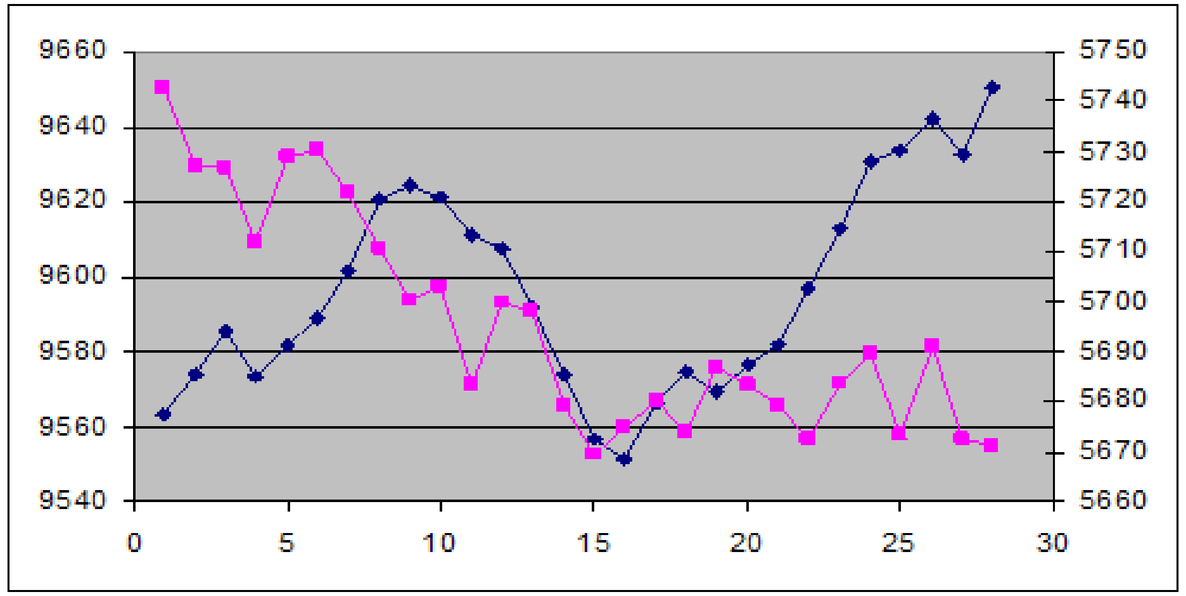


# Diagram Techniques

## Line graphs

- Connection between points (*interpolation*!)
- Appropriate for interval scale
  - Shows trend of patterns in the data
- Example
  - Change from one day to the next or from one price to the next
    - Appropriate to use lines for display

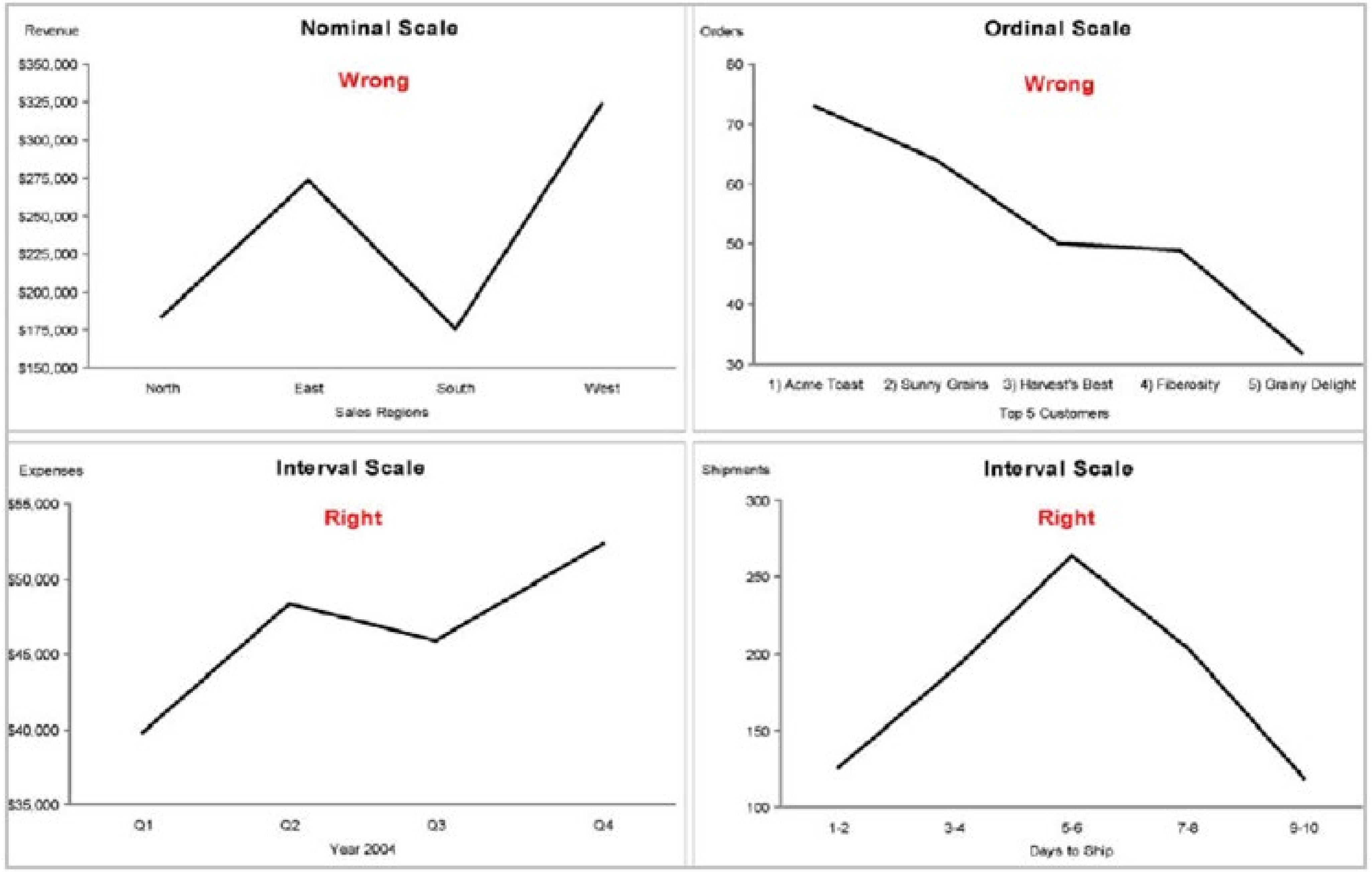


# Diagram Techniques

## Line graphs

- Inappropriate and appropriate use of lines

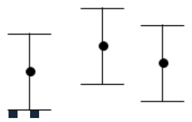


ohm

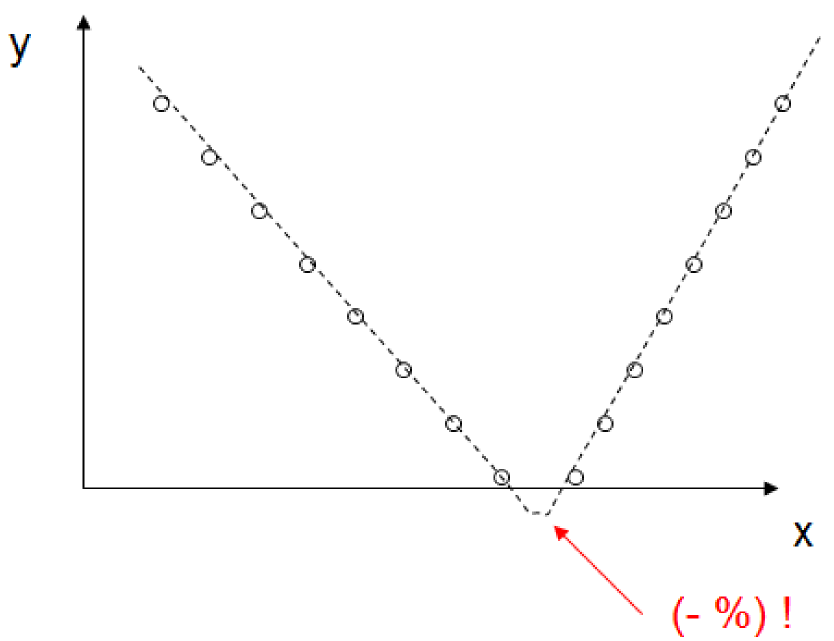
# Diagram Techniques

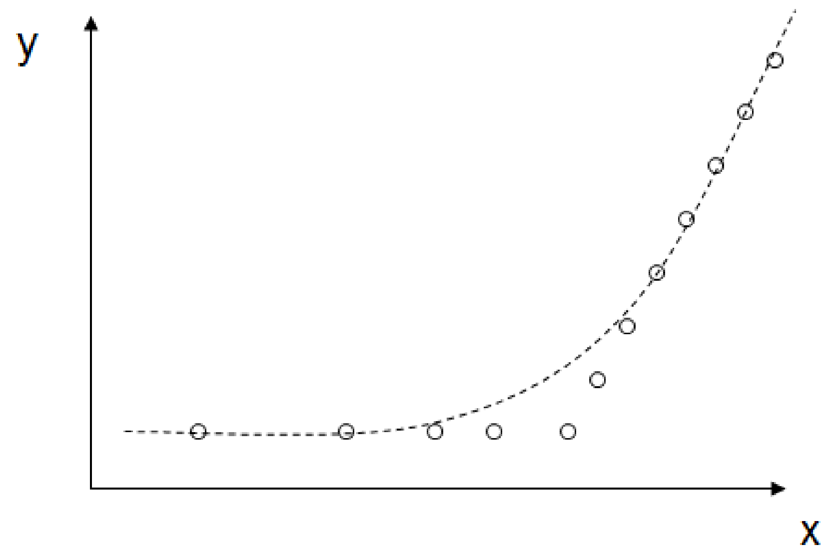
## Line graphs

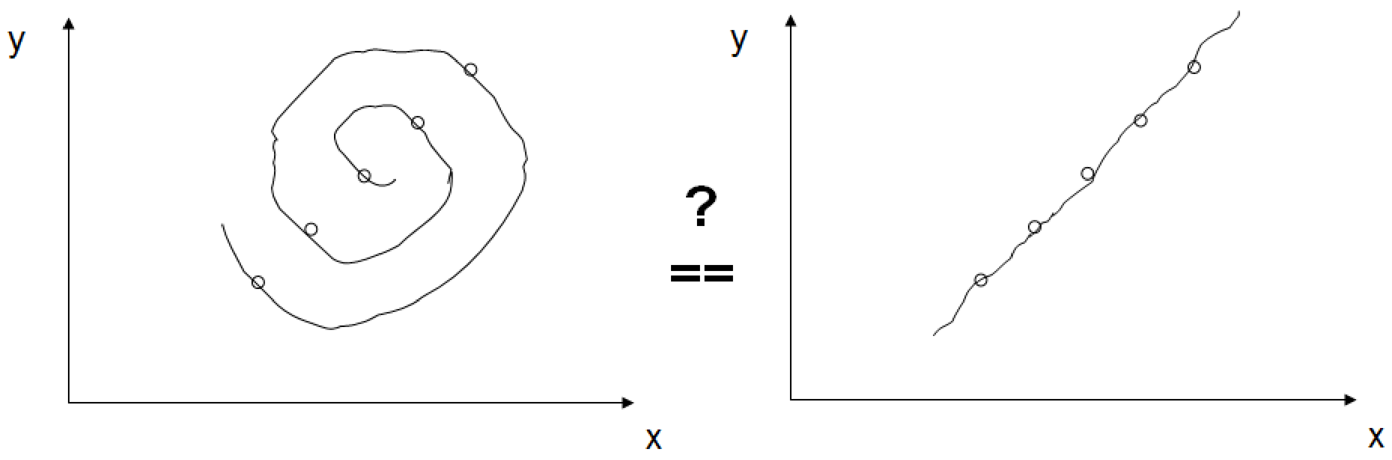
- Avoid "blind" interpolation
- Show all data points
- Set an order
- Caption of axes
    - Several ordinates / y-axes
- Additional annotations
    - e.g. Labels, error bars
- Line type
    - Dashed, dotted, color, ...

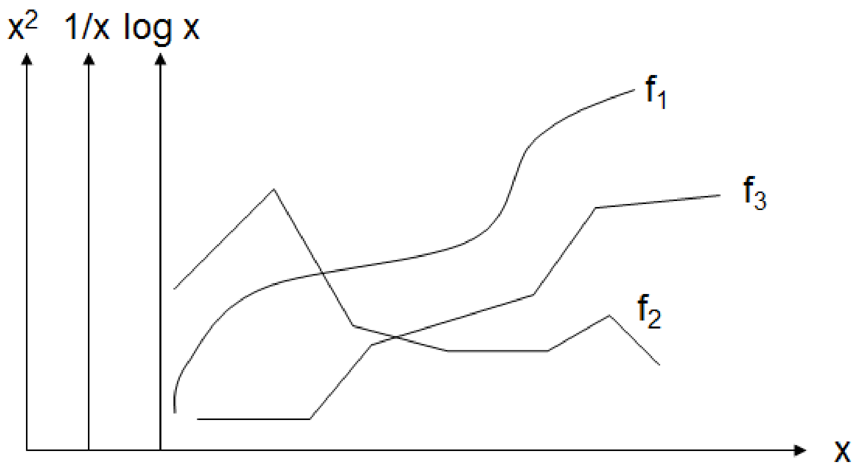


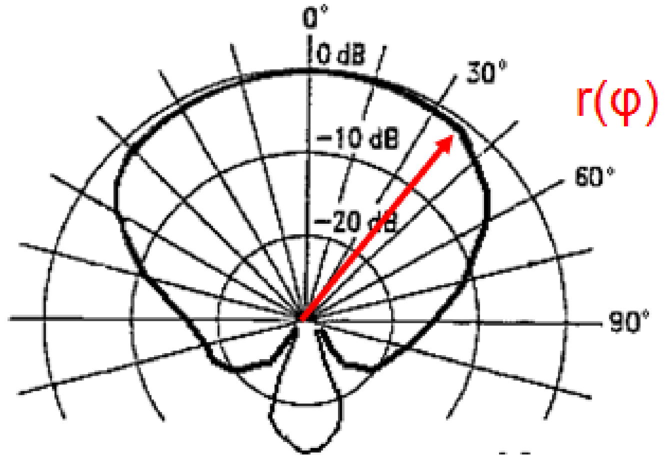
- Continuous range - parametric representation



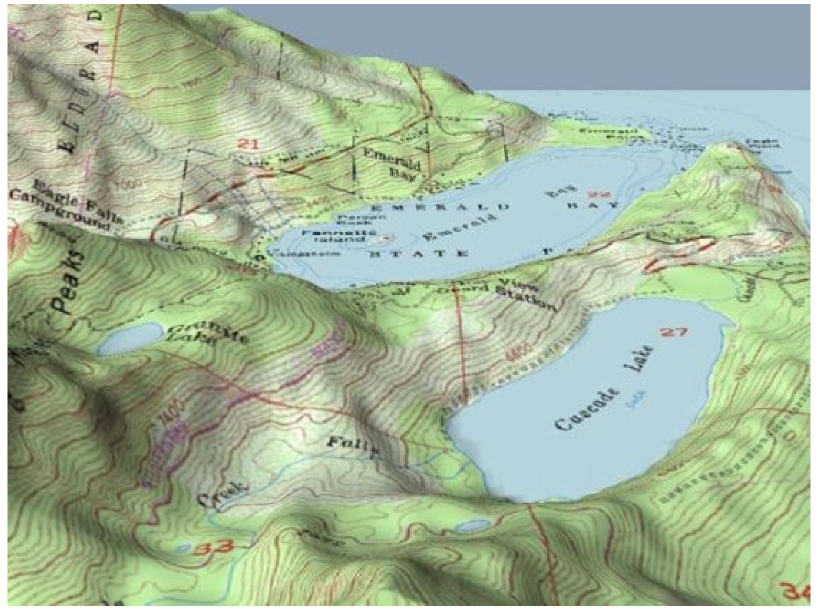



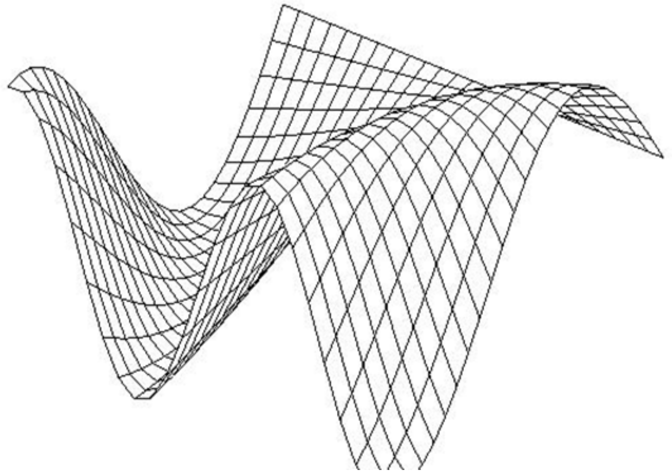



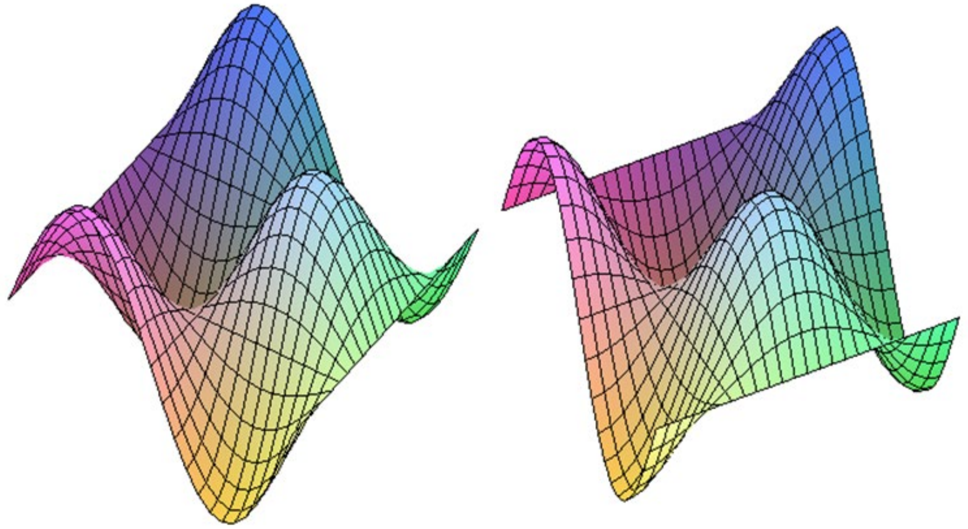
# Height Fields

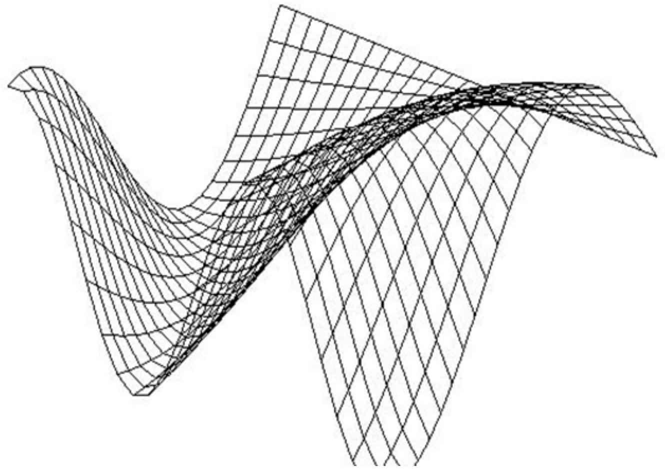
## Function plot for a 2D scalar field

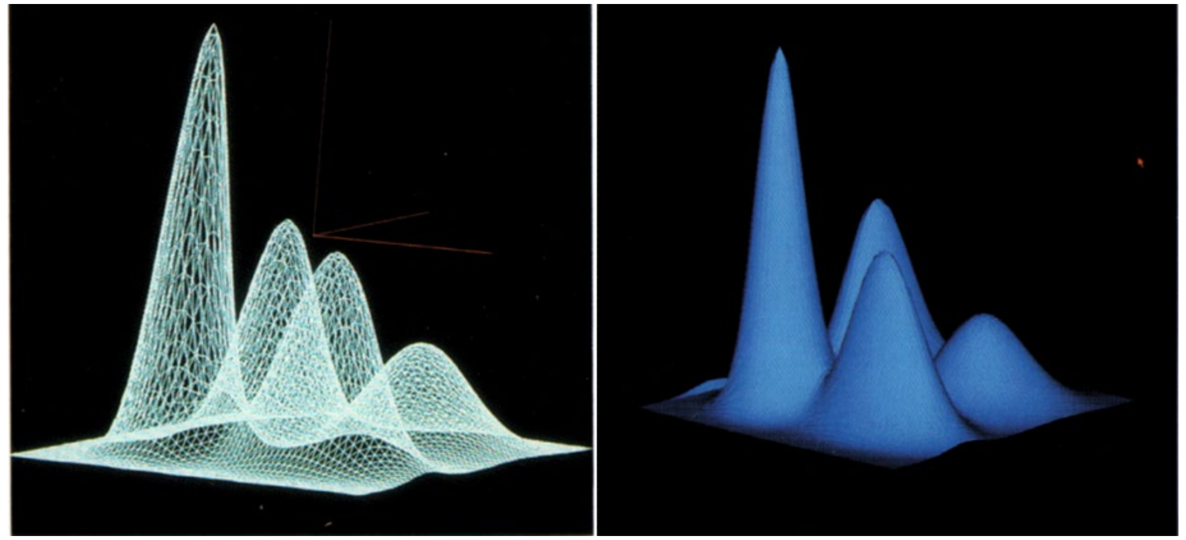
- 2D manifold: surface
- Plot styles
  - Wireframe
  - Hidden lines
  - Shaded surface

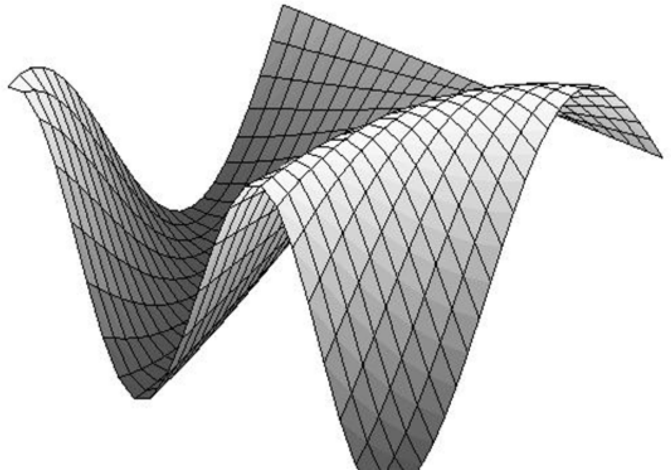












# 4.2 Isolines

# Isolines

## Visualization of 2D scalar fields

- Given a scalar function $f: \Omega \rightarrow R$ and a scalar value $c \in R$
- The isoline / isocontour or contour line consist of points $\{(x, y) \mid f(x, y) = c\}$

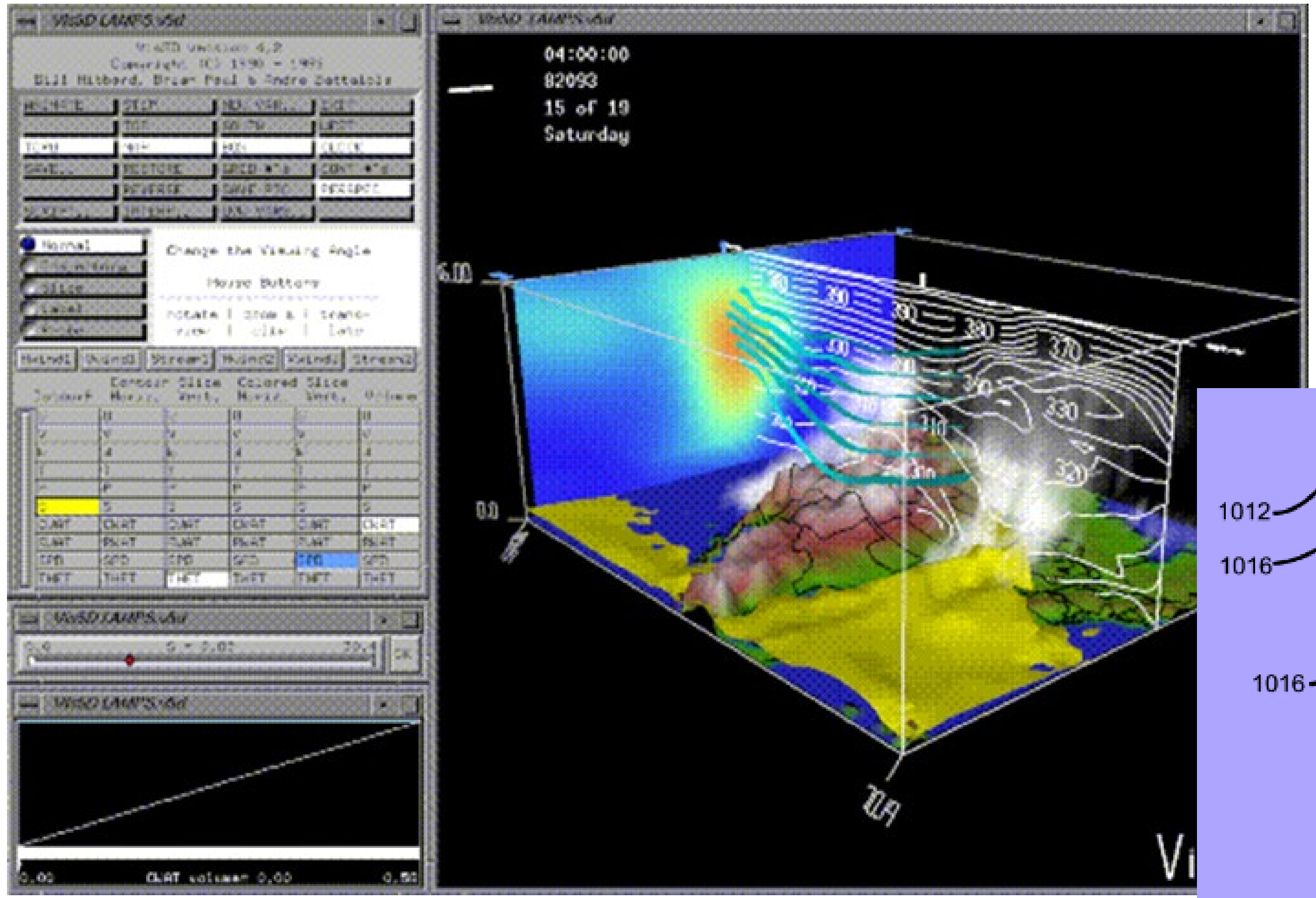
Punkt -> Konstante bei Funktion für Punkt

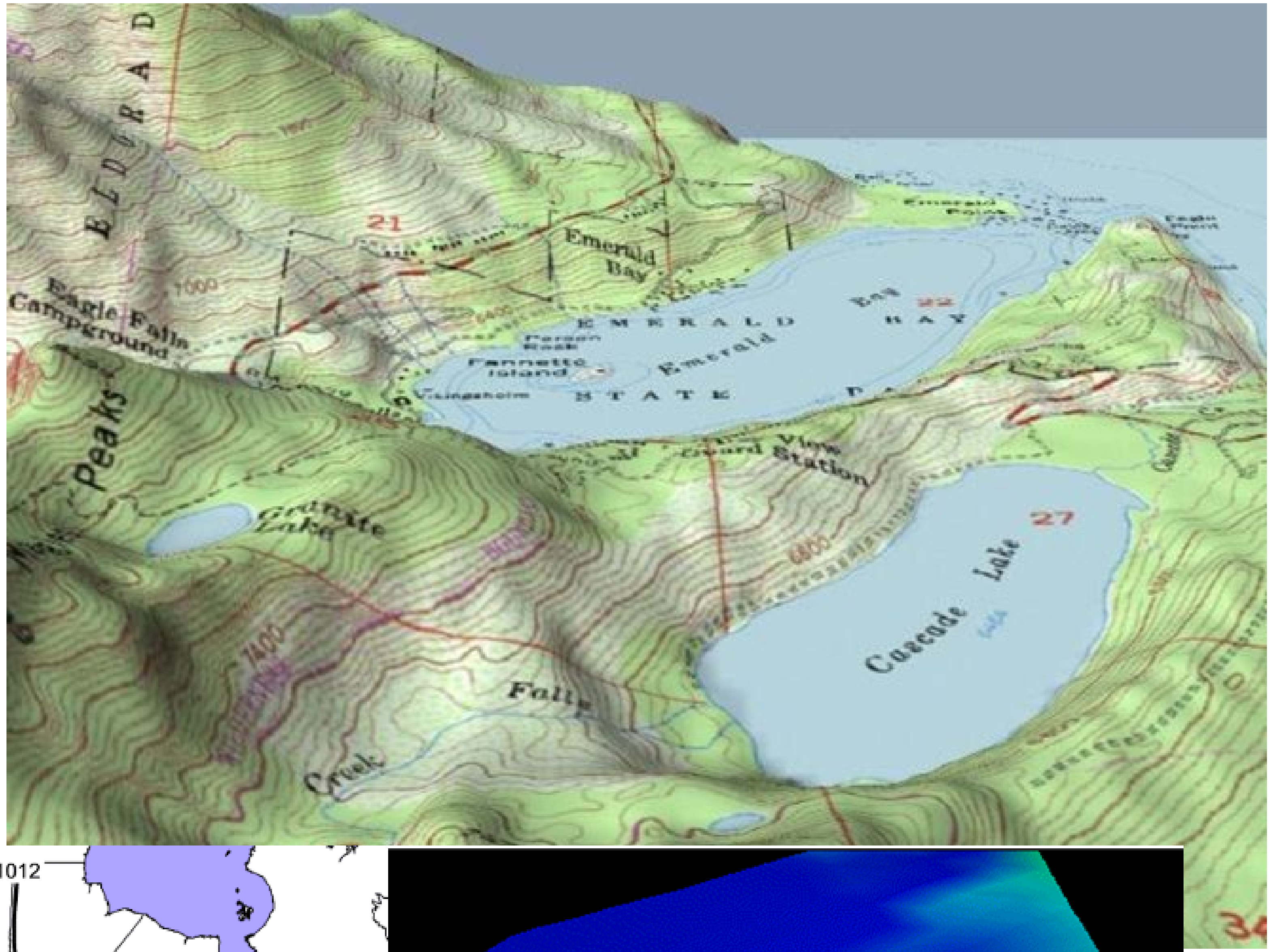
- If f is differentiable and $grad(f) \neq 0$, then isolines are curves
  - if $grad(f) = 0$ in some points, "isolines" may be single points or whole areas of the domain

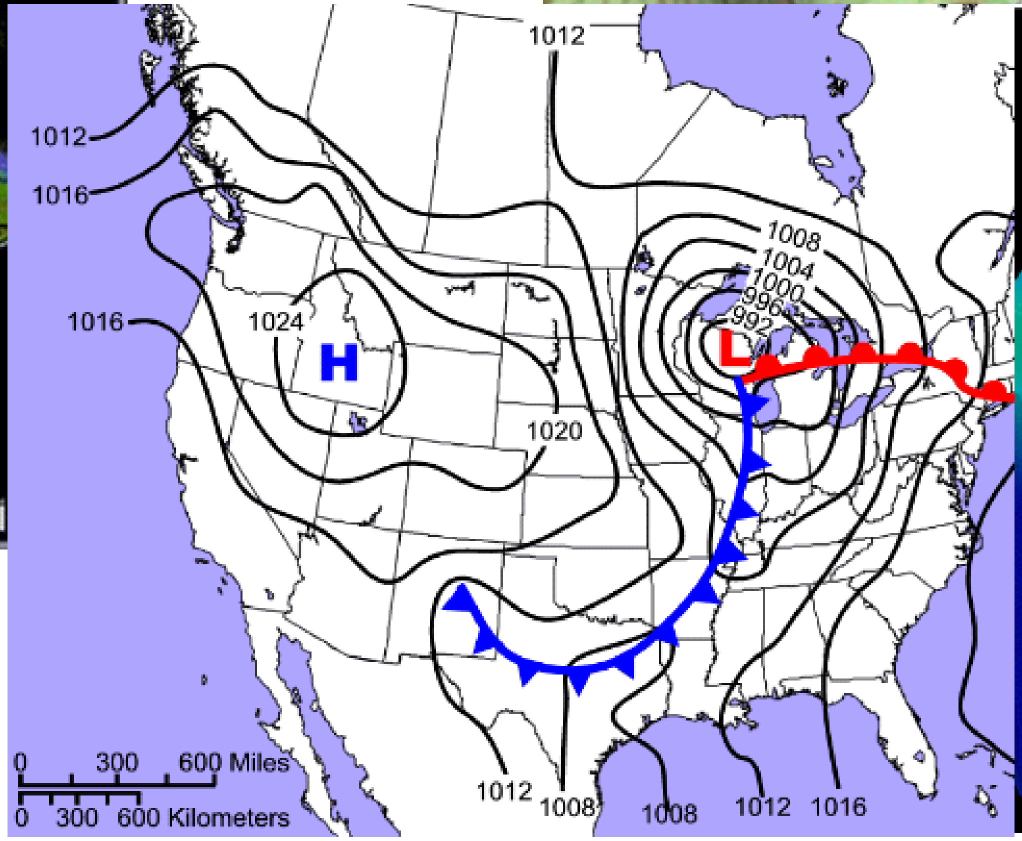
- Remark
  - Isolines are closed curves, or they get in and out of the domain. They **do not terminate** in the domain!

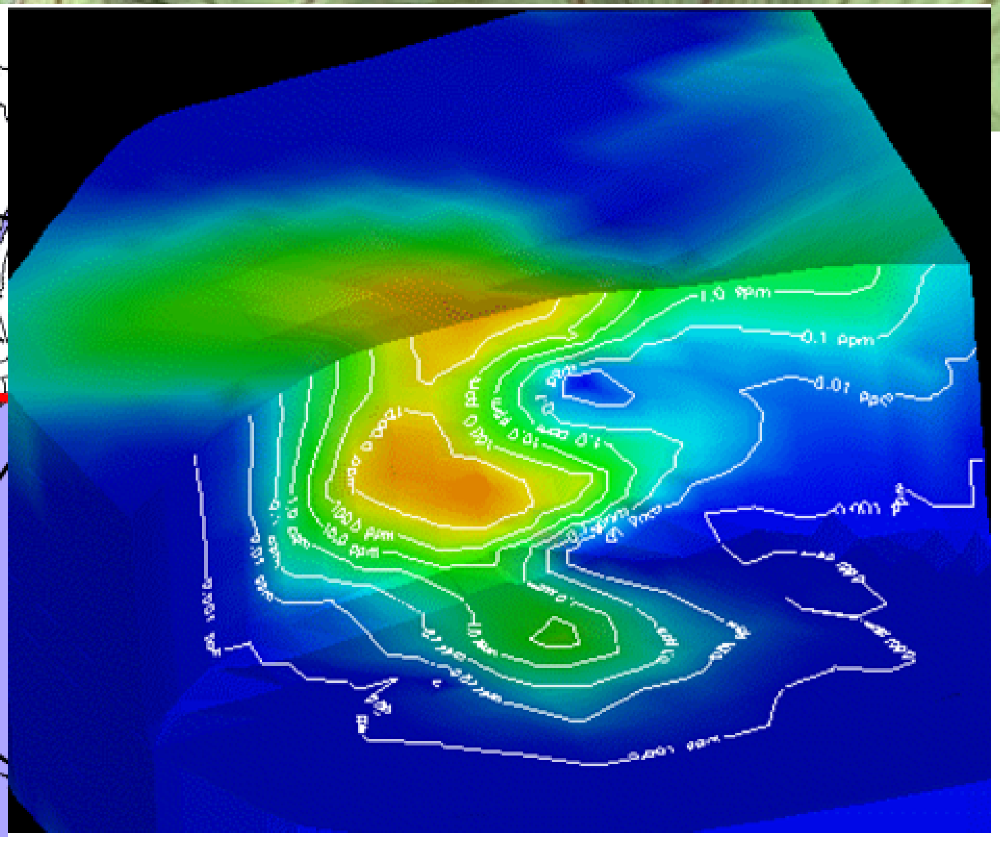
# Isolines

## Examples



# Isolines

## Pixel by pixel contouring

- Straightforward approach
  - Scanning all pixels for equivalence with iso-value (this is an image space method)
- Input
  - $f : (1,\ldots,x_{max}) \times (1,\ldots,y_{max}) \rightarrow R$
  - Iso-values $I_1, \ldots, I_n$ and iso-colors $c_1, \ldots, c_n$
- Algorithm

```
for all (x,y) ∈ (1,...,xmax) x (1,...,ymax) do
    for all k ∈ { 1,...,n } do
        if |f(x,y) - Iₖ| < ε then
            draw(x, y, cₖ)
```

- Problems
  - Isoline missed if gradient of f is too large, produces no additional data, only visualization

# Isolines

## Isolines on triangle meshes

- Cell search *over all cells* (isovalue $f_c$)
    - No intersection

```
if (f_i > f_c or f_i < f_c)
  for i = 1,2,3
```

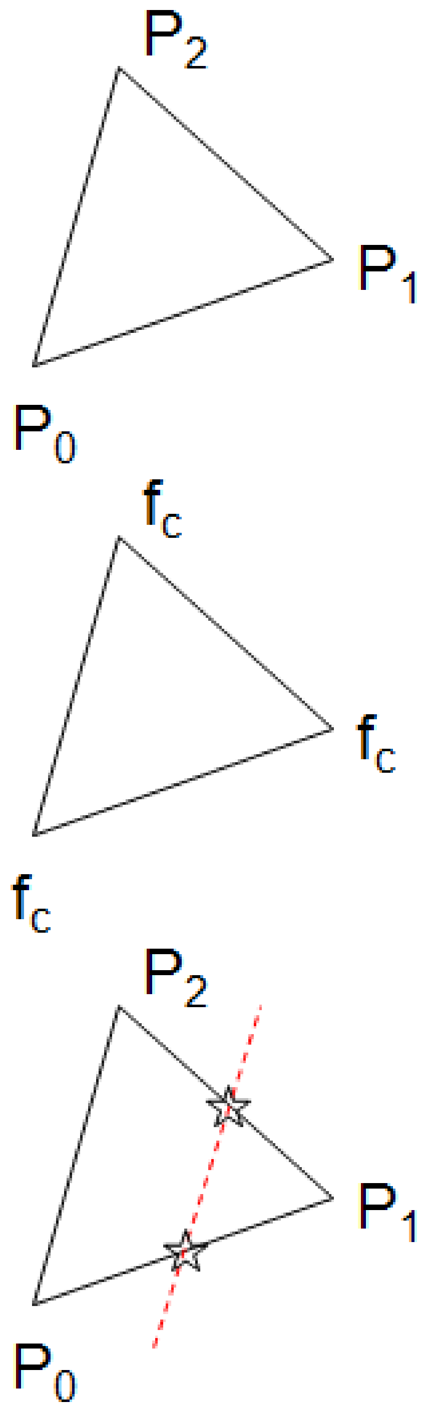
    - Trivial reject

```
if (f_i == f_c)
  for i = 1,2,3
```

    - Normal case: find 2 intersection points

```
if (f_0 < f_c < f_1 or f_1 < f_c < f_0 )
  then calculate intersection point

do the same for P_1P_2 and P_2P_0
```



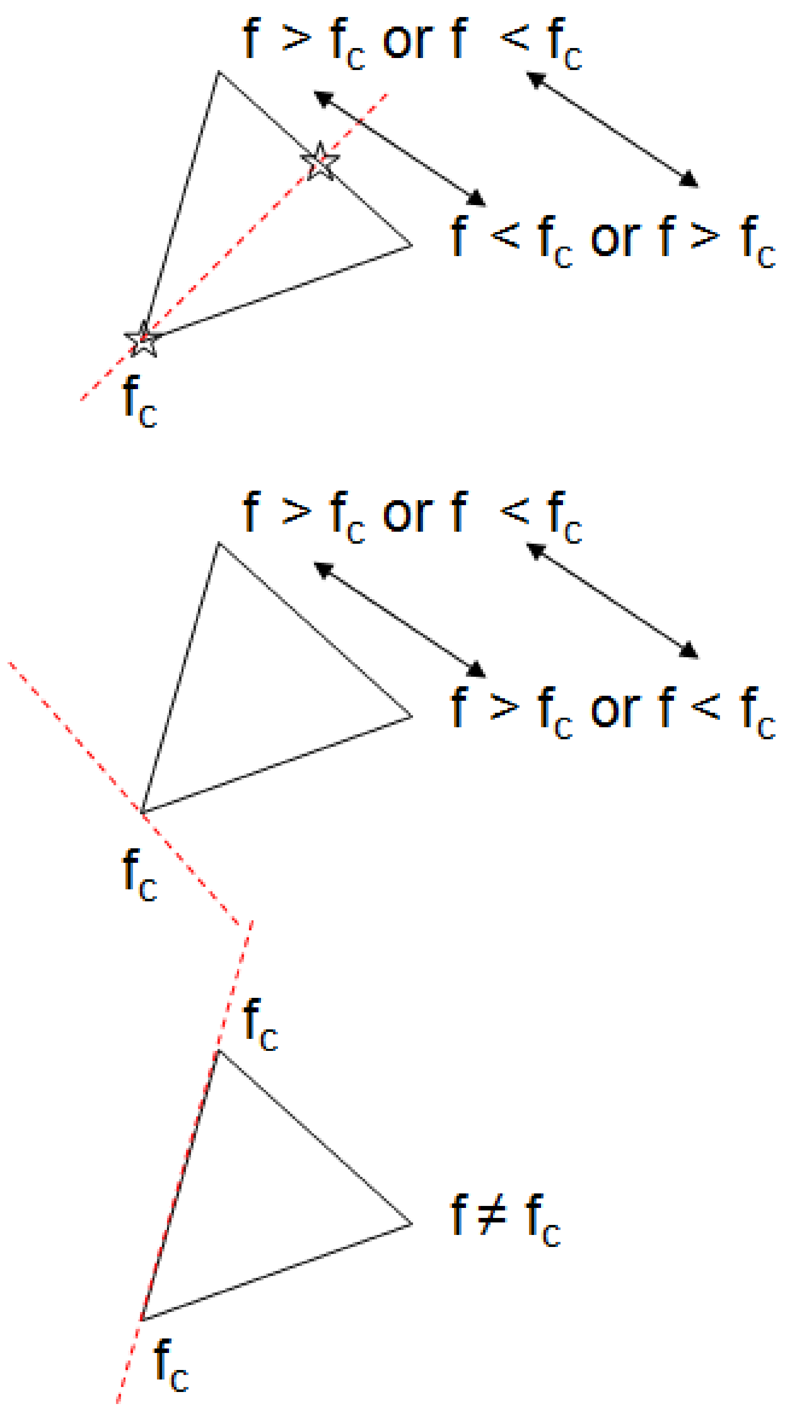

Check Verhältnis aller
Funktionswerte mit dem Isowert

ohm

# Isolines

## Special cases

- Only one "normal" intersection point
- No "normal" intersections
  - Draw no lines
  - Consider in other triangle
  - Draw line along edge
  - Draw twice
    - For this and the neighboring triangle

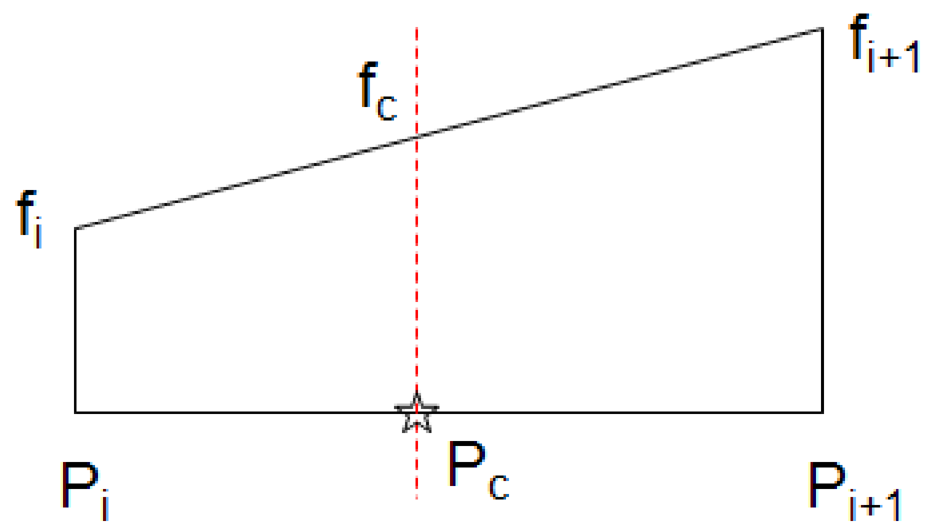


ohm

# Isolines

## Calculate intersection points

- Interpolate point from data value



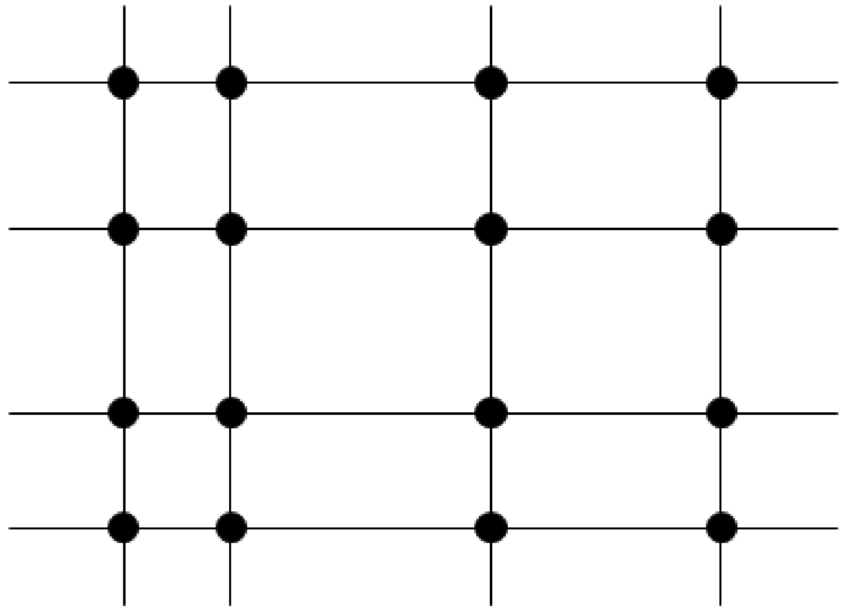

$$P_c = \frac{1}{f_{i+1} - f_i}\left[P_i\left(f_{i+1} - f_c\right) + P_{i+1}\left(f_c - f_i\right)\right]$$

ohm

# Isolines

## Rectilinear meshes

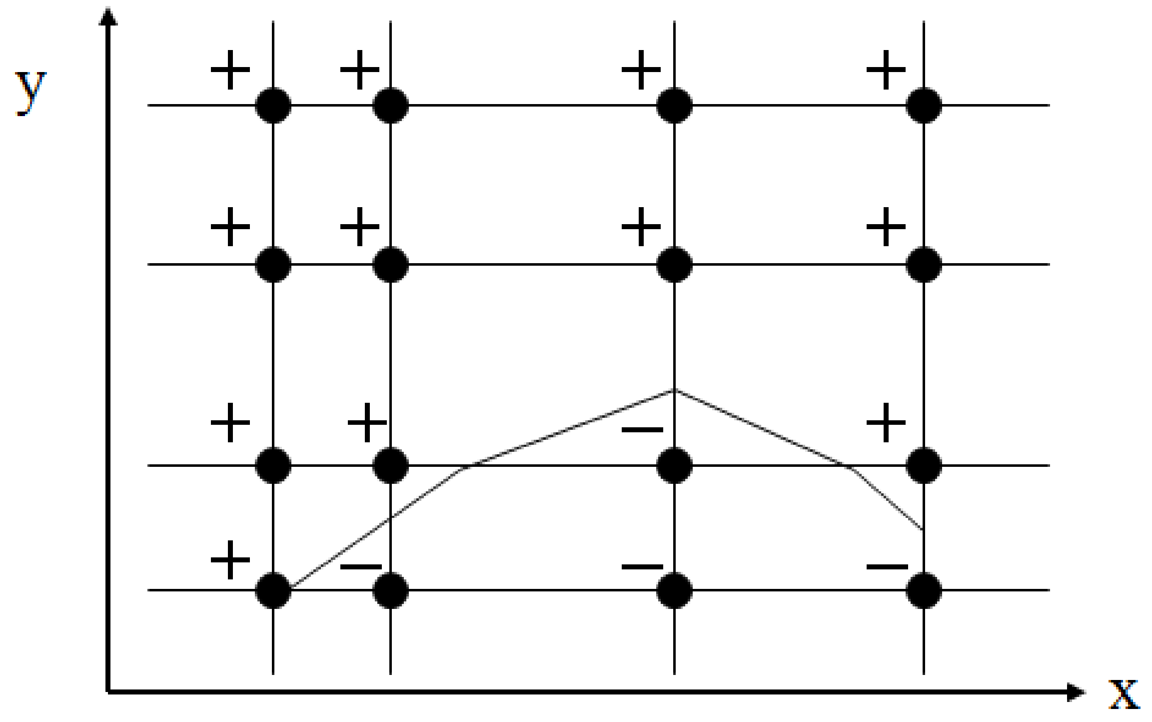
- Representation of scalar function on a rectilinear grid
  - Scalar values are given at each vertex $f \leftrightarrow f_{ij}$
  - Consider interpolation within cells (bilinear!)
  - *Isolines cannot be missed*
- Divide and conquer - "**Marching Squares**"
  - Consider cells *independently* of each other

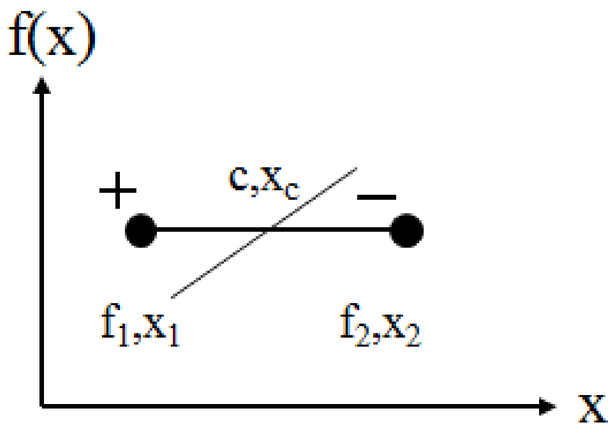


# Isolines

## Marching Squares

- Which cells will be intersected?
  - Initially mark all vertices with "+" or "-", depending on the condition $f_{ij} \geq c$ , $f_{ij} < c$
  - Isoline *does not pass* through cells which have the *same sign* at all four vertices
    - Only determine edges with different signs and find intersection by linear interpolation



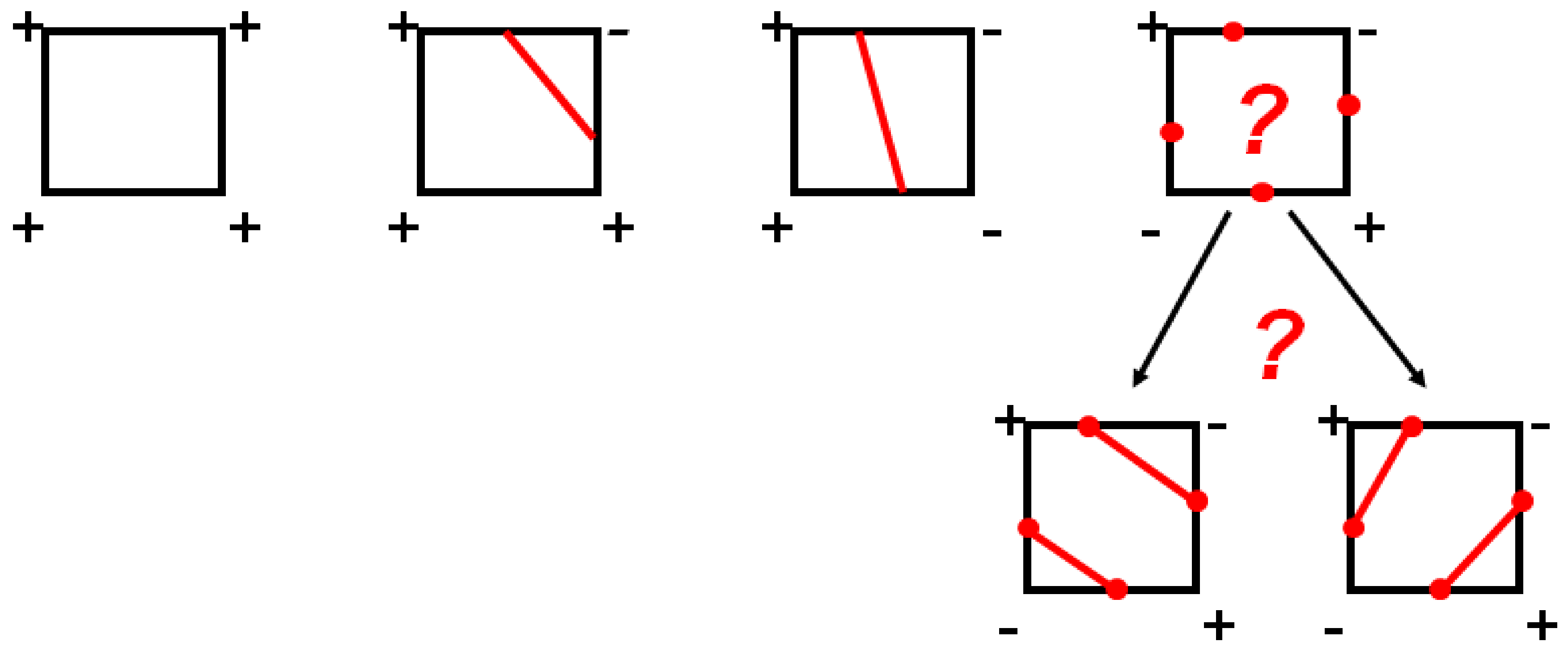




$$x_c = \frac{1}{f_2 - f_1}\left[(f_2 - c)x_1 + (c - f_1)x_2\right]$$

# Isolines

## Marching Squares

- There are $2^4 = 16$ different intersection possibilities (combinations of signs)
  - Using symmetries, only 4 different cases remain (rotation, reflection)
  - Compute intersections between isoline and cell edge (linear interpolation)




  - Distinguish ambiguous case by a decider

ohm

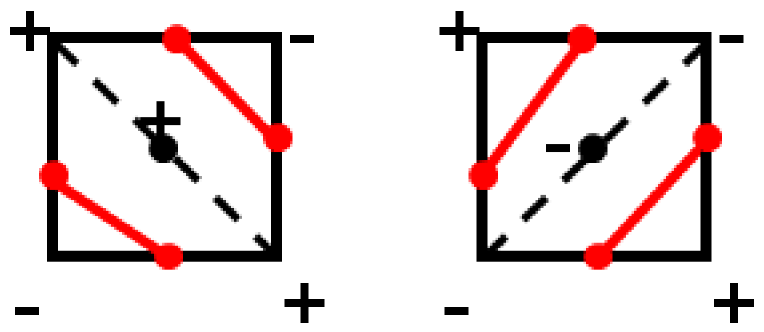
# Isolines

## Mid-point decider

- Interpolate the function value at the center

$$f_{\text{center}} = \frac{1}{4}\left(f_{i,j} + f_{i+1,j} + f_{i,j+1} + f_{i+1,j+1}\right)$$

- If $f_{\text{center}} < c$ choose right case, otherwise left case

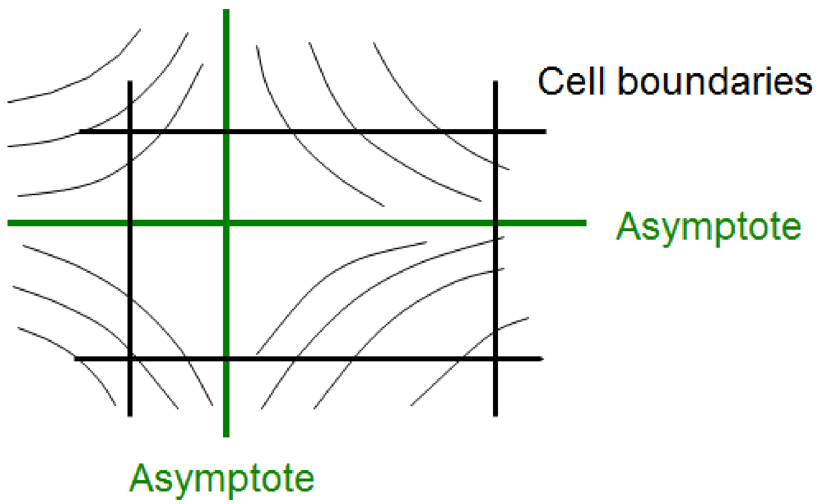


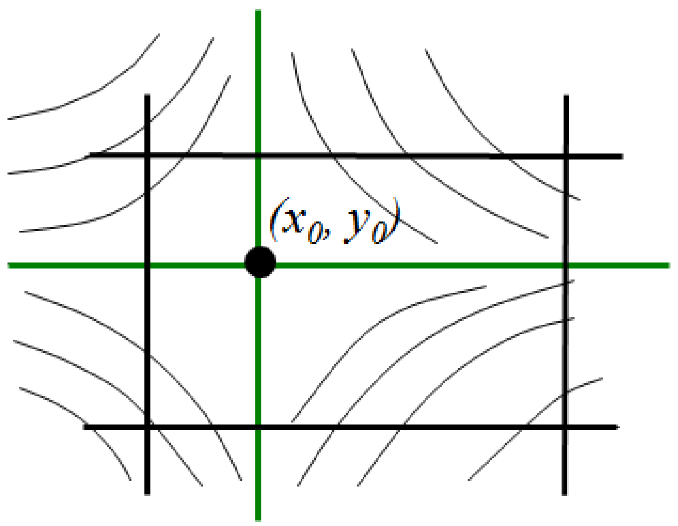

- This is not always the correct solution!

ohm

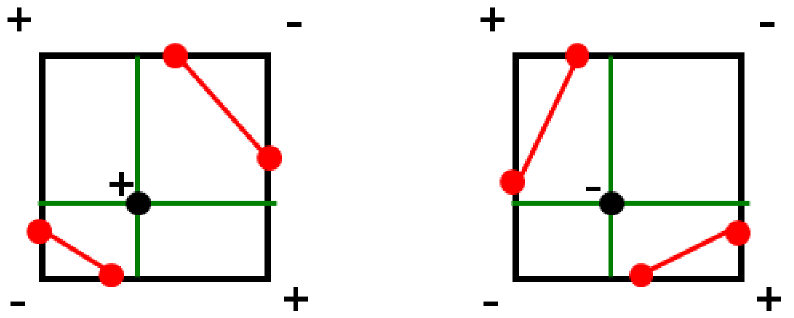
# Isolines

## Asymptotic decider



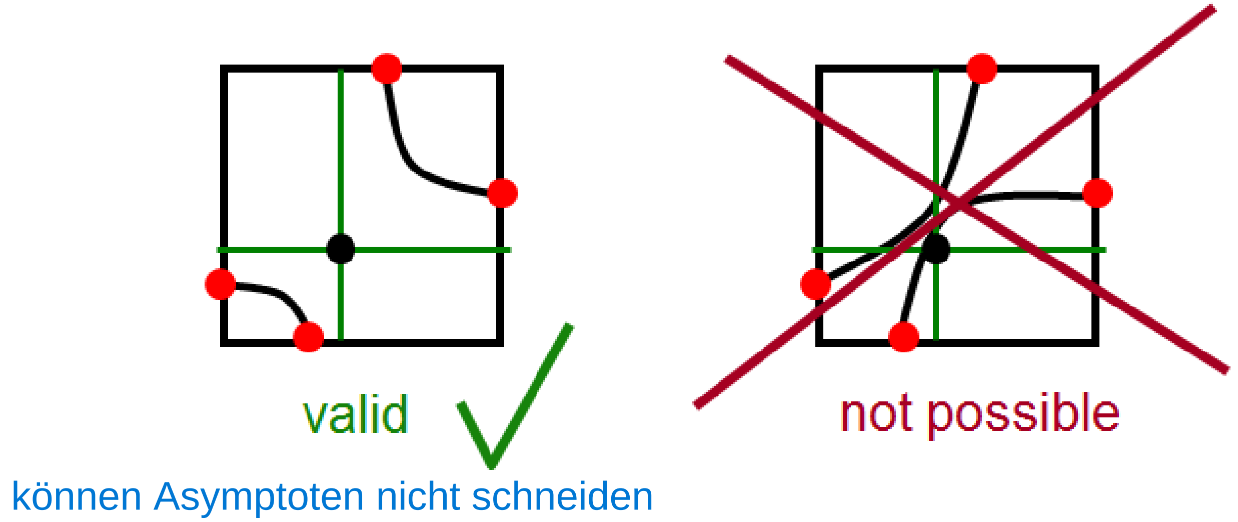

- We know that the true isolines within a cell are hyperbolas
  - Why? - Consider the bilinear interpolant within a cell
    - i.e. find stationary point where grad f = 0
- Interpolate function bilinearly $f(x,y) = f_{i,j}(1-x)(1-y) + f_{i+1,j}x(1-y) + f_{i,j+1}(1-x)y + f_{i+1j+1}xy$
- Transform to normal form by comparison of coefficients $f(x,y) = \eta(x-x_0)(y-y_0) + \gamma$
  - $\gamma$ is the function value in the intersection point of the asymptotes
  - If $\gamma \leq$ c choose right case, otherwise left case



# Isolines

## Explicit transformation of f can be avoided

- Consider order of intersection points either along x or y axis
  - Build pairs of first two and last two intersections




  - Isolines cannot intersect asymptotes!

ohm

# Isolines

## Algorithms: Cell order approach for marching squares

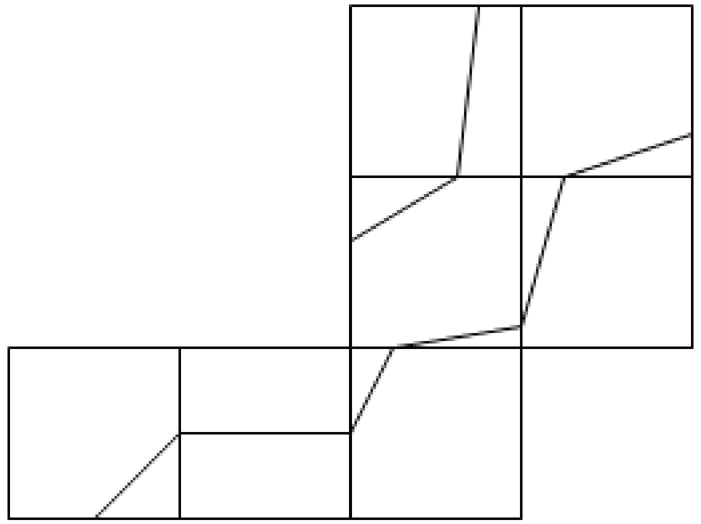
- Apply marching squares to each individual cell

  - Advantage
    - Simple approach
  - Disadvantage
    - Process every vertex and every edge twice
    - Output is just a collection of pieces of isolines which have to be post-processed to get (closed) isoline

# Isolines

## Algorithms: Contour tracing approach

- Start at a seed point of the isoline
  - Move to neighboring cell (into which the isoline enters)
  - Trace isoline until bounds of domain are reached or isoline is closed
- Advantage
  - Polygons
  - Information about inclination
- Disadvantage
  - Marking of processed cells (list of started contours)
  - Problem: how to find seed points efficiently?
  - Remove marker from traversed cells (unless 4 intersections!)



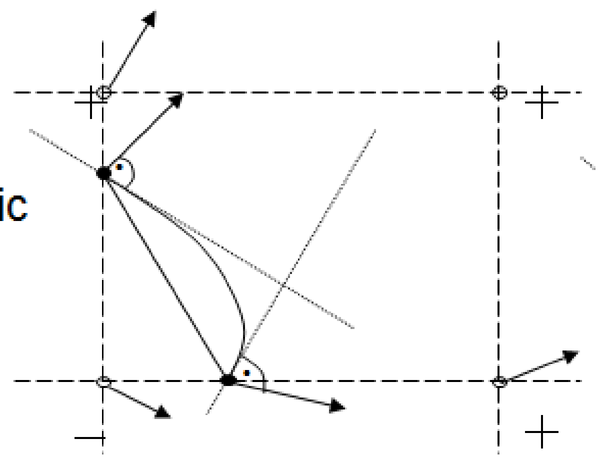
ohm

# Isolines

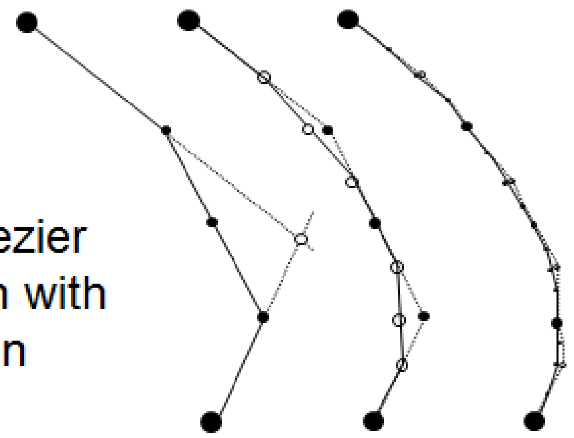
## Smoothing isolines

- Evaluate gradient at vertices by central differences
- Estimate tangents at intersection points by linear interpolation - Gradient is perpendicular to isoline!
- Draw parabolic arc which is tangential to the estimated tangents at the intersections
  - Quadratic Bezier curve (2-3 subdivisions per step is sufficient!)

Quadratic Bezier



Quadratic Bezier approximation with subdivision



# Isolines

## Summary

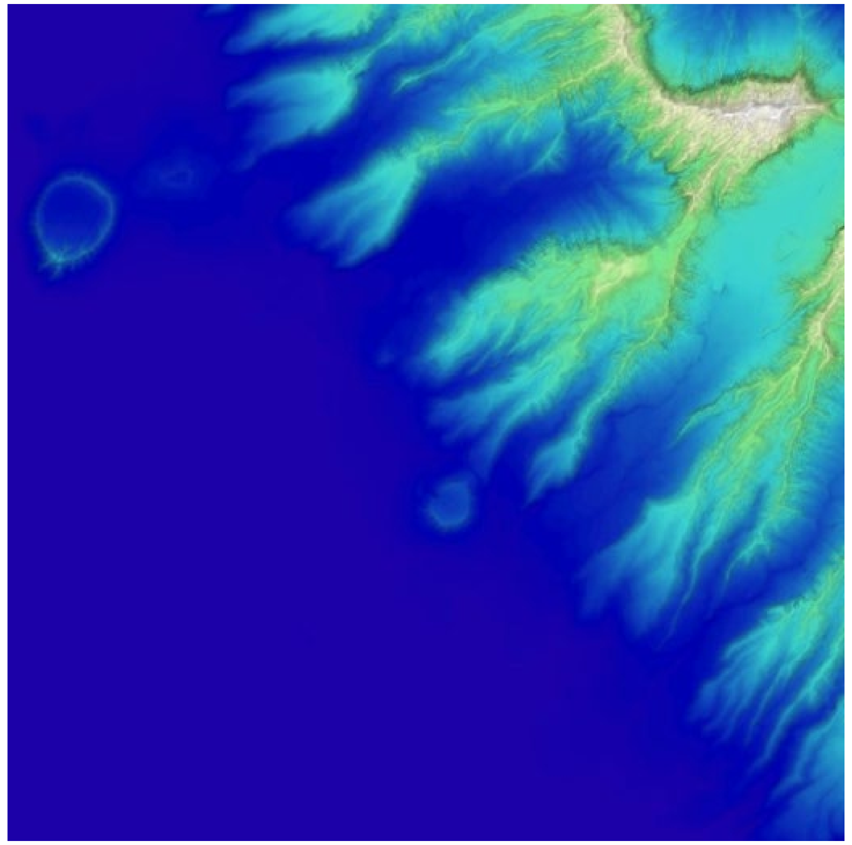
- Smoothing by interpolation falsifies the contour
- Multiple isolines
  - Line type (dashed, dotted, ... ), annotation, color
- Advantages
  - Line plots in black/white possible
  - Annotation within image
  - Shows gradient within image
  - Simple extension for complex grids
- Disadvantages
  - Max/min not directly visible
  - Strongly depends on choice of isovalue
  - No use of color and rasterization features

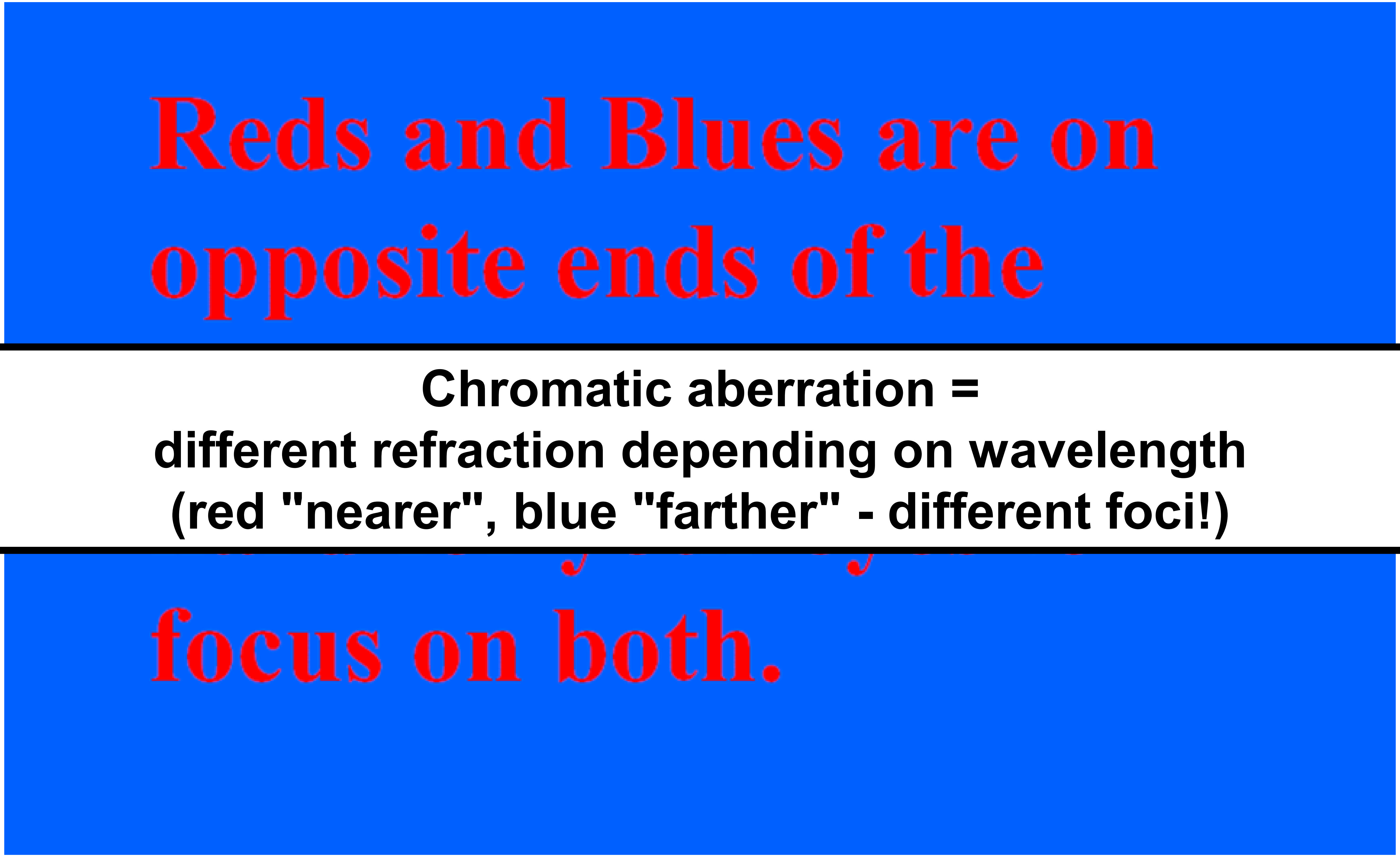
# 4.3 Color Coding

# Color Coding

- Questions
  - What kind of data can be color-coded?
  - What kind of information can be efficiently visualized?
- Areas of application
  - Provide information coding
  - Designate / emphasize target in crowded display
  - Provide sense of realism (or virtual realism)
  - Warning signals or signify low probability events
  - Group, categorize and chunk information
  - Convey emotional content
  - Provide an aesthetically pleasing display
- Possible problems
  - Distract the user when inadequately used
  - Dependent on viewing and stimulus conditions
  - Ineffective for color deficient individuals
  - Results in information overload
  - Unintentionally conflict with cultural conventions
  - Cause unintended visual effects and discomfort





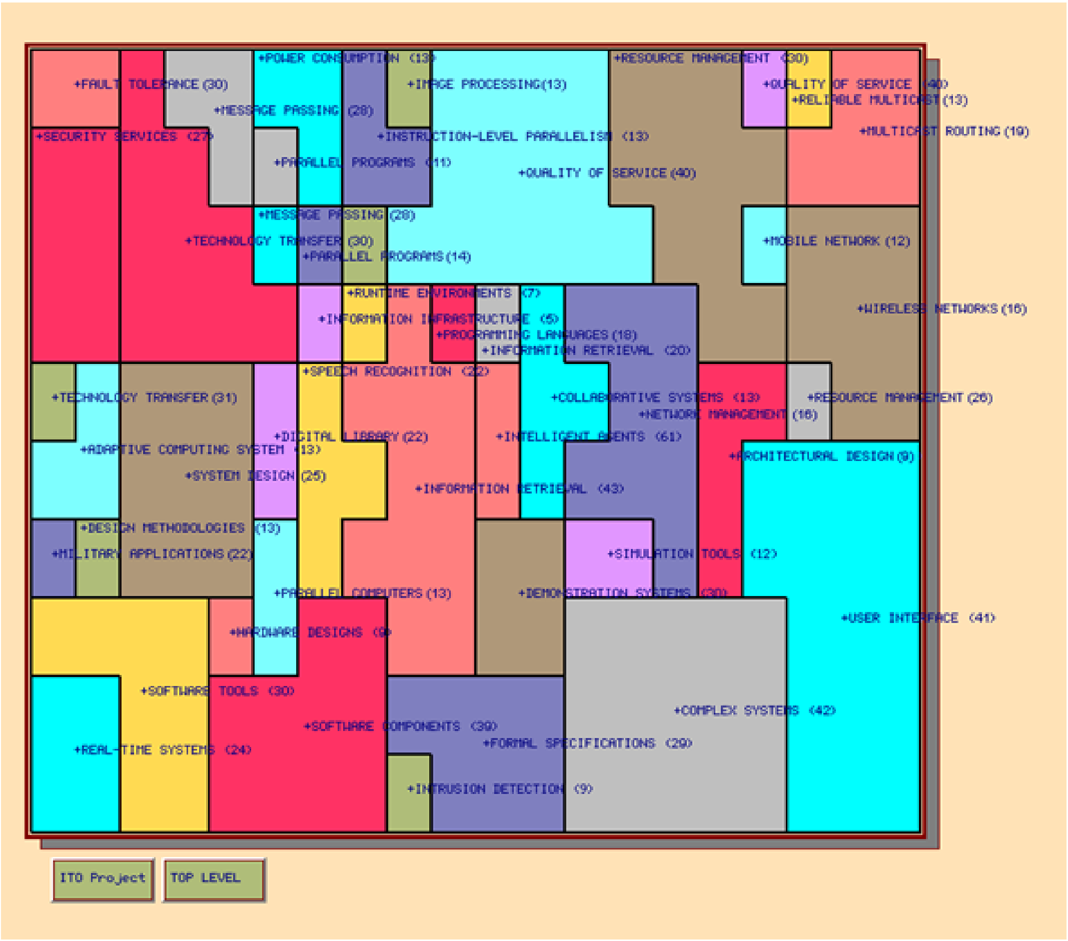

**Chromatic aberration =**
**different refraction depending on wavelength**
**(red "nearer", blue "farther" - different foci!)**

ohm

# Color Coding

## Nominal data

- Colors need to be distinguished
- Localization of data
- Around 8 different  basis colors

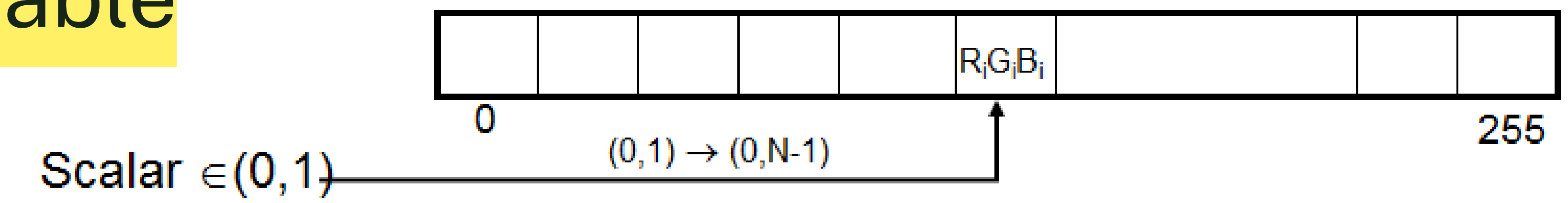



co-citation analysis

# Color Coding

## Ordinal and interval data

- Represent ordering of data by ordering of colors
  - Psychological aspects
  - $x_1 < x_2 < \ldots < x_n \rightarrow E(c_1) < E(c_2) < \ldots < E(c_n)$
- Color coding for scalar data
  - Assign to each scalar value a different color value
  - Assignment via transfer function T : scalar value → color value (RGB or HSV)
  - Code color values into a color lookup table

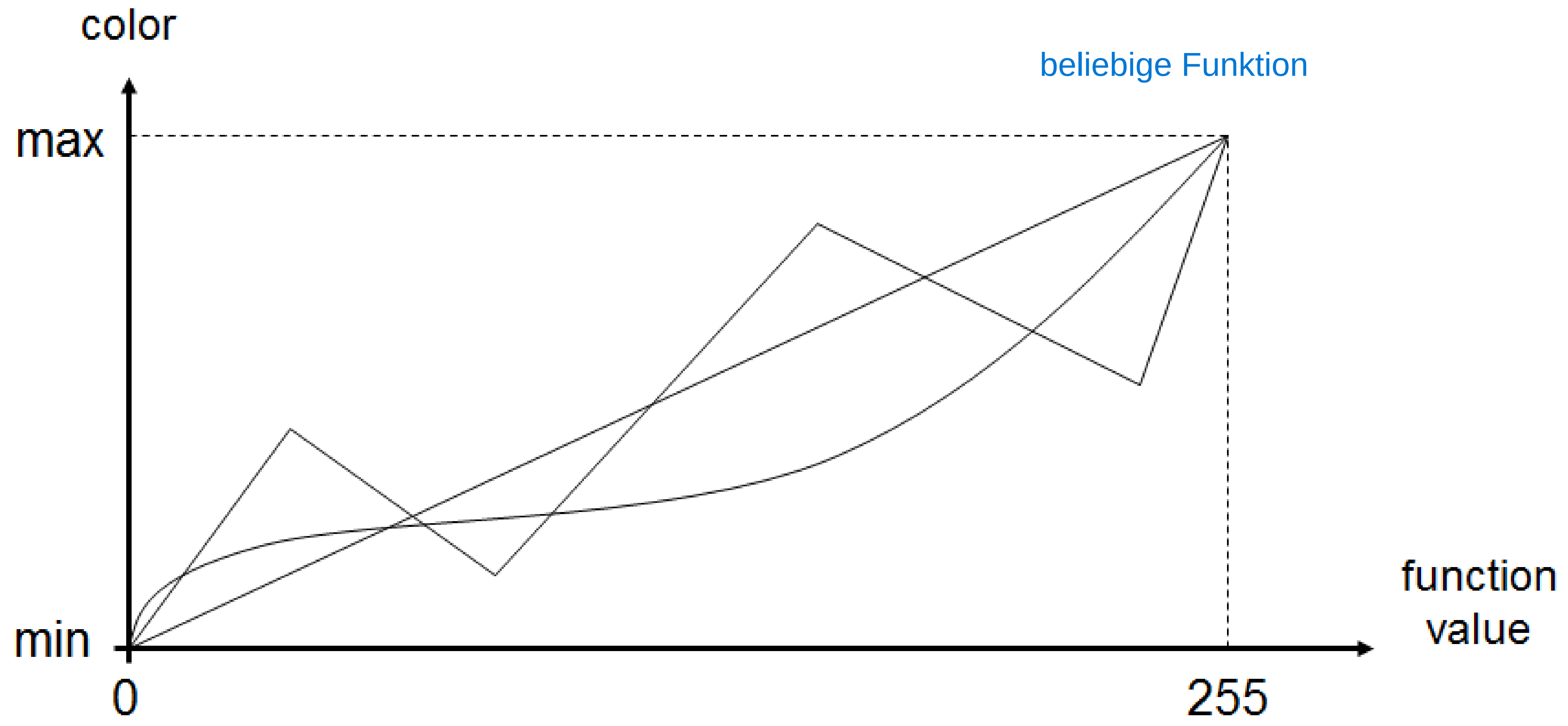


ohm

# Color Coding

## Assignment via transfer function T

- T: scalar value → color value (RGB, HSV, ...)



ohm

# Color Coding

## Linear transfer function for color coding

- Specify color for $f_{min}$ and $f_{max}$
  - ($R_{min}$ ,$G_{min}$ , $B_{min}$) and ($R_{max}$ , $G_{max}$ , $B_{max}$)
  - Linearly interpolate between them

$$f \mapsto \frac{f - f_{min}}{f_{max} - f_{min}}(R_{min}, G_{min}, B_{min}) + \frac{f_{max} - f}{f_{max} - f_{min}}(R_{max}, G_{max}, B_{max})$$

lineare Interpolation

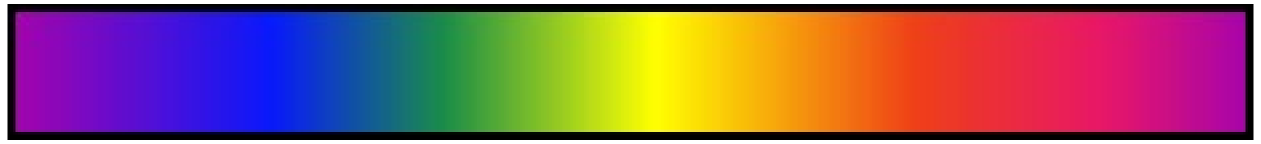
- Different color spaces
  - Lead to different interpolation functions
  - In order to visualize (enhance / suppress) specific details, non-linear color lookup tables are needed

# Color Coding

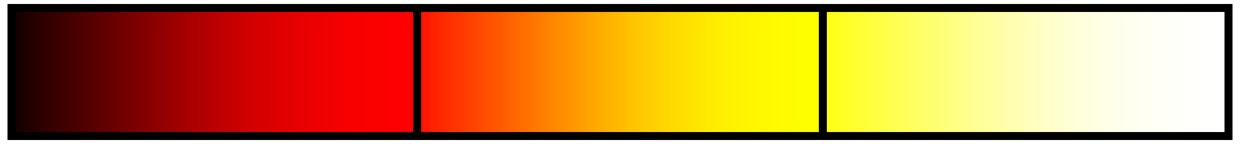
## Examples of different color tables

- Gray scale
  - Intuitive ordering

- Rainbow
  - Less intuitive



- Temperature



# Color Coding

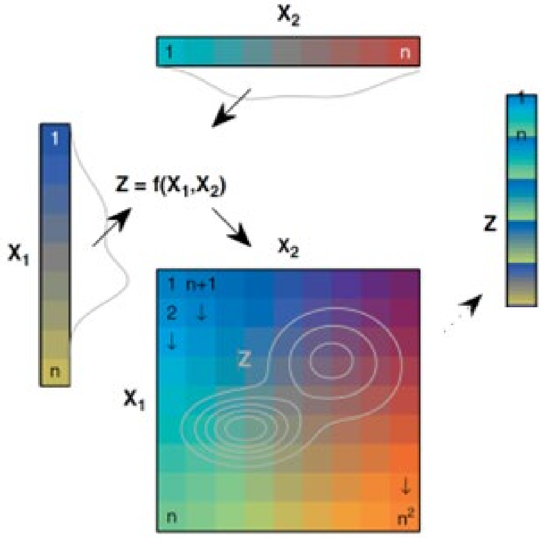
## Bivariate and trivariate color tables

2-dim 3-dim

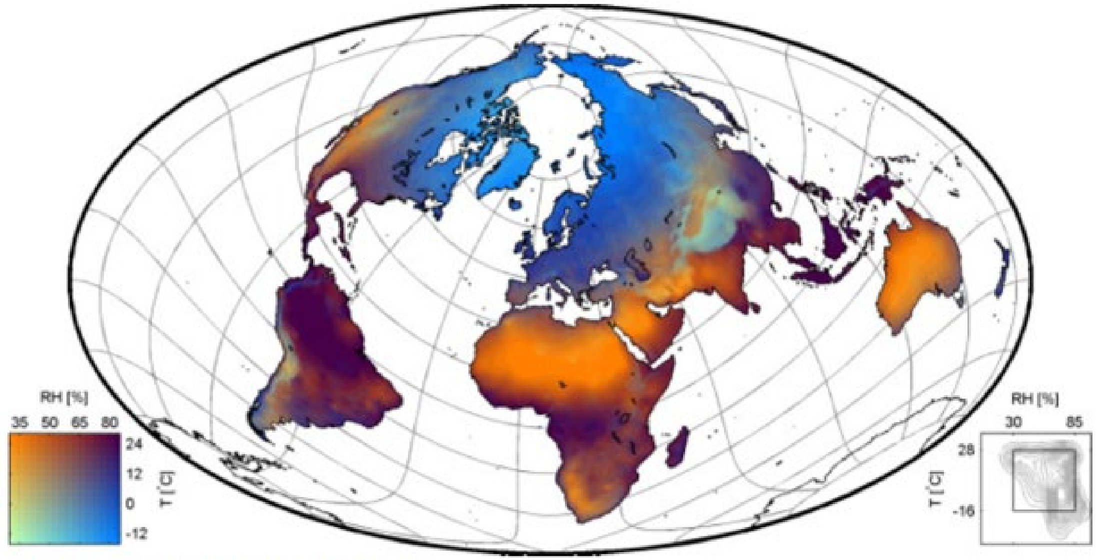
- Not very useful as there is no intuitive ordering
  - Colors hard to distinguish
- Many more color tables
  - For specific applications
  - Design of good color tables depends on psychological / perceptional issues
- Important requirement
  - Often, interactive specification of transfer function is needed to extract important features (which leads to nonlinear transfer functions), e.g. medical data

# Color Coding

## Examples



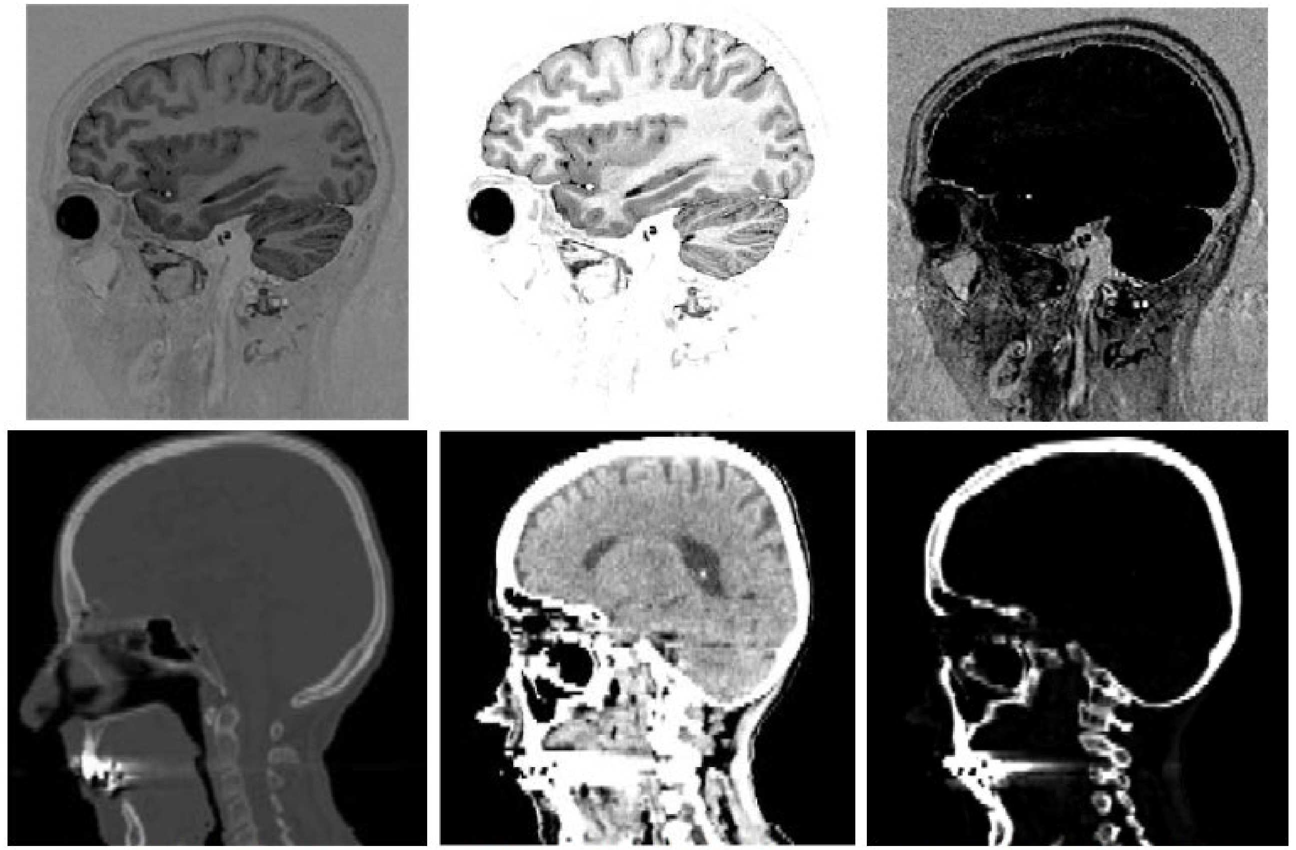

Distribution of world's climate zones
*(period: 1961–1990)*
$x_1$: average near-surface temperature (T )
$x_2$: relative humidity (RH) for the




*Int. J. Climatol.* **31**: 1408–1412 (2011)

Magnetic Resonance Imaging (MRI)

Computed Tomography (CT)



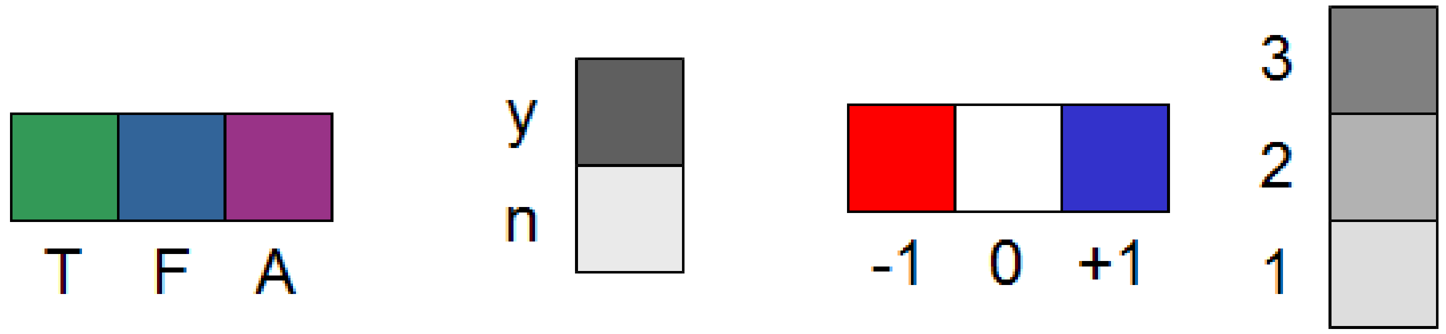
Linear mapping of all data values

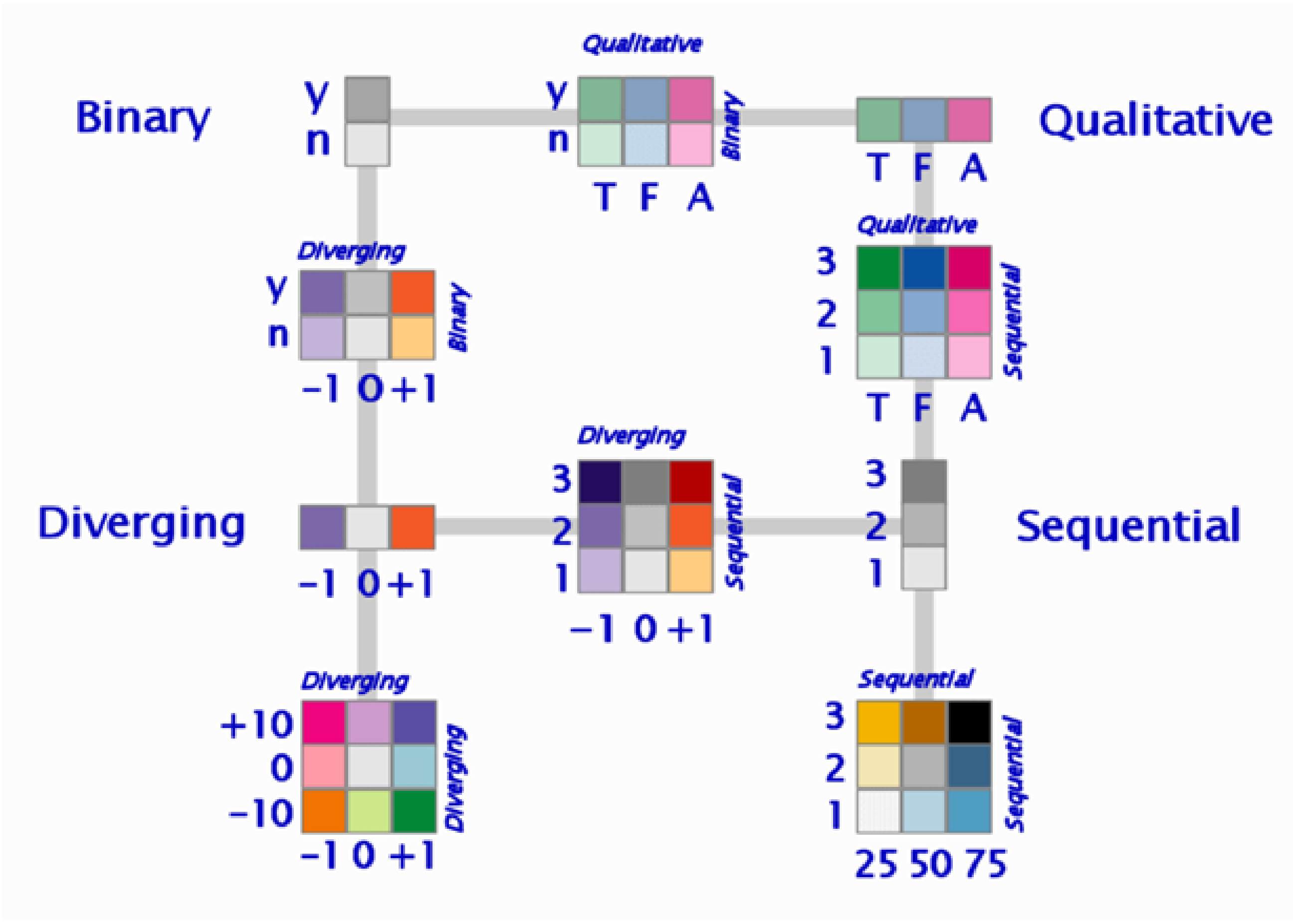
Special transfer function for brain tissue

Special transfer function for bone structure

# Color Coding

- Color guidelines
  - [C.A. Brewer, 1994, "Color Use Guidelines for Mapping and Visualization", Chapter 7 (pp. 123-147) in Visualization in Modern Cartography]
- 4 different types of color schemes
  - Qualitative - data of different categories
    - Represented by differences in hue
    - E.g. blue - rivers, brown - mountains, red - road
  - Binary - only two categories
    - Use difference in hue or lightness
  - Diverging - data showing different trends
    - Use colors with very different hues in extremes
    - E.g. red - blue, red - green (yellow - blue for color blind)
  - Sequential - ordered data in sequential form
    - Use variations in lightness or saturation of given hue
- Bivariate color schemes. www.infovis.net (2006, No. 184)



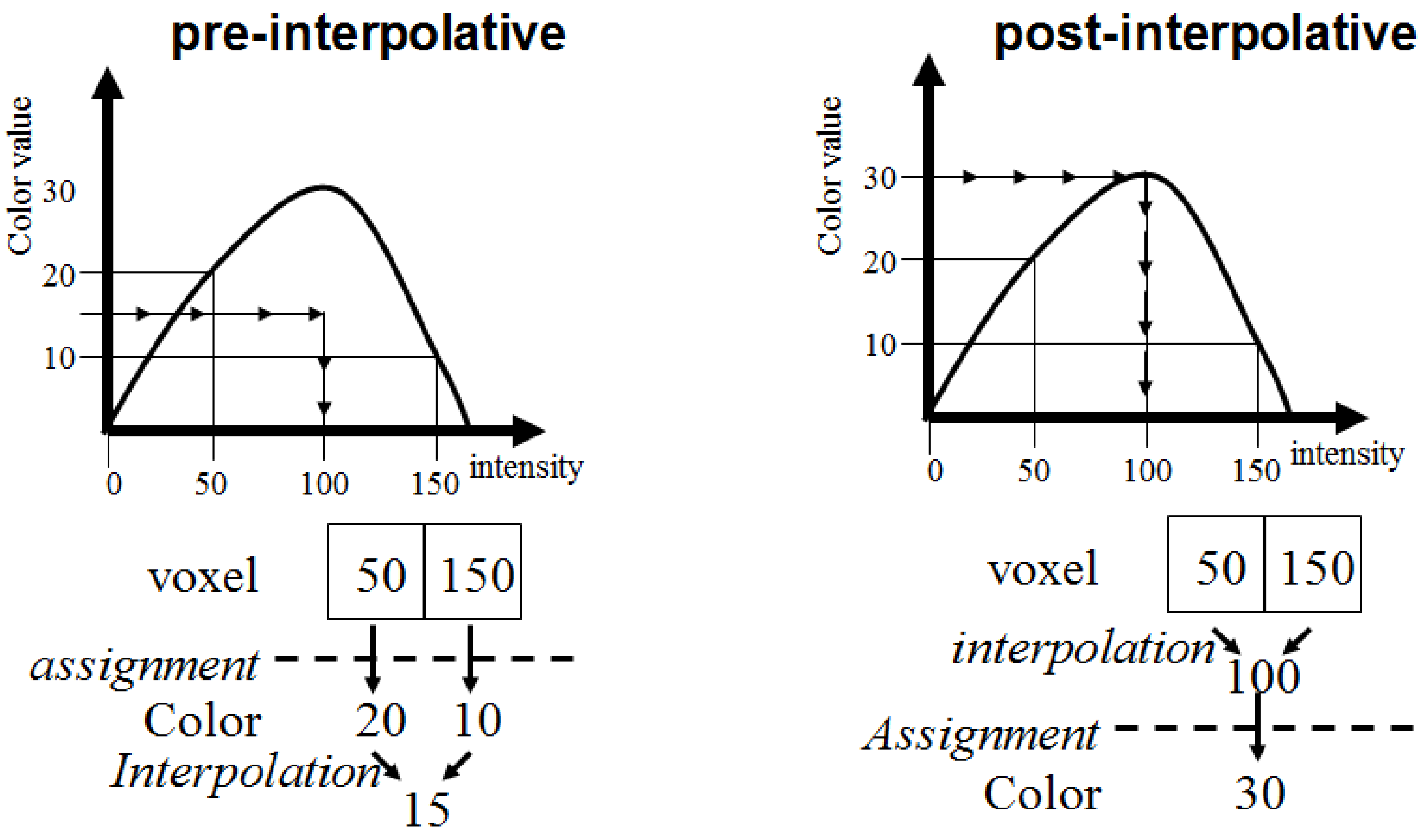



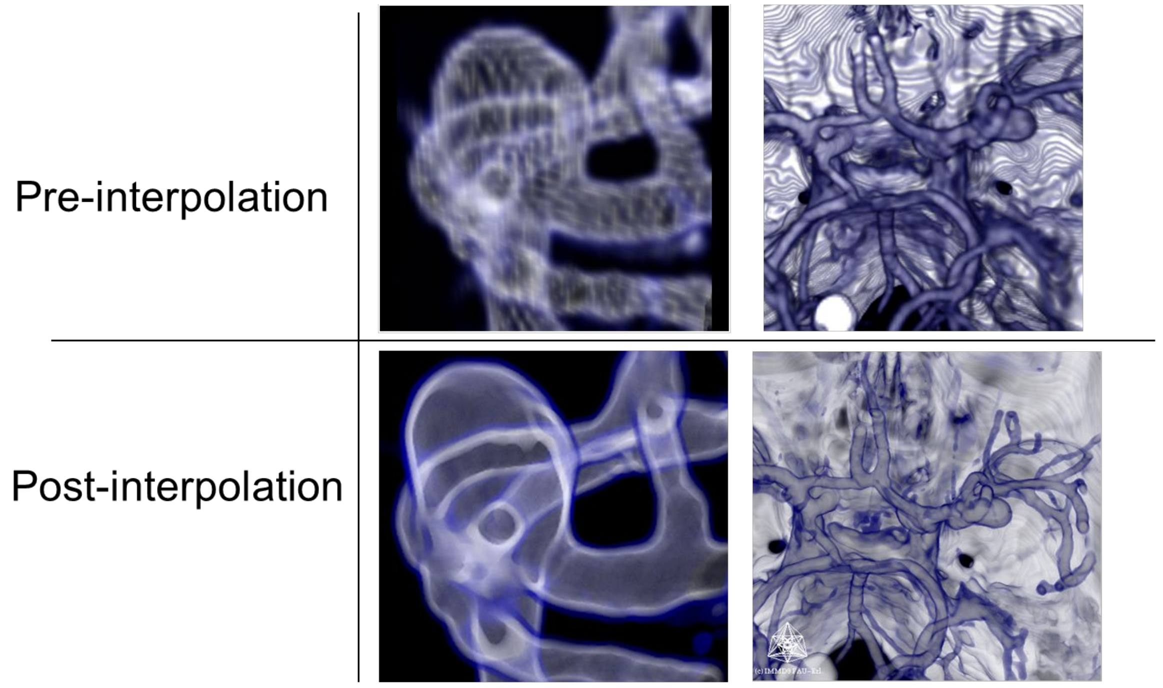
ohm

# Color Coding

## Pre-shading vs. post-shading

- Pre-shading (pre-interpolation)
  - Assign color values to original function values (i.e. vertices of a cell)
  - Interpolate between color values (within a cell)
- Post-shading (post-interpolation)
  - Interpolate between scalar values (within a cell)
  - Assign color values to interpolated scalar values

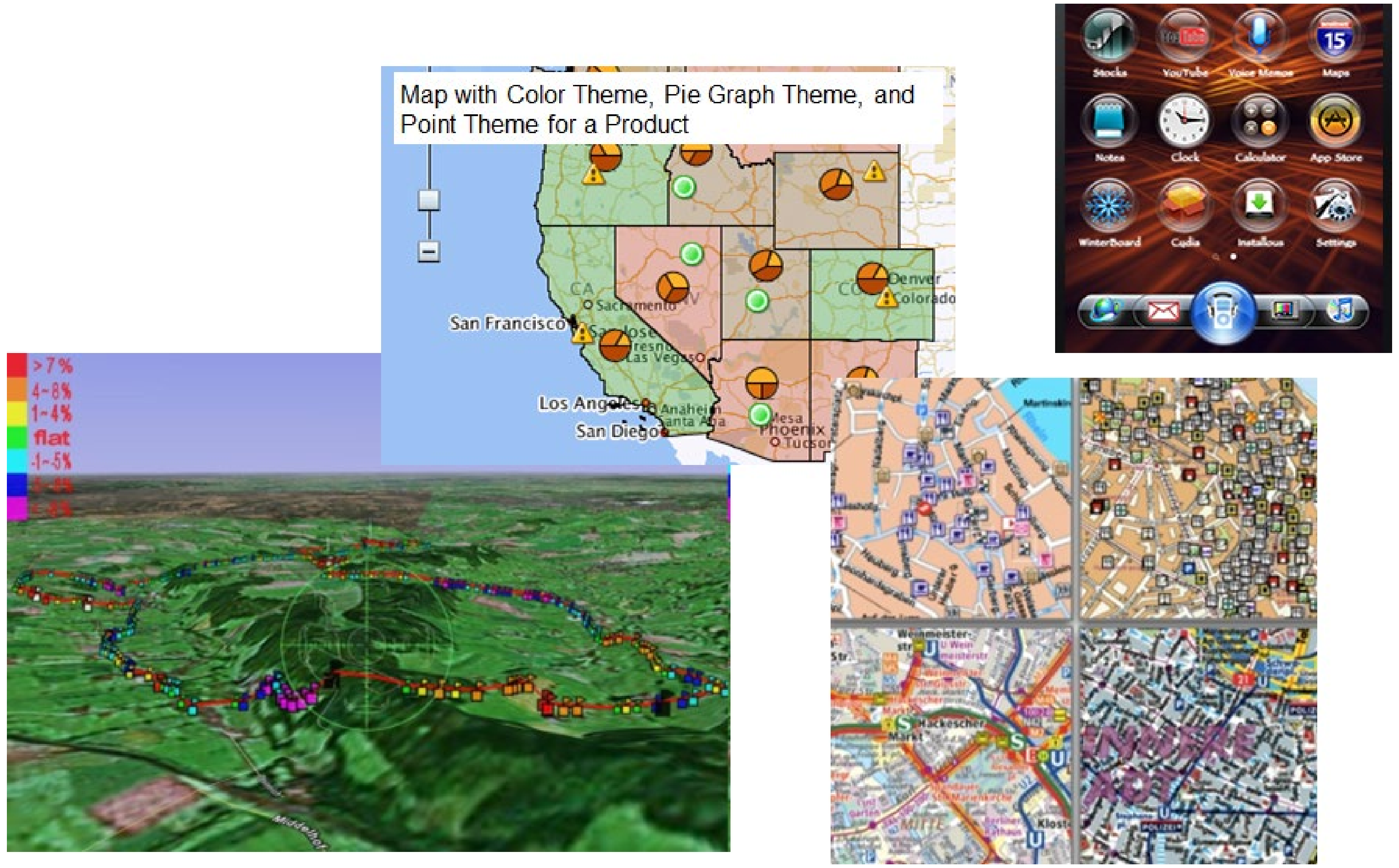


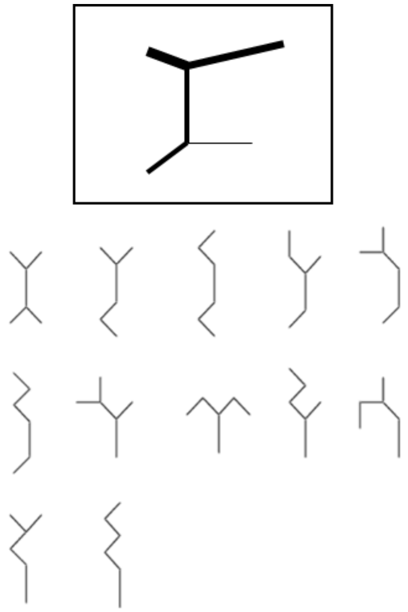



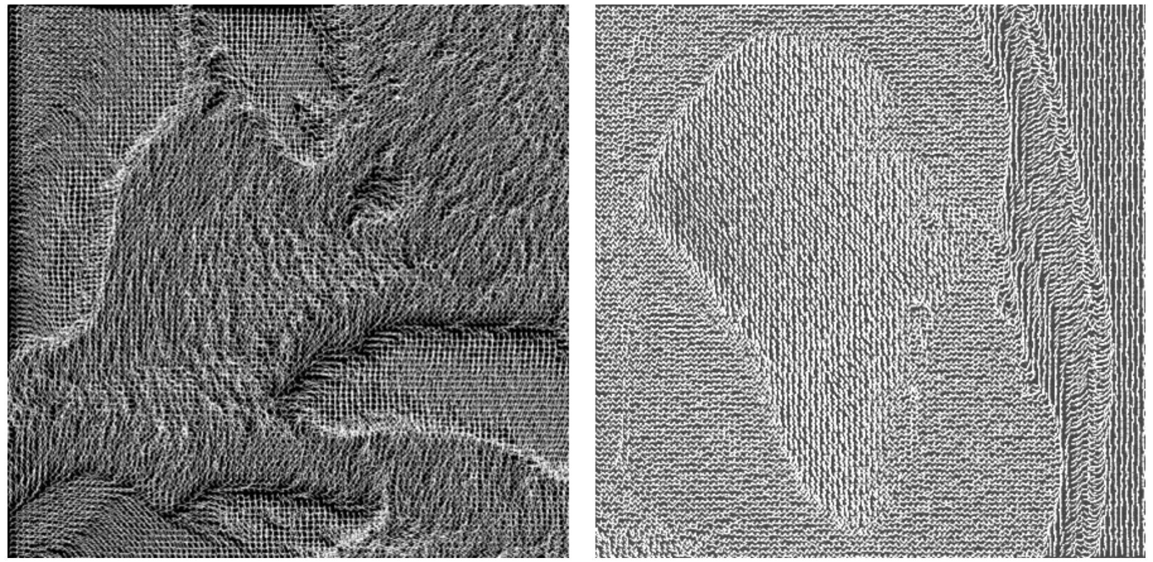
# 4.4 Glyphs, Icons and others…

# Glyphs and Icons

- Features
  - Should be easy to distinguish and combine
- Icons
  - Should be separated from each other
- Mainly used for multivariate data
  - More than one data value at each coordinate
- Stick-figure icon [Picket & Grinstein 88]
  - 2D figure with 4 limbs
  - Coding of data via
    - Length, thickness,
      angle with vertical axis
  - 12 attributes
  - Exploits human capability
    to recognize patterns/textures

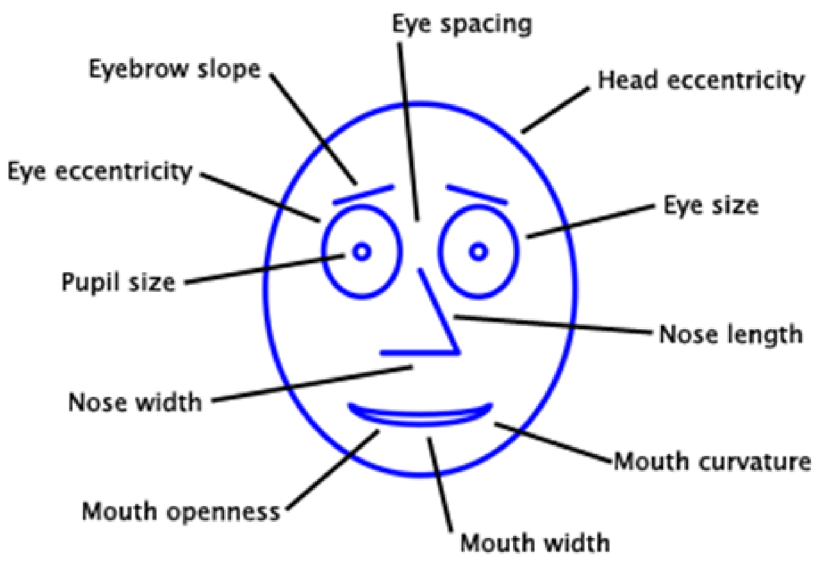




# Glyphs and Icons

- Chernoff faces/icons
  [H. Chernoff (1973). The use of faces to represent points in k-dimensional space graphically. Journal of the American Statistical Association , 68 :361-368]
  - Each facial feature represents one variable
  - Useful for qualitative and multivariate data
  - Human ability to distinguish small features in faces
  - Useful for sparse data representations
  - Possible assignment in decreasing order of importance
    Area of face, shape of face, length of nose, location of mouth
    - Curve of smile, width of mouth, location, separation, angle, shape and width of eyes, location of pupil
    - Location, angle and width of eyebrows
  - Additional variables
    - Could be encoded by making faces asymmetric




Codierung von Daten
in einem Icon (=Gesicht)

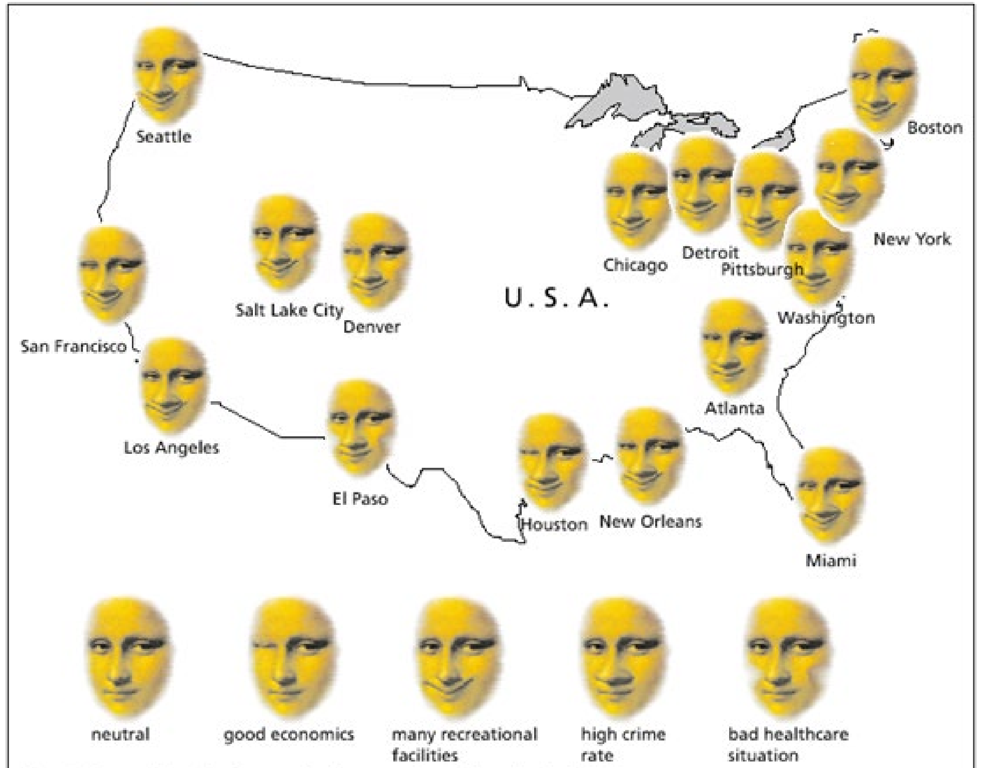
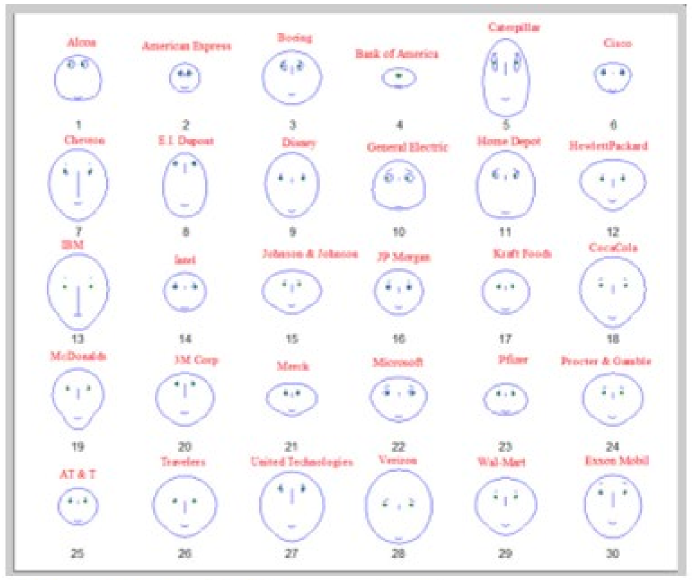
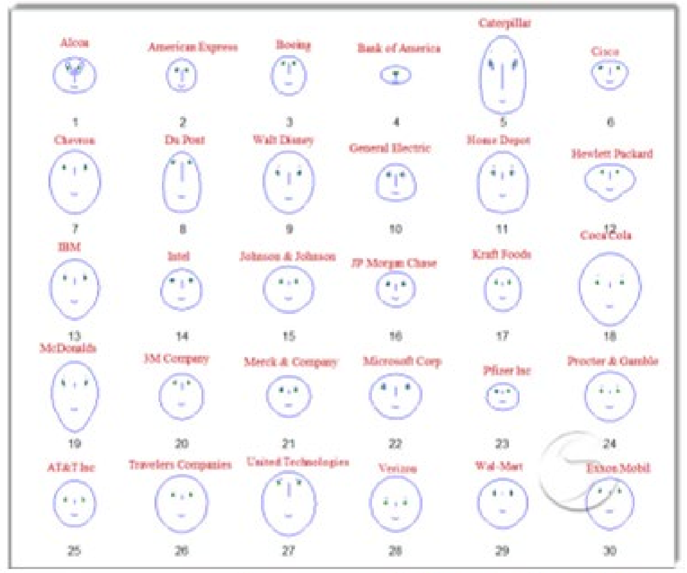



Figure 1: Distorted Mona Lisa faces as visual representations of city ratings in the U.S.



31 January 2011



15 December 2010

| A | B | C |
|---|---|---|
| Column | Attribute | ValuEngine Model Variable |
| 1 | Size of face | Valuation (+/-) % |
| 2 | Forehead/jaw relative arc length | Average Volume |
| 3 | Shape of forehead | Last 12-m Return % |
| 4 | Shape of jaw | Sharpe Ratio |
| 5 | Width between eyes | 5-Year Return % |
| 6 | Vertical position of eyes | 1-Month Forecast Return % |
| 7 | Height of eyes | Beta |
| 8 | Width of eyes (this also affects eyebrow width) | Volatility % |
| 9 | Angle of eyes (this also affects eyebrow angle) | Expected EPS |
| 10 | Vertical position of eyebrows | Market Capitalization |
| 11 | Width of eyebrows (relative to eyes) | P/E Ratio |
| 12 | Angle of eyebrows (relative to eyes) | M/B Ratio |
| 13 | Direction of pupils | P/S Ratio |
| 14 | Length of nose | Actual EPS |
| 15 | Vertical position of mouth | Forecast EPS |