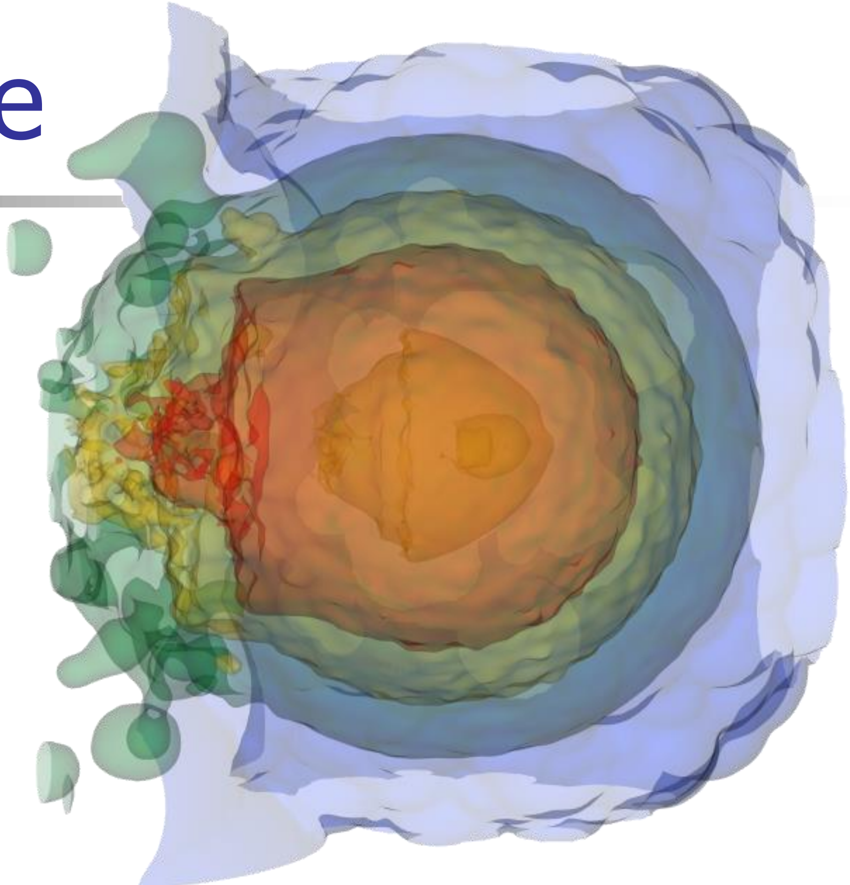
Data visualisatie



Robert G. Belleman
Universiteit van Amsterdam

FCG: Ch. 26 and other sources

Achtergrond

Wat is visualisatie?

- De transformatie van gegevens in plaatjes
- Grafische weergave van gegevens en informatie in beelden gemaakt met behulp van computer graphics
- Gebruik van het menselijk visueel systeem voor de interpretatie van grote hoeveelheden en/of complexe gegevens aan de hand van computer gegenereerde plaatjes

Waarom visualisatie?

- De hoeveelheid en complexiteit van gegevens neemt explosief toe terwijl de middelen om die te kunnen verwerken beperkt zijn

Waarom visualisatie?

X	Y	X	Y	X	Y	X	Y
10,00	8,04	10,00	9,14	10,00	7,46	8,00	6,58
8,00	6,95	8,00	8,14	8,00	6,77	8,00	5,76
13,00	7,58	13,00	8,74	13,00	12,74	8,00	7,71
9,00	8,81	9,00	8,77	9,00	7,11	8,00	8,84
11,00	8,33	11,00	9,26	11,00	7,81	8,00	8,47
14,00	9,96	14,00	8,10	14,00	8,84	8,00	7,04
6,00	7,24	6,00	6,13	6,00	6,08	8,00	5,25
4,00	4,26	4,00	3,10	4,00	5,39	19,00	12,50
12,00	10,84	12,00	9,11	12,00	8,15	8,00	5,56
7,00	4,82	7,00	7,26	7,00	6,42	8,00	7,91
5,00	5,68	5,00	4,74	5,00	5,73	8,00	6,89

A B C D

Waarom visualisatie?

X	Y	X	Y	X	Y	X	Y
10,00	8,04	10,00	9,14	10,00	7,46	8,00	6,58
8,00	6,95	8,00	8,14	8,00	6,77	8,00	5,76
13,00	7,58	13,00	8,74	13,00	12,74	8,00	7,71
9,00	8,81	9,00	8,77	9,00	7,11	8,00	8,84
11,00	8,33	11,00	9,26	11,00	7,81	8,00	8,47
14,00	9,96	14,00	8,10	14,00	8,84	8,00	7,04
6,00	7,24	6,00	6,13	6,00	6,08	8,00	5,25
4,00	4,26	4,00	3,10	4,00	5,39	19,00	12,50
12,00	10,84	12,00	9,11	12,00	8,15	8,00	5,56
7,00	4,82	7,00	7,26	7,00	6,42	8,00	7,91
5,00	5,68	5,00	4,74	5,00	5,73	8,00	6,89

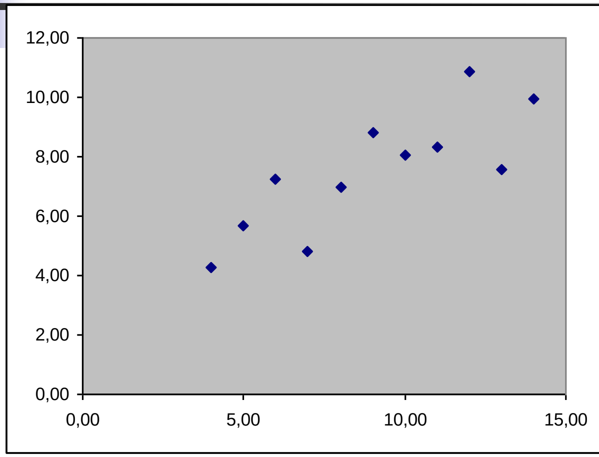
A **B** **C** **D**

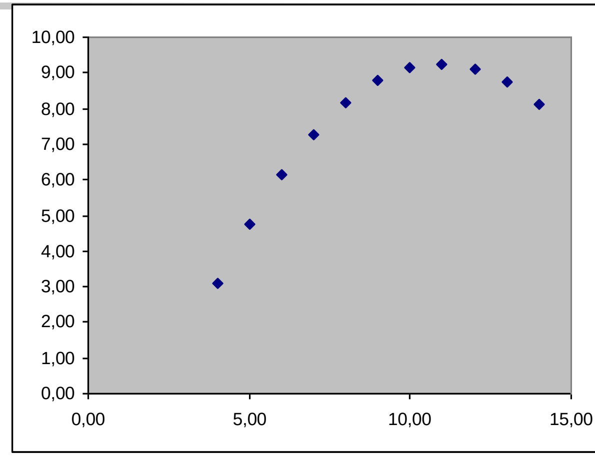
$\mu_x = 9.00$, $\sigma_x = 3.32$

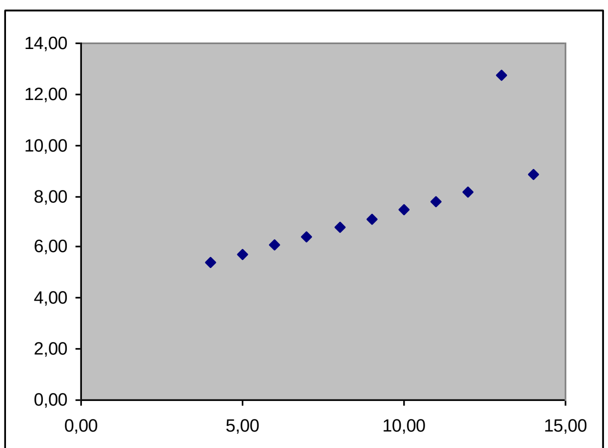
$\mu_y = 7.50$, $\sigma_y = 2.03$

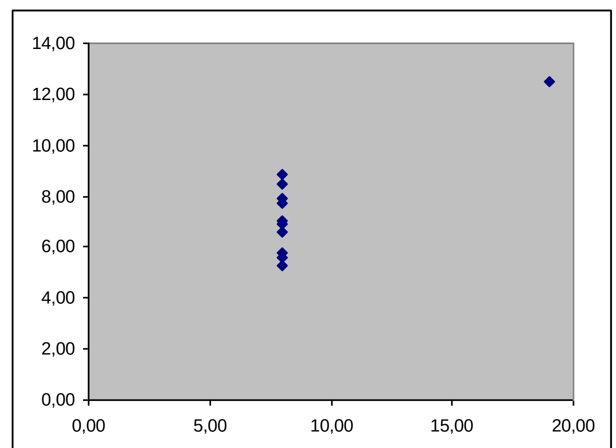
linear regression: $y = 0.5\,x + 3.0$

Waarom visualisatie?







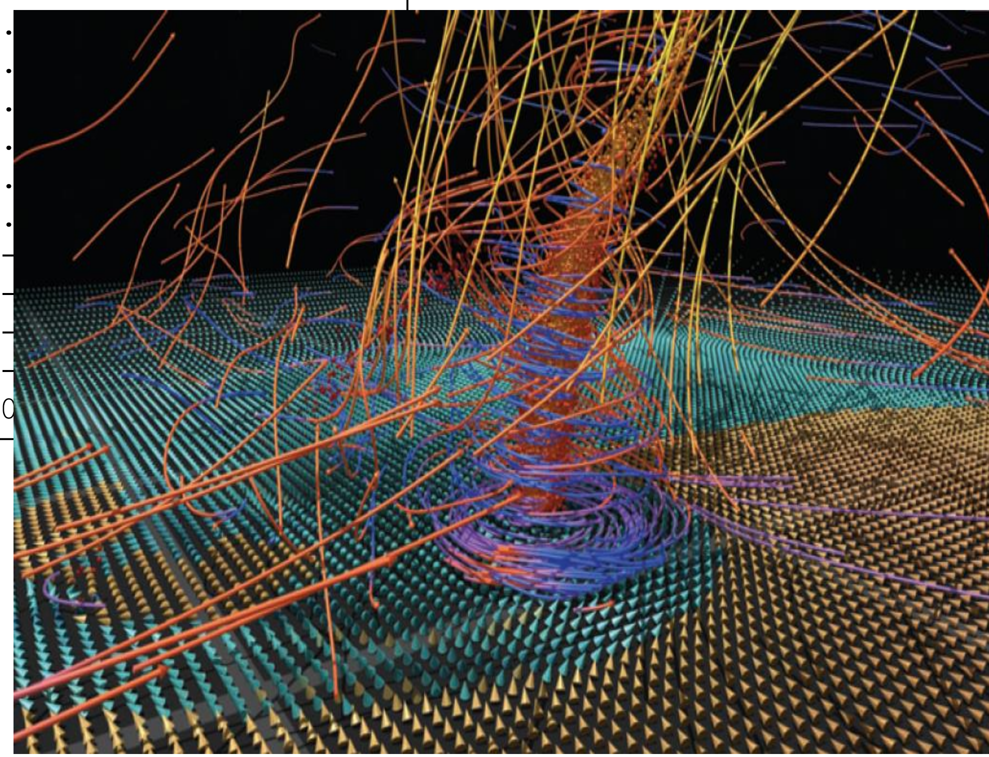


Waarom visualisatie?

van:

```
1.980000 0.027228 -0.013616 1.990000 0.038601 -0.015680
1.990000 0.050290 -0.017887 2.000000 0.062348 -0.020251
2.010000 0.074714 -0.022829 2.020000 0.087367 -0.025576
2.030000 0.100000 -0.028577 2.050000 0.110000 -0.031821
2.060000 0.130000 -0.035351 2.080000 0.140000 -0.039209
2.100000 0.150000 -0.043431 2.120000 0.
2.140000 0.170000 -0.053352 2.170000 0.
2.190000 0.170000 -0.066239 2.210000 0.
2.230000 0.170000 -0.084400 2.250000 0.
2.260000 0.130000 -0.110000 2.260000 0.
2.250000 0.063758 -0.160000 2.230000 0.
2.190000 -0.038076 -0.230000 2.120000 -
2.020000 -0.160000 -0.340000 1.880000 -
1.700000 -0.260000 -0.470000 1.460000 -
1.150000 -0.210000 -0.510000 0.720000 -
0.380000 -0.033295 -0.230000 0.000001 0
```

naar:



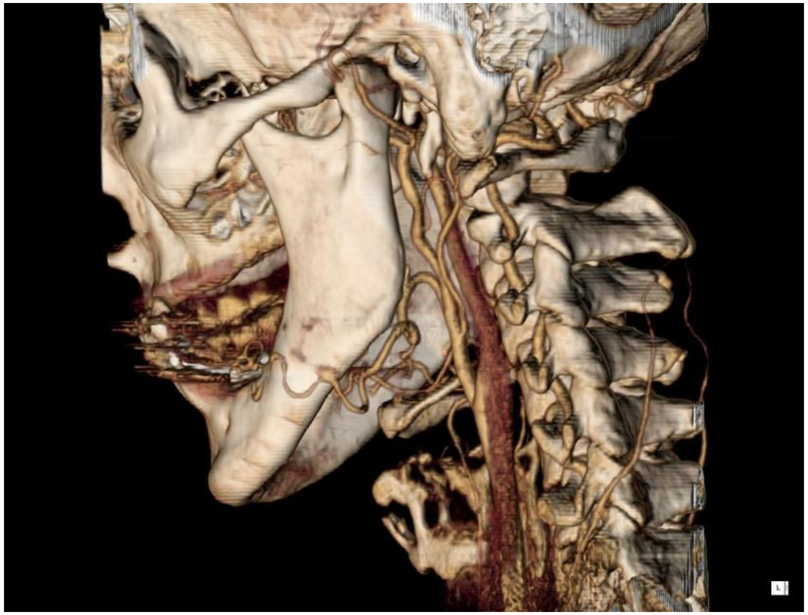
Visualisatie taxonomie

- Wetenschappelijke visualisatie
  - Scientific visualization (“scivis”)
  - Data with a natural geometric structure
    - Measurements, results from calculations or simulations
- Informatie visualisatie
  - Information visualization (“infovis”)
  - Data with an abstract structure
    - Relations, data structures

Wetenschappelijke visualisatie: “SciVis”

Numerieke gegevens afkomstig van metingen of van computer simulaties:

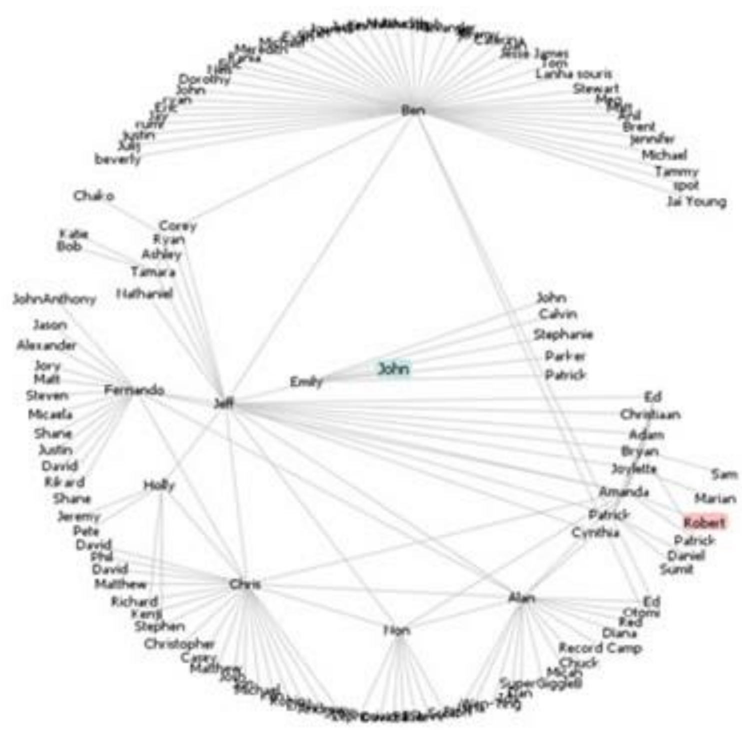
- Wetenschap & techniek
- Uniform: numerieke gegevens van fysische eigenschappen
- Duidelijk referentiekader
- Doel: verkrijgen van inzicht



Informatie visualisatie: "InfoVis"

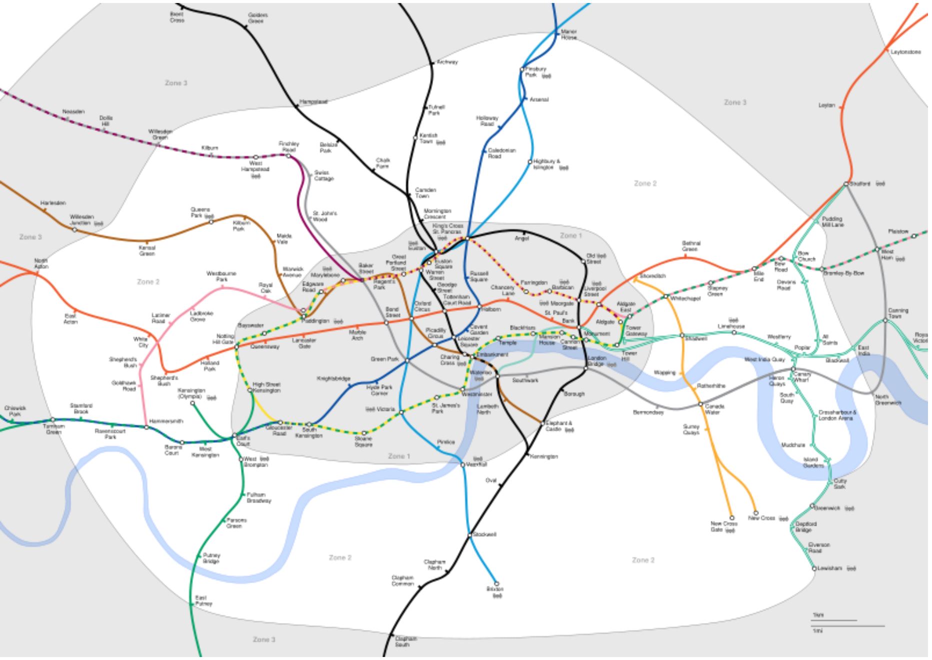
Gegevens en informatie afkomstig van databases, netwerken, etc.

- Grote diversiteit
- Numeriek, teksten, beelden, relaties, multimedia, ...
- Discreet, non-uniform
- Geen duidelijk referentiekader
- Doel: verkrijgen van inzicht

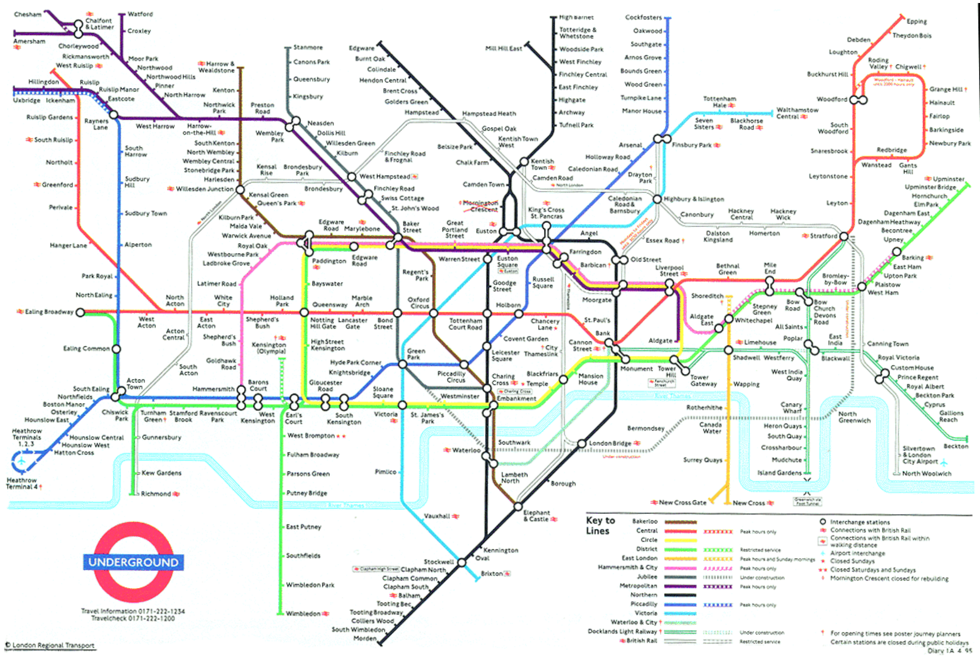


Information visualization

London underground

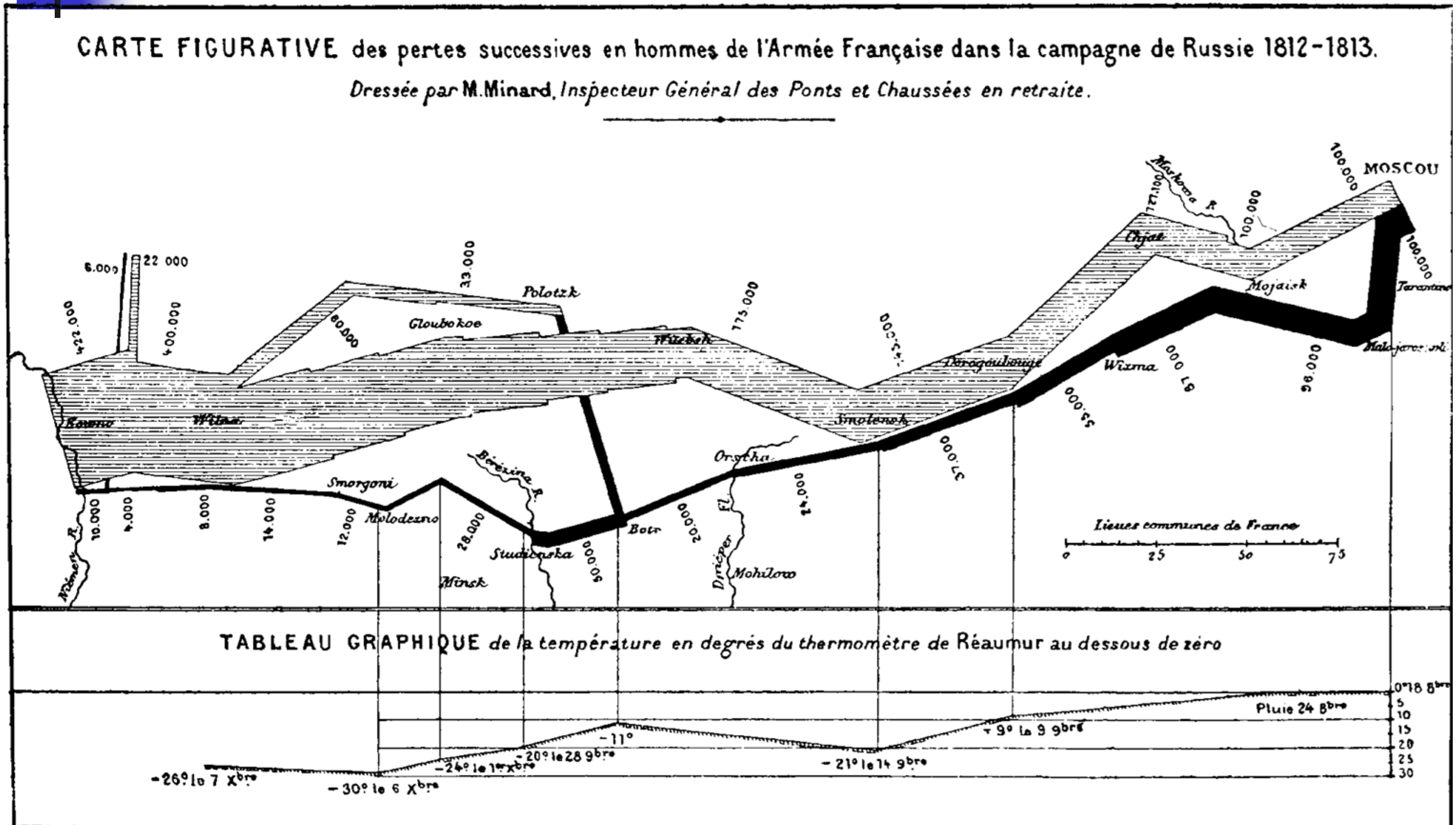


Topographical representation of the London underground.

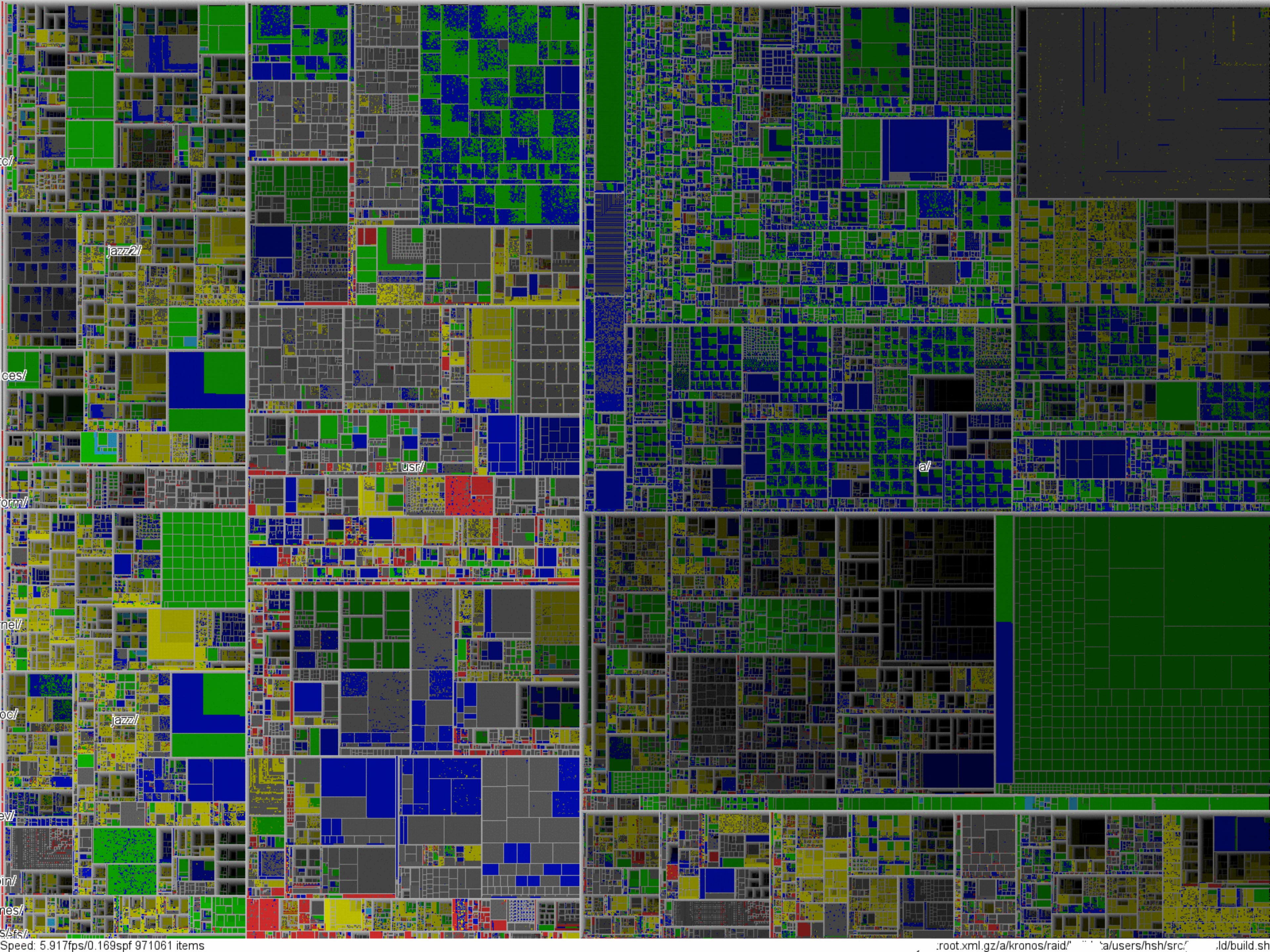


Schematic representation of the London underground, invented by Harry Beck in 1931.

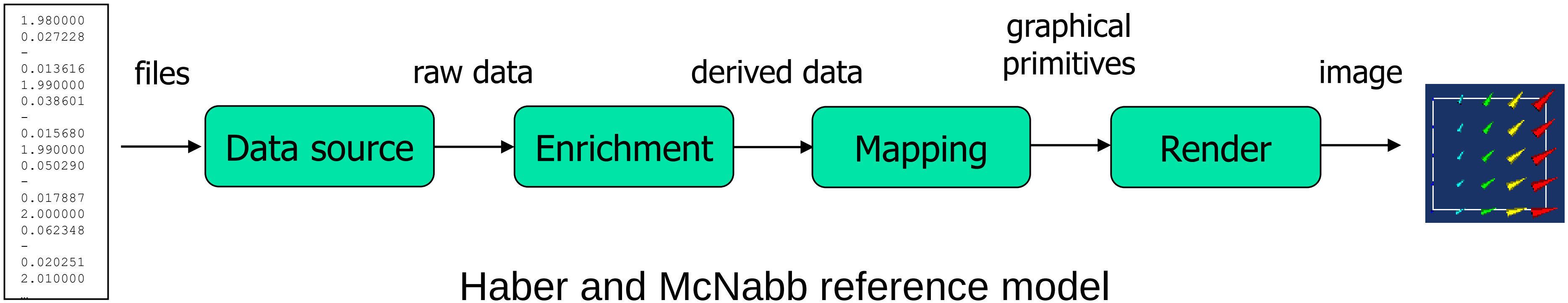
Information visualization



Xbre = December   9bre = November   8bre = October


tc/
jazz2/
ces/
usr/
a/
orm/
nel/
oc/
jazz/
ev/
in/
nes/
Speed: 5.917fps/0.169spf 971061 items
:root.xml.gz/a/kronos/raid/
a/users/hsh/src/
ld/build.sh

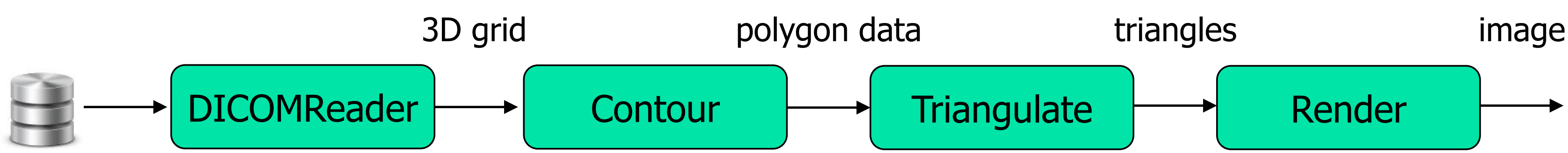
The visualization pipeline

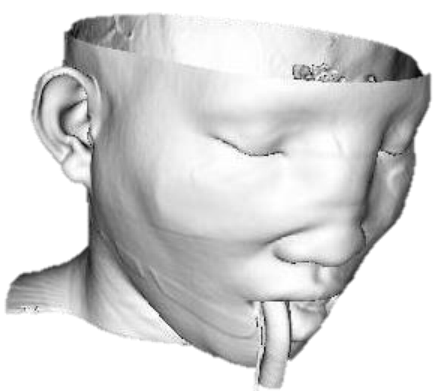


Haber and McNabb reference model

Haber, R.B. and McNabb, D.A. *Visualization idioms: A conceptual model for scientific visualization systems*. In *Visualization in Scientific Computing*, pages 74–93. IEEE Computer Society Press. 1990

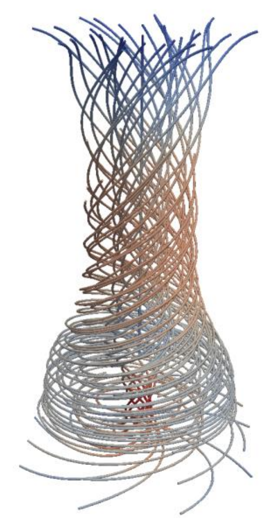
Example: extracting a contour from medical data



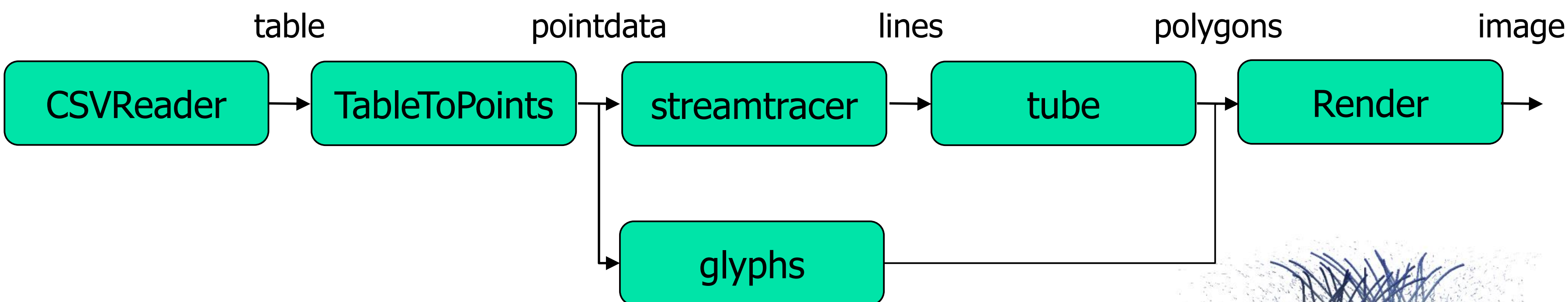


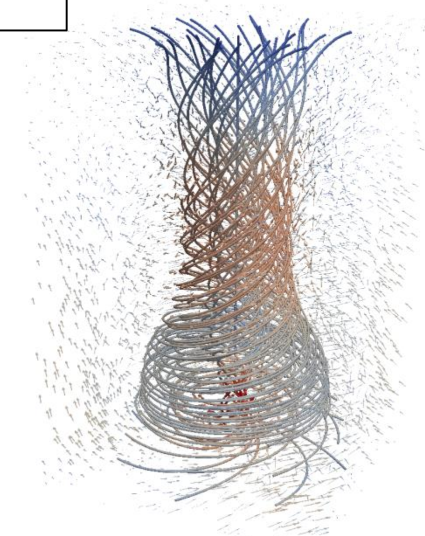
Example: visualizing flow data with streamlines

CSVReader $\xrightarrow{\text{table}}$ TableToPoints $\xrightarrow{\text{pointdata}}$ streamtracer $\xrightarrow{\text{lines}}$ tube $\xrightarrow{\text{polygons}}$ Render $\xrightarrow{\text{image}}$

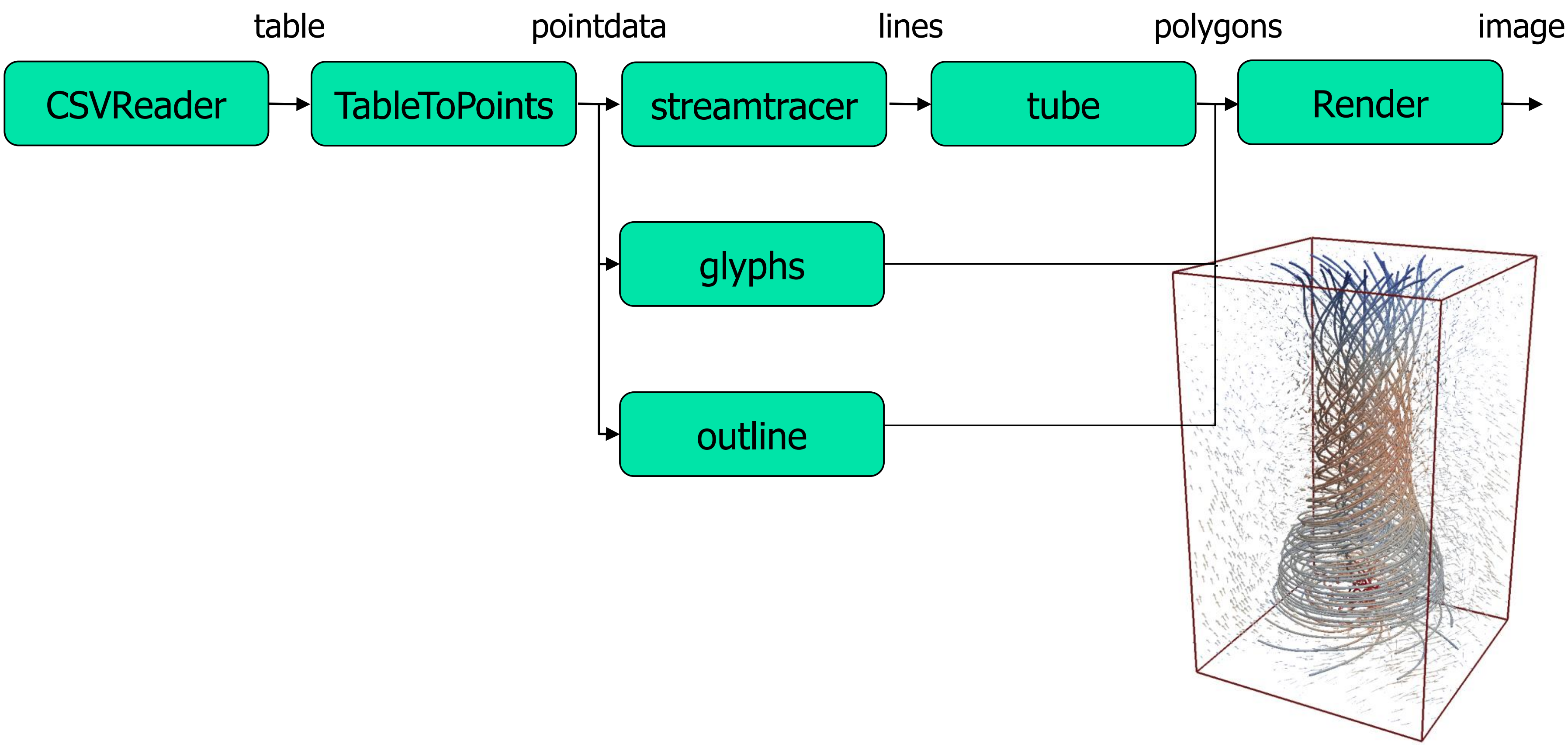


Example:
visualizing flow data with streamlines





Example:
visualizing flow data with streamlines



Pipeline creation

- Filters are connected together to form a “visualization pipeline” (a.k.a. “dataflow network”)
- The input port of a filter may only be connected to the output port(s) of (an)other filter(s) if the port types are “similar"

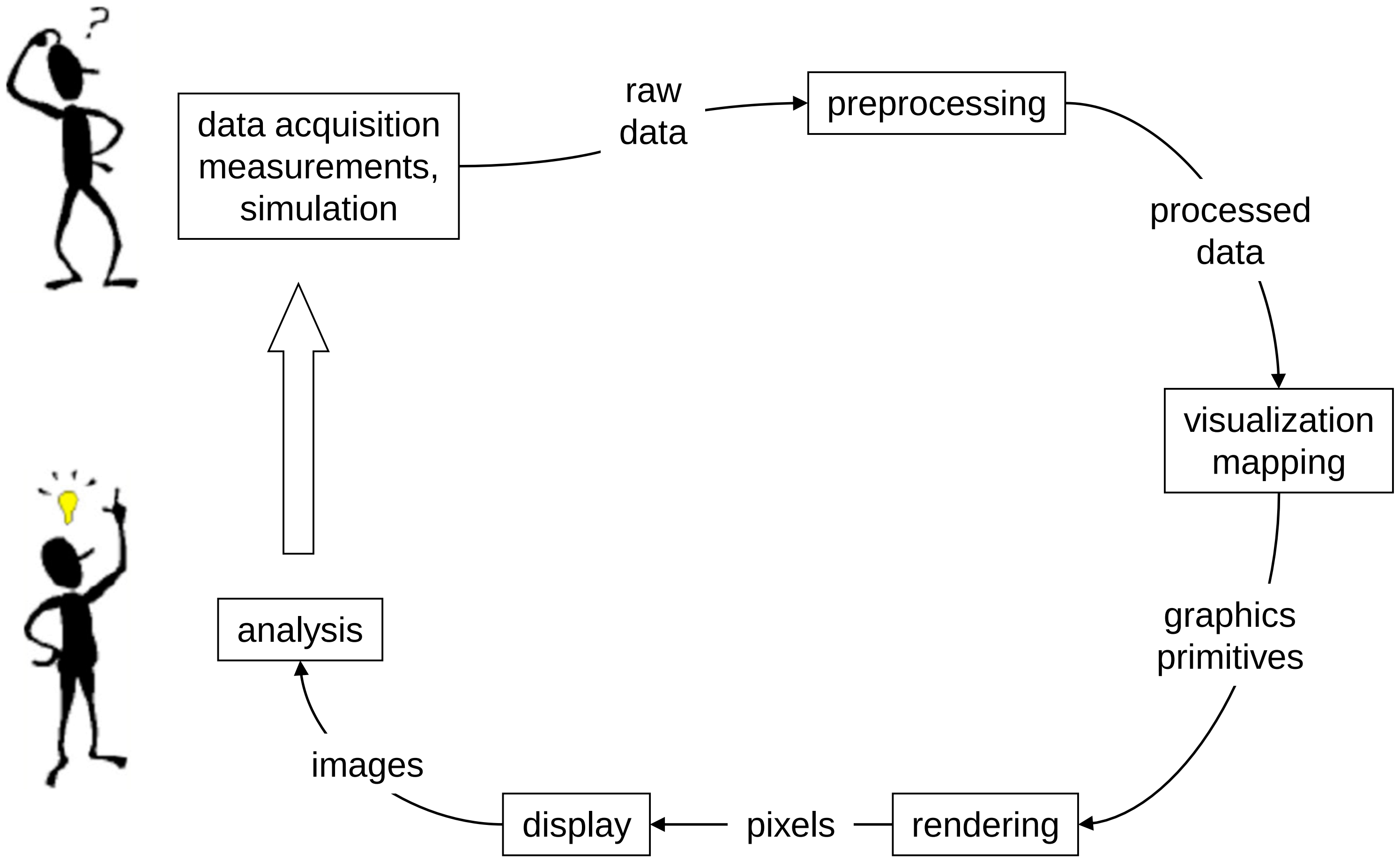
Pipeline behaviour

- Filters in a pipeline only execute when necessary
  - When data at the input has changed, or a parameter
- Data flows downstream, update checks go upstream

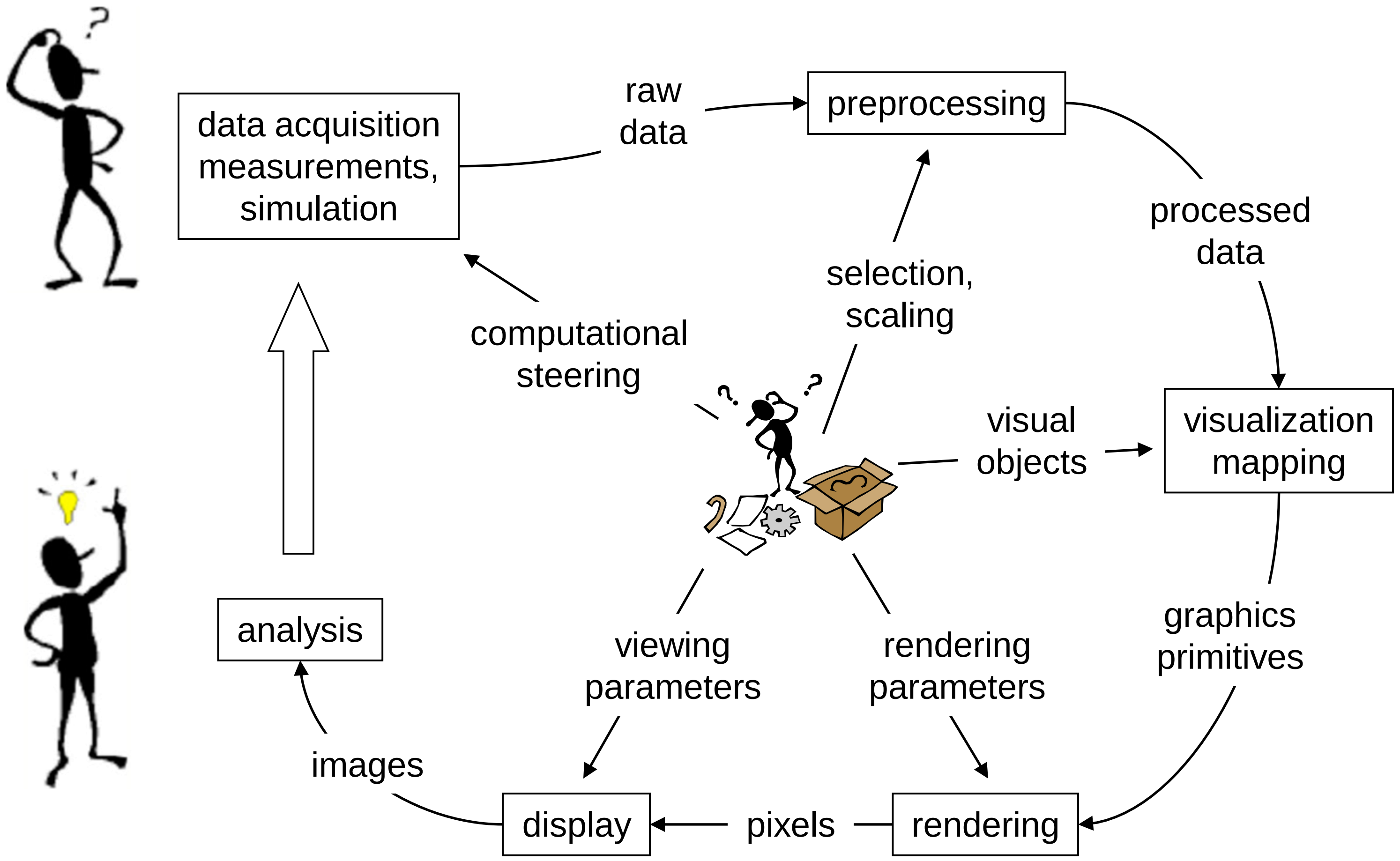
Wetenschappelijke visualizatie

Achtergrond, algoritmen en toepassingen

Visualization cycle



Interaction



Interactie

Als je niet alle gegevens tegelijkertijd kan laten zien:

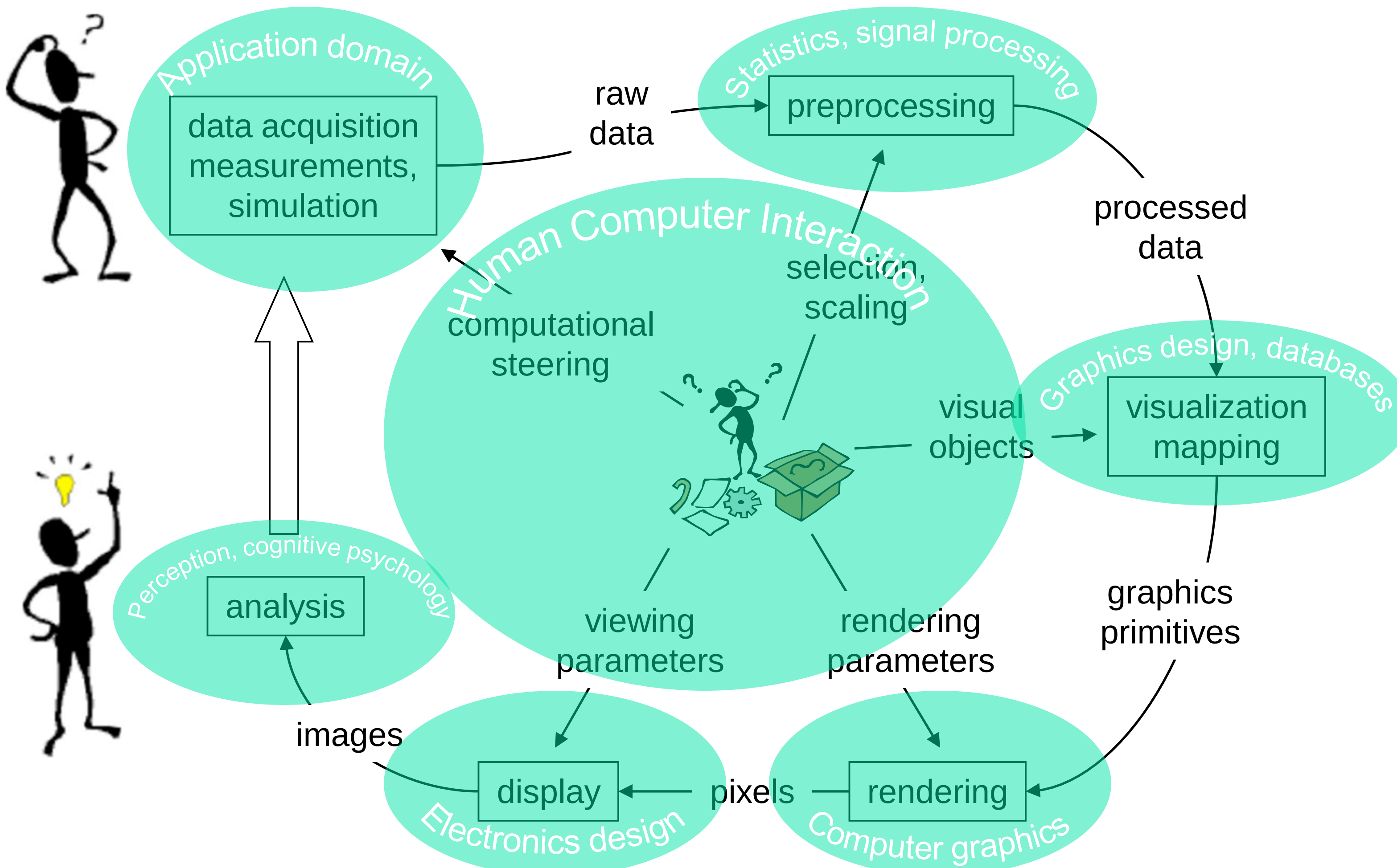
- Gebruik interactie
- Laat de gebruiker door de data navigeren

Schneiderman's mantra:

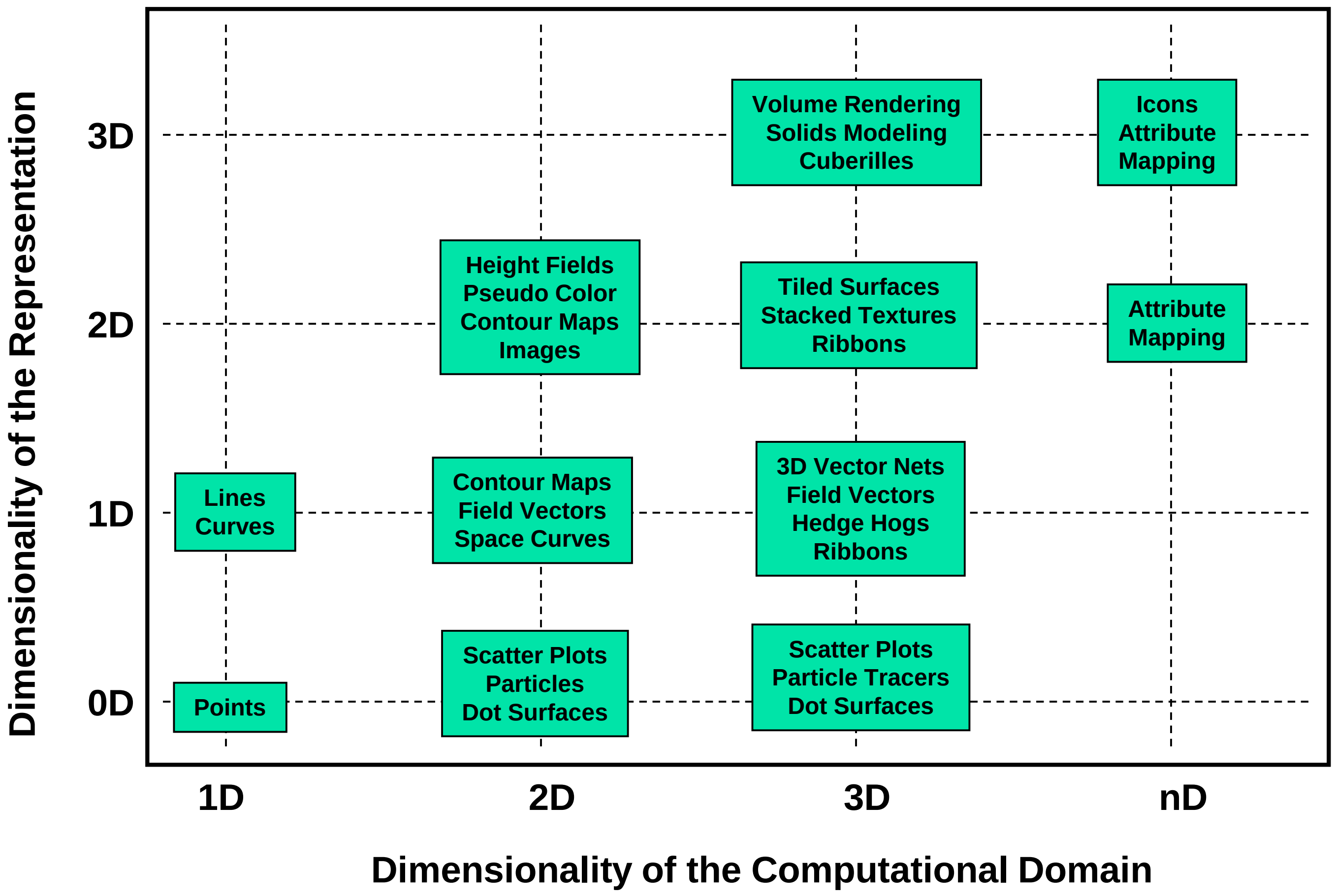
- "Overview first, then
- zoom & filter, then
- details on demand"

Ben Schneiderman

Interdisciplinary



Visualization mapping space



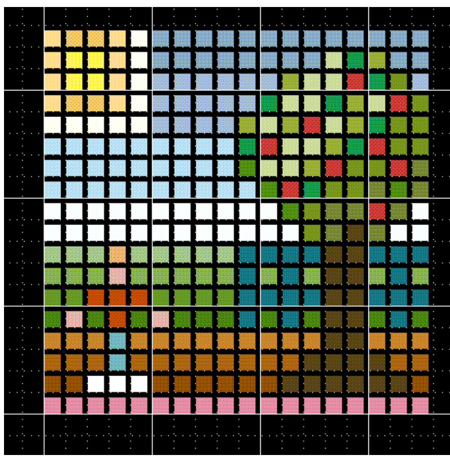
Overview of some algorithms

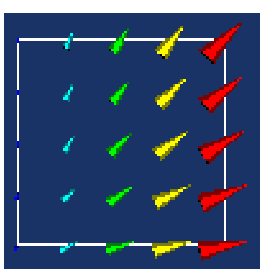
- Glyph visualization
- Contour visualization
- Vector field visualization

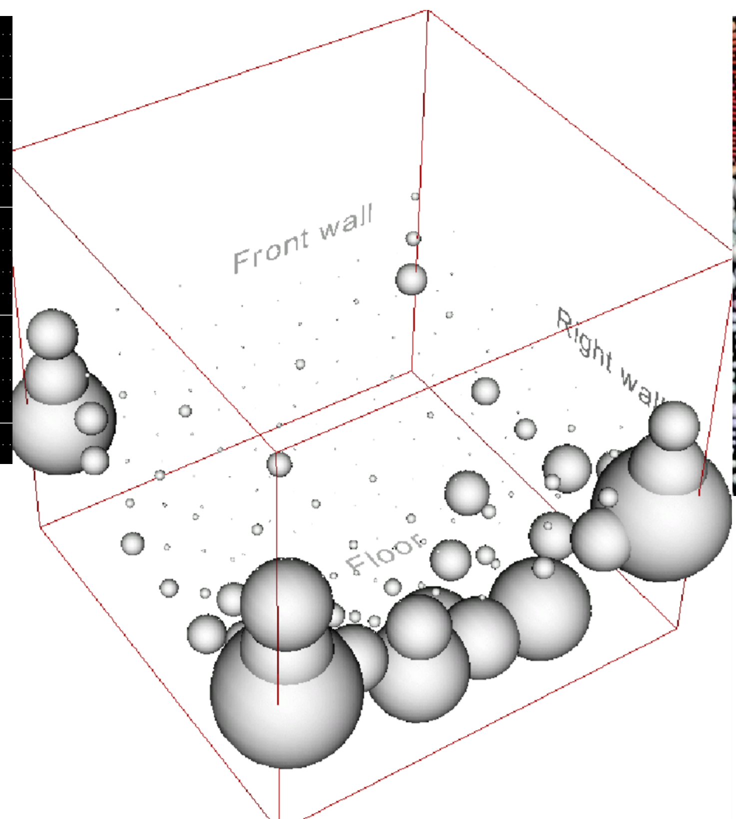
Glyph visualization

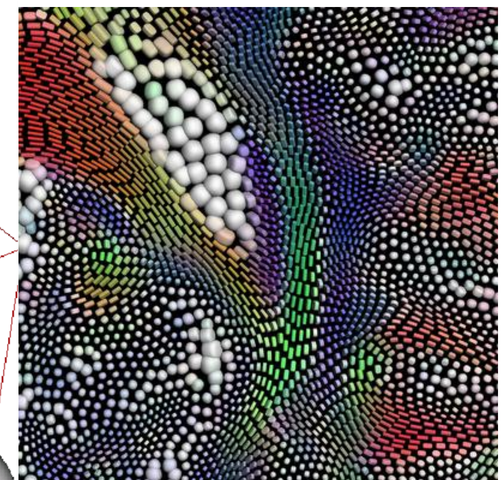
Glyph visualization

- Glyph: “object” transformed by input data
  - Position, orientation, scaling, colour, …





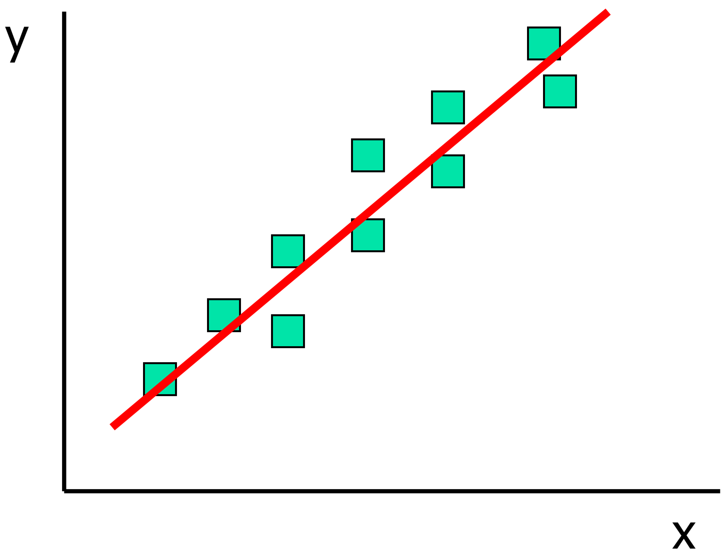




Glyph visualization

Examples:

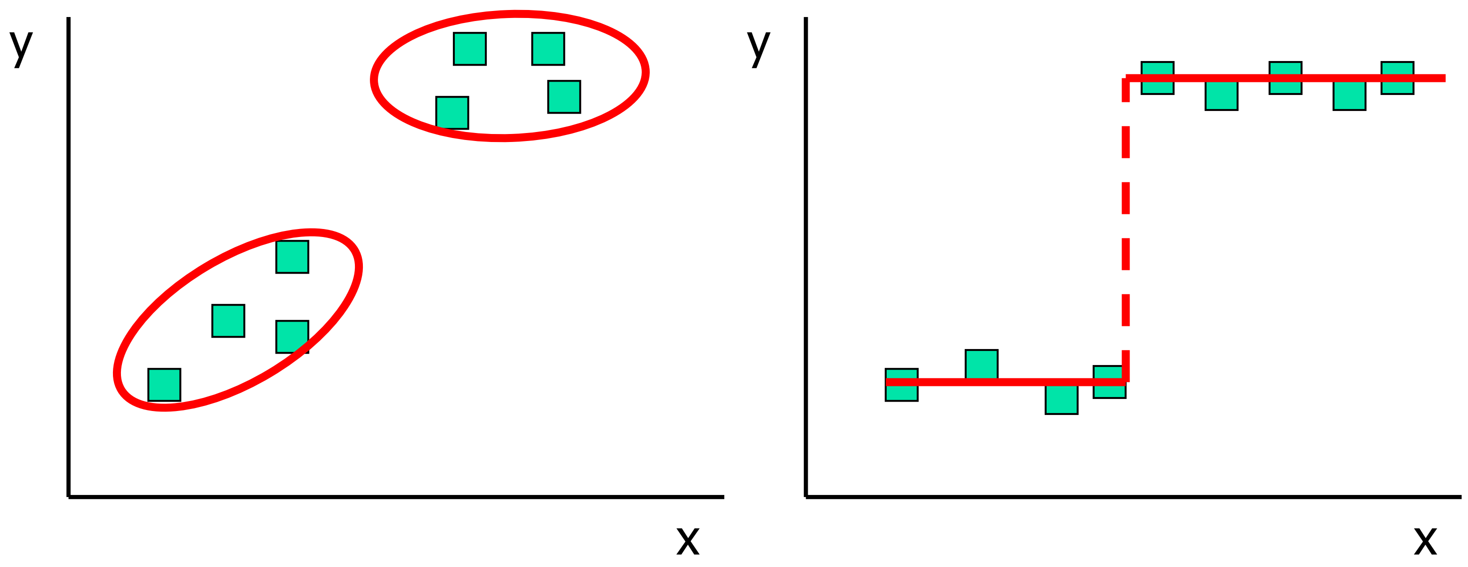
- Scatterplots (2 variables)
  - Detect correlations, trends, patterns, clusters



Glyph visualization

Examples:

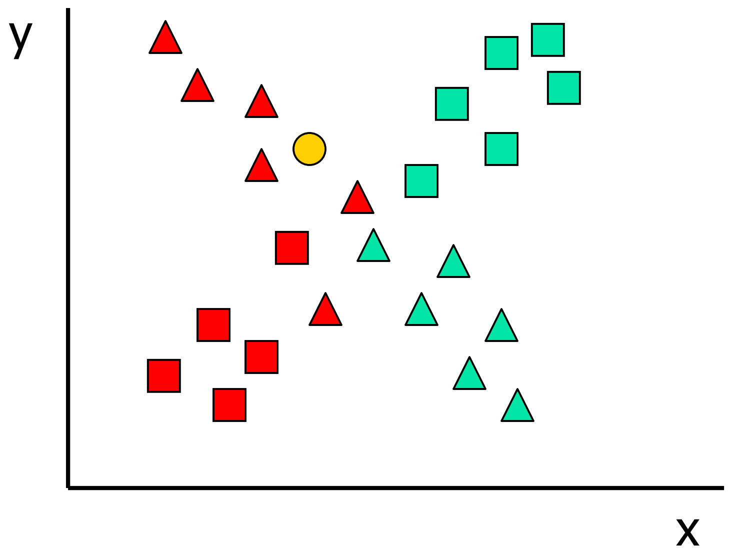
- Scatterplots (2 variables)
  - Detect correlations, trends, patterns, clusters



Glyph visualization

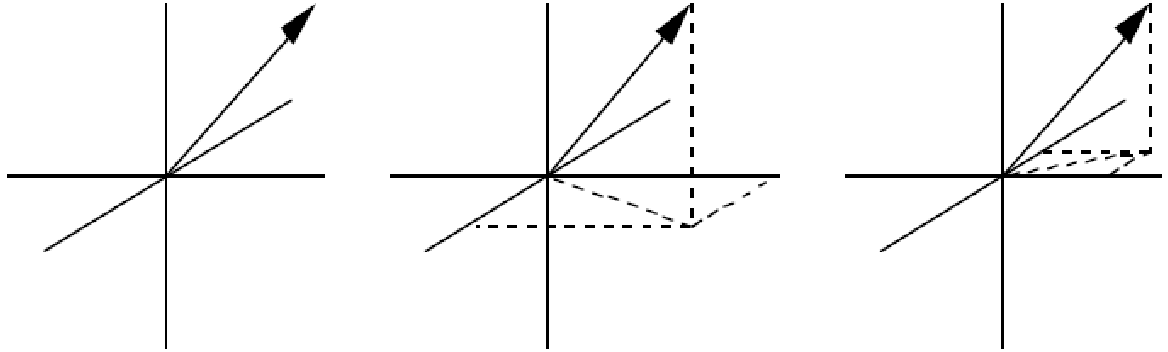
Examples:

- Scatterplots (*n* variables)
  - Detect correlations, trends, patterns, clusters



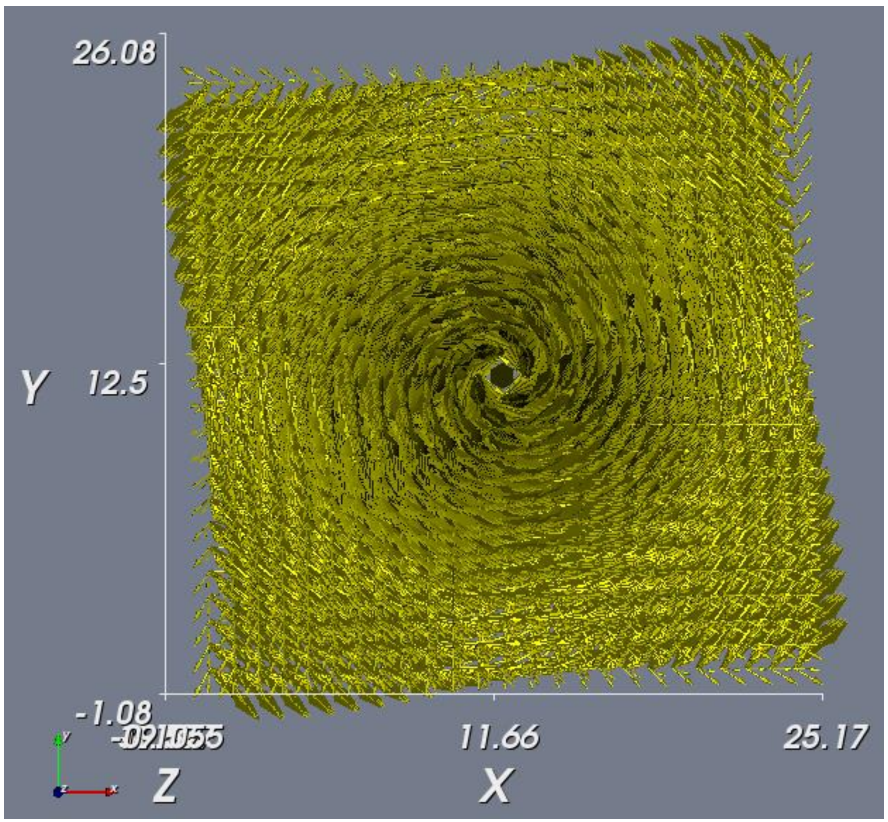
Direct techniques for 3D vector fields

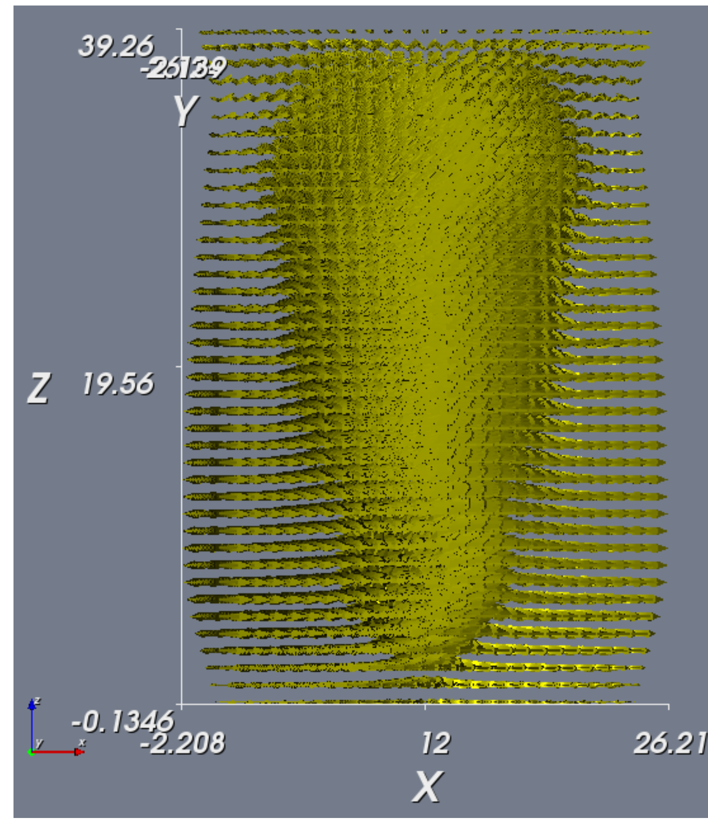
- Glyphs/hedgehogs
- Reasonable for 2D
- Less suitable for 3D:



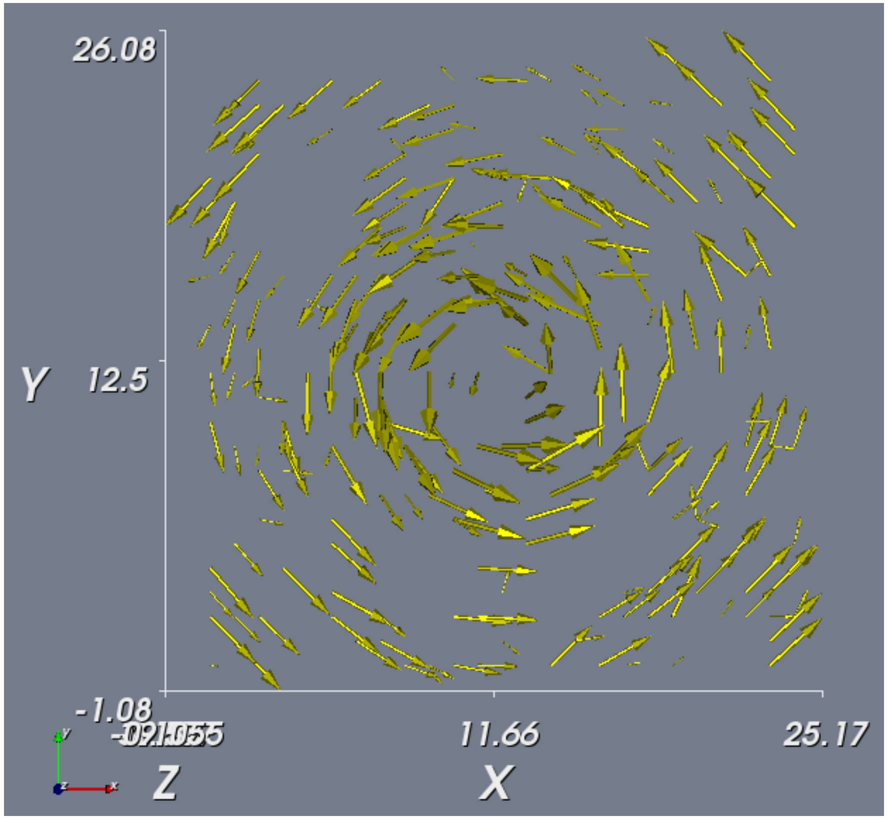
  - Clutter
  - No global overview

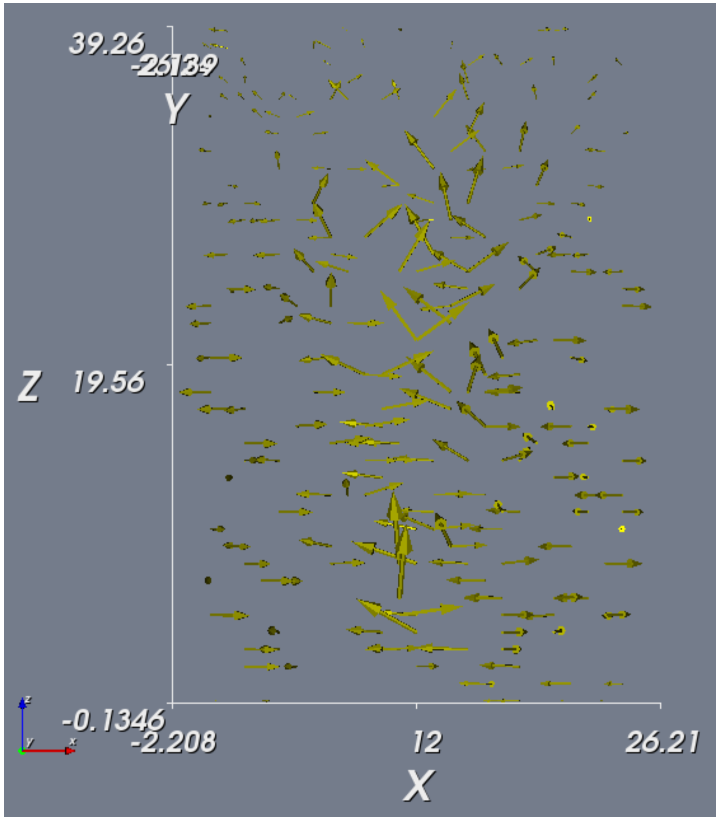
Clutter





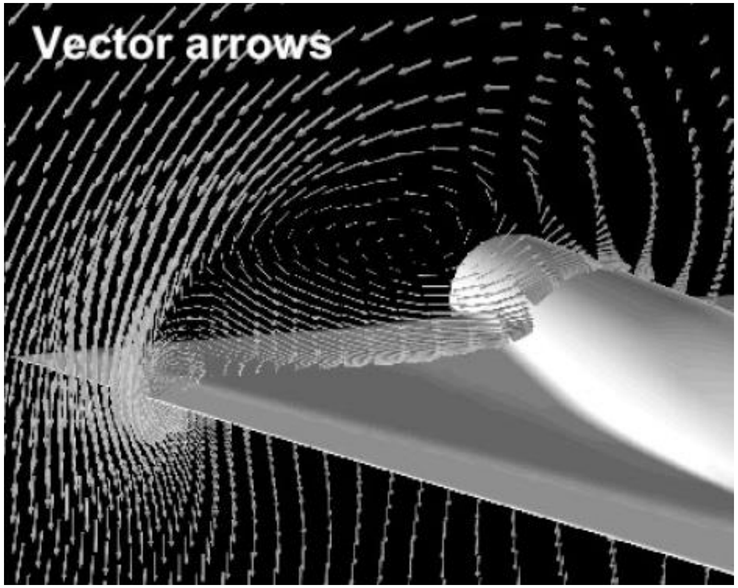
Random subsampling

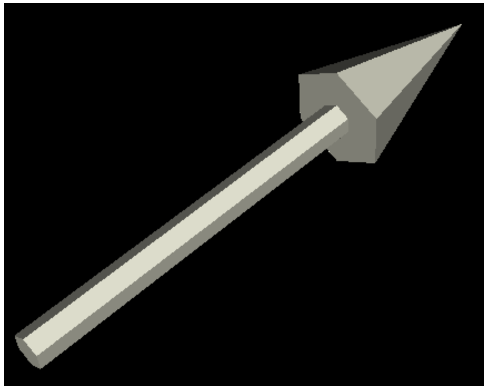




3D glyph plots

- 3D arrow plots don’t really work, but everybody does it!
- Possible improvements:
  - Glyphs on a constrained surface (e.g. plane)
  - Add depth cues (3D glyphs, shading, shadows)
  - Reduce number of glyphs





Iconification

Chernoff faces:

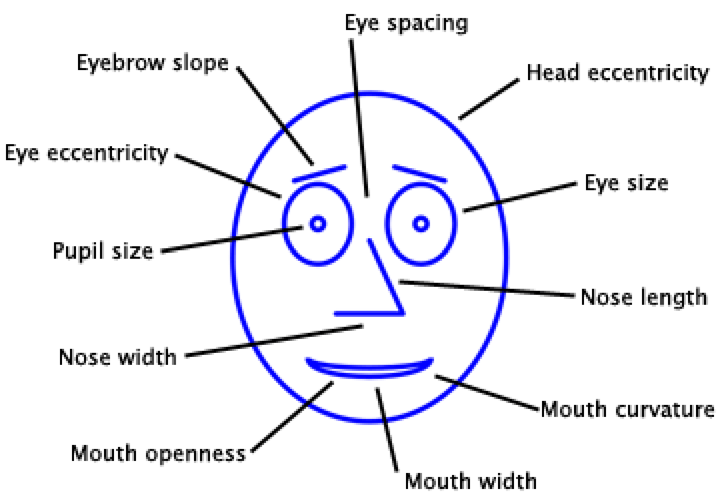
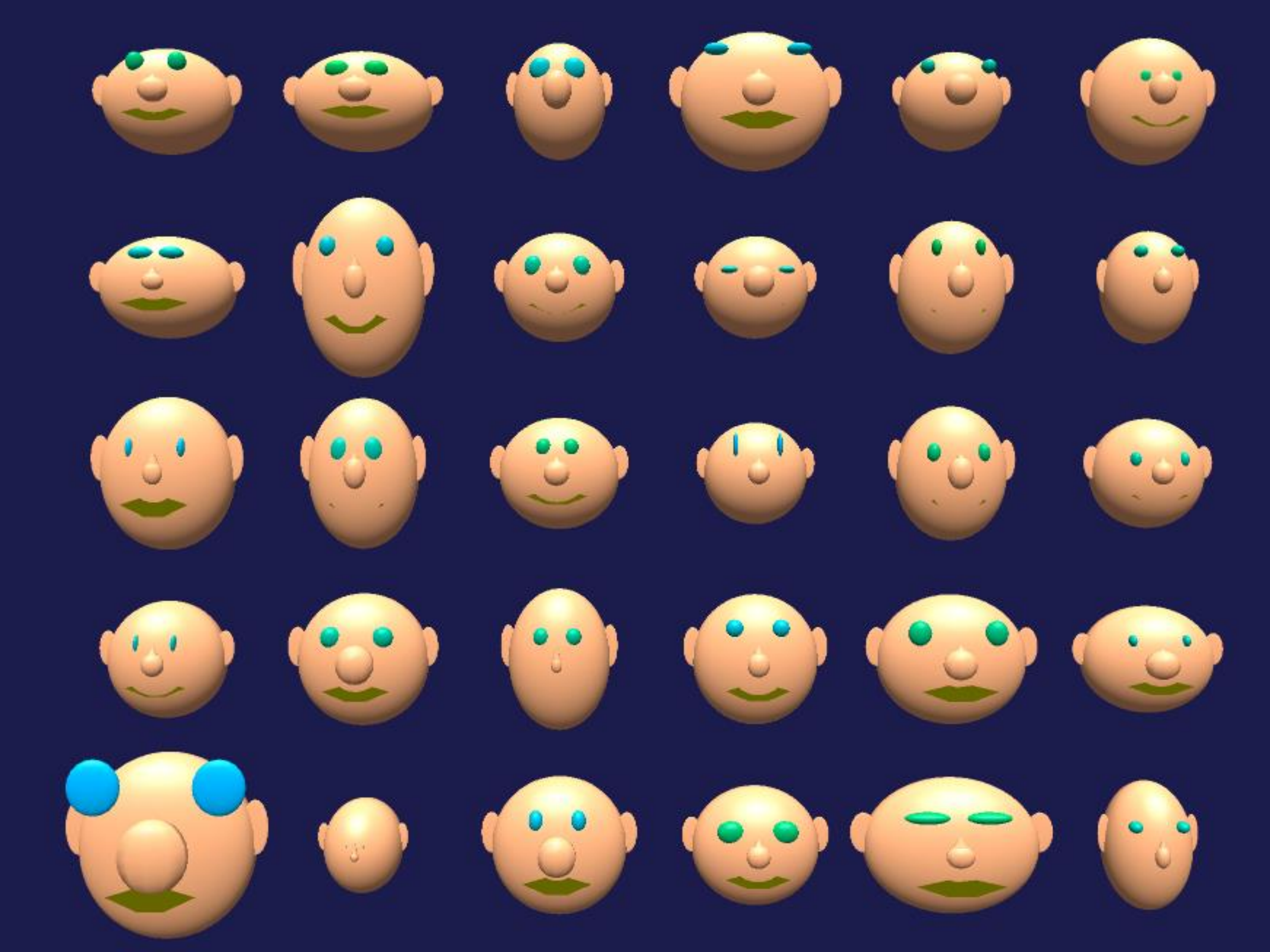
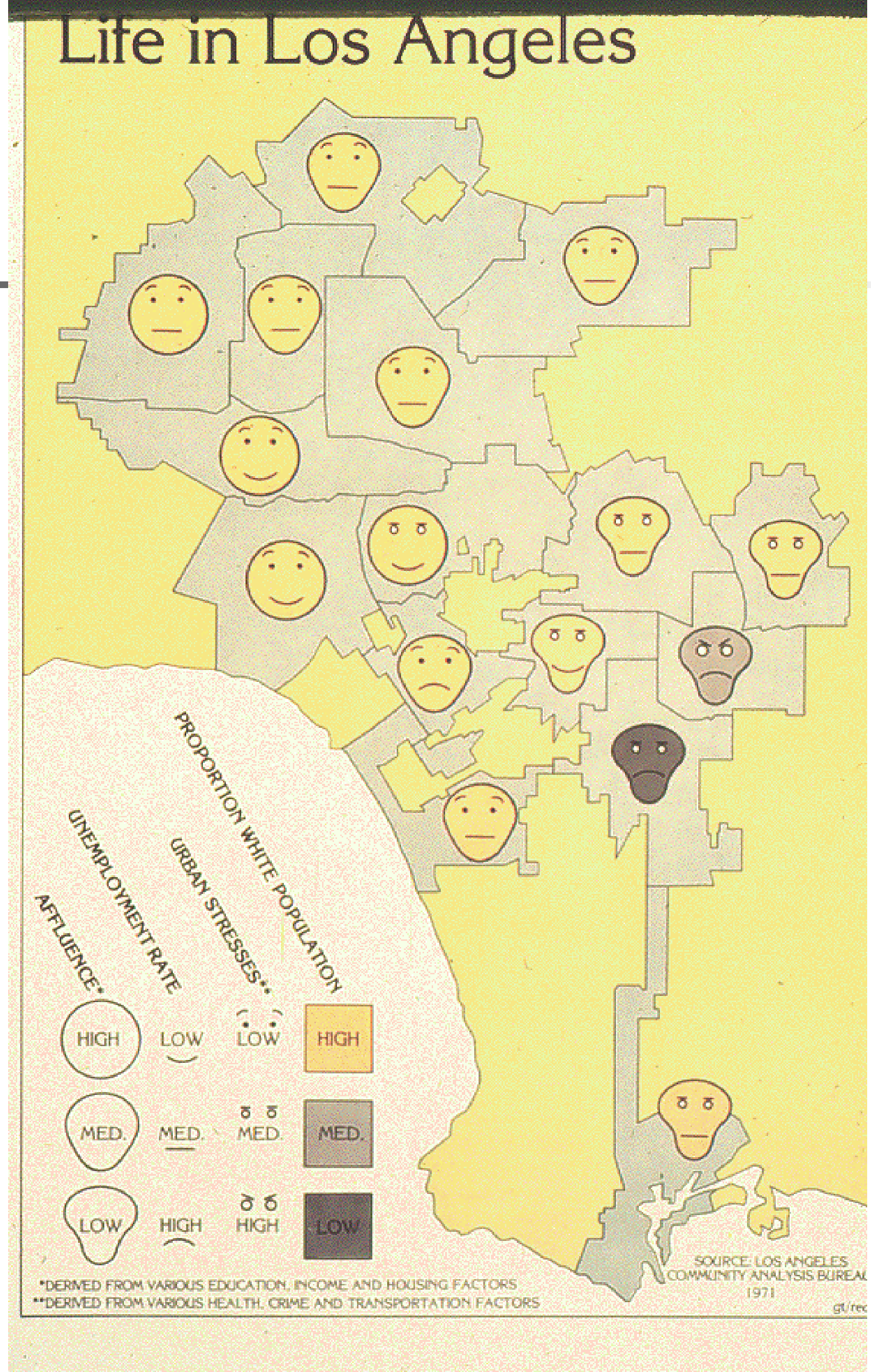


Table 1: Description of facial features of Chernoff face

Dimension	Facial Feature
1	Face width
2	Ear level
3	Half face height
4	Eccentricity of upper ellipse of face
5	Eccentricity of lower ellipse of face
6	Length of nose
7	Position of centre of mouth
8	Curvature of mouth
9	Length of mouth
10	Height of centre of eyes
11	Separation of eyes
12	Slant of eyes
13	Eccentricity of eyes
14	Half length of eye
15	Position of pupil
16	Height of eyebrow
17	Angle of brow
18	Length of brow
19	Radius of ear
20	Nose width





Life in Los Angeles
PROPORTION WHITE POPULATION
URBAN STRESSES**
UNEMPLOYMENT RATE
AFFLUENCE*
HIGH
LOW
LOW
HIGH
MED.
MED.
MED.
MED.
LOW
HIGH
HIGH
LOW
*DERIVED FROM VARIOUS EDUCATION, INCOME AND HOUSING FACTORS
**DERIVED FROM VARIOUS HEALTH, CRIME AND TRANSPORTATION FACTORS
SOURCE: LOS ANGELES
COMMUNITY ANALYSIS BUREAU
1971

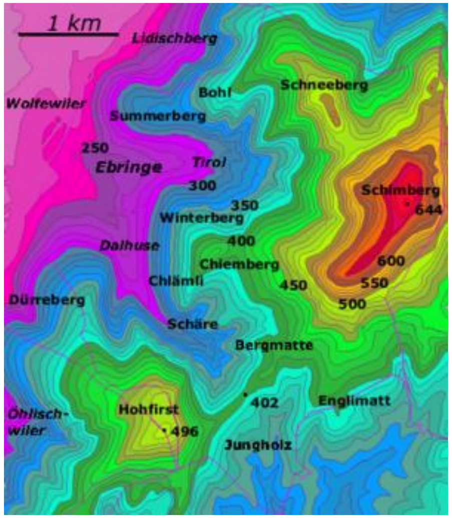
Contour visualization

Contour extraction

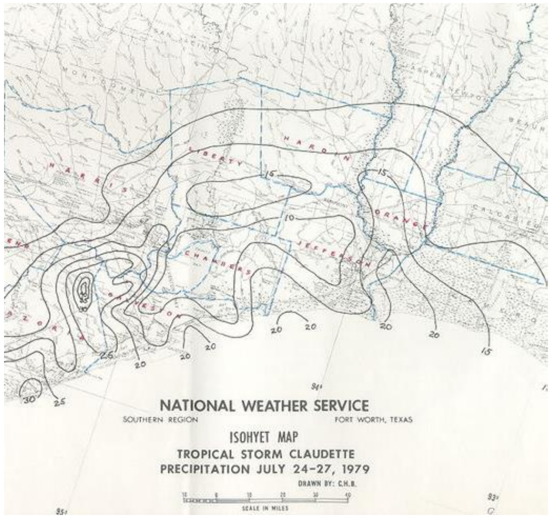
- 2D: contour lines
  - Marching squares
- 3D: isosurfaces
  - Marching cubes
  - Marching tetrahedra

Wat is een contour?

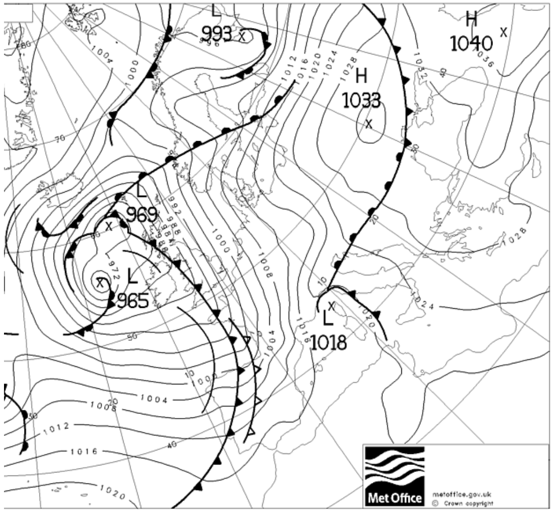
- Een contour beschrijft de omtrek van een gebied met dezelfde (of een vergelijkbare) waarde
  - een isobar kaart verbindt gebieden met dezelfde luchtduk
  - een hoogtekaart (isohypse) verbindt gebieden met dezelfde hoogte
  - een isotherm verbindt gebieden met dezelfde temperatuur



Isohypse (zelfde hoogte)

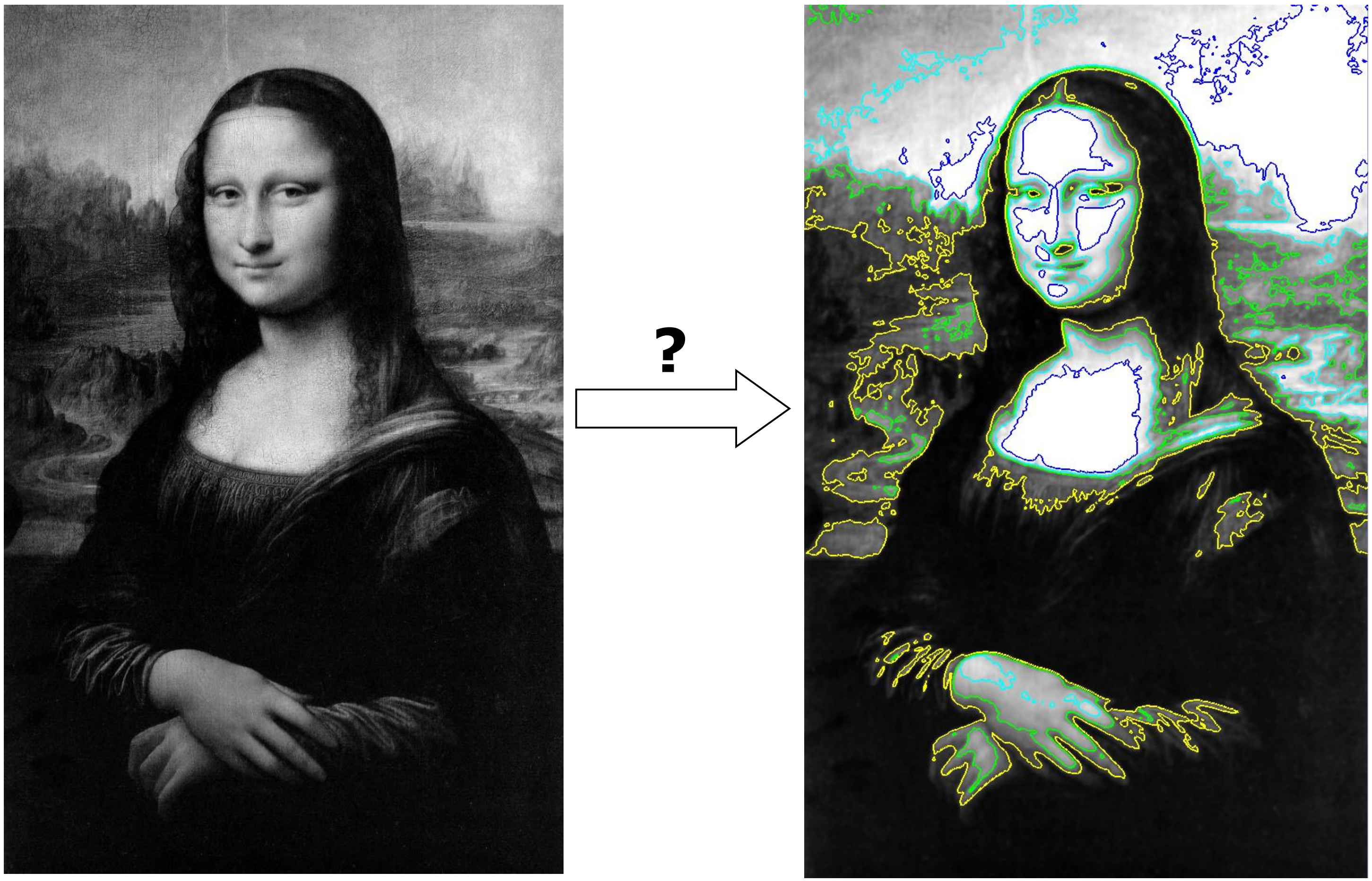


Isohyeet (zelfde hoeveelheid neerslag)

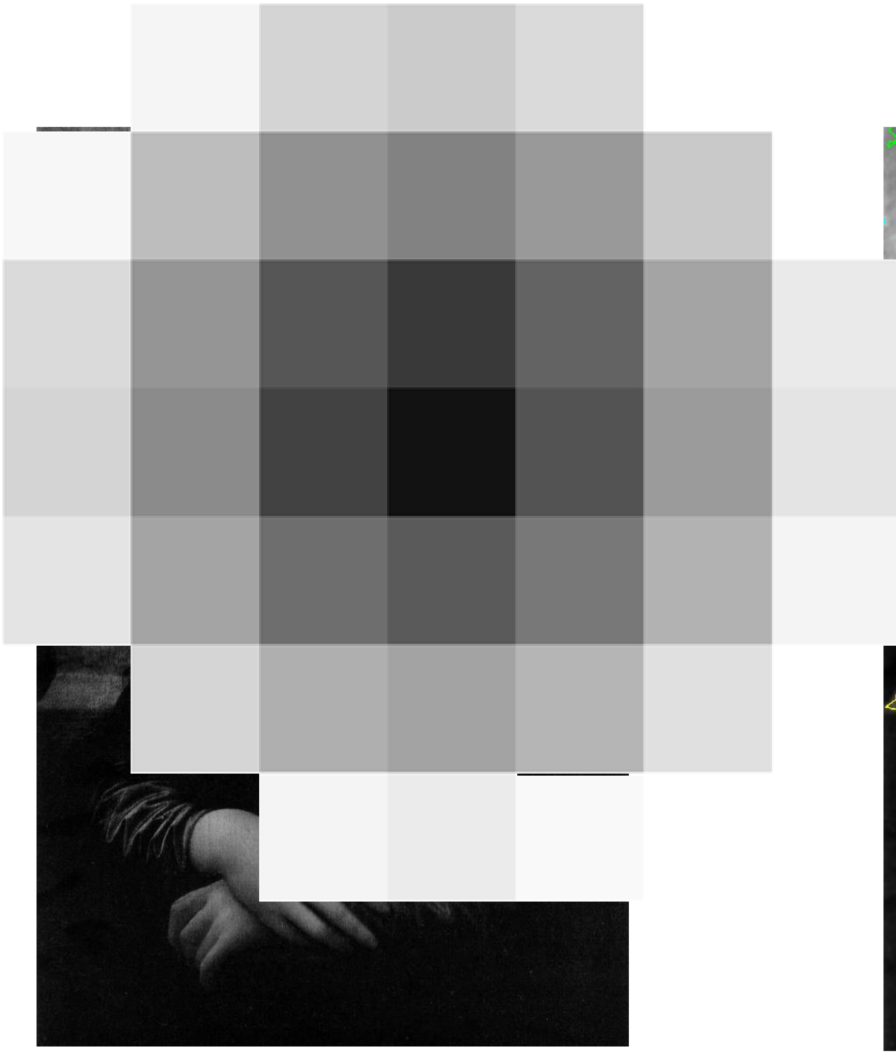


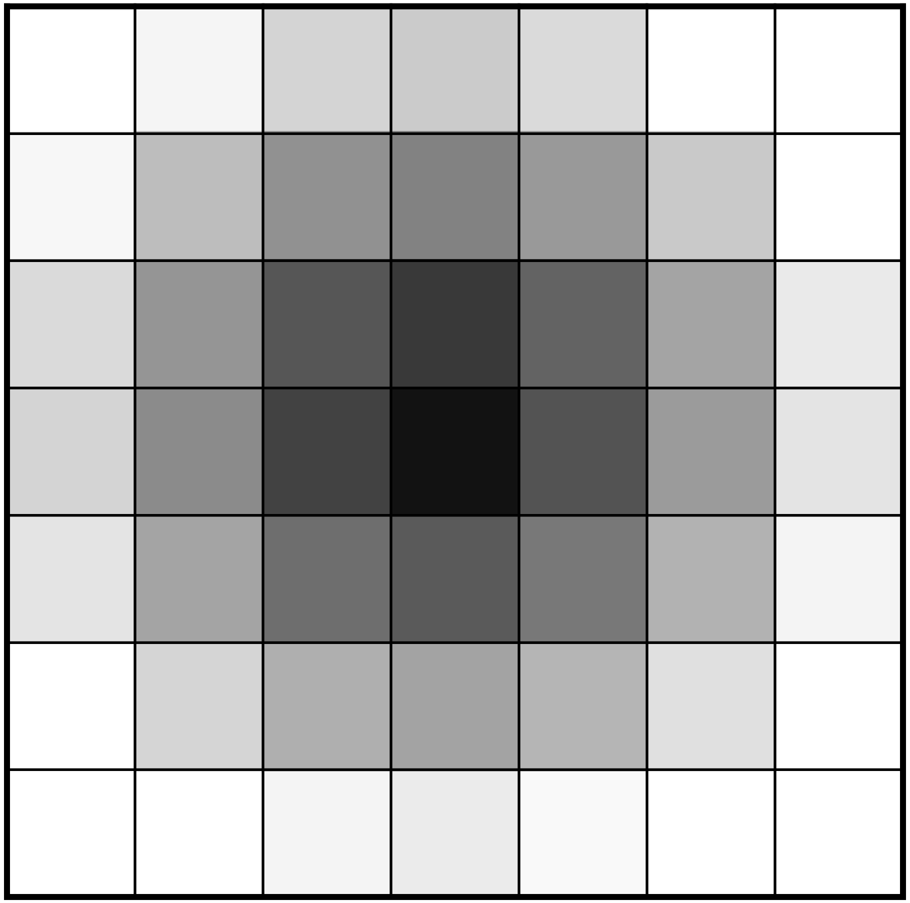
Isobar (zelfde luchtdruk)

Contour lijnen: hoe maak je die?



Plaatjes bestaan uit pixels





Elk pixel is een klein, vierkant gebiedje met een kleur.

Hier is de kleur een grijswaarde, een tint tussen zwart en wit.

0	4	18	21	15	0	0
3	27	46	52	43	22	0
15	44	71	83	65	38	8
18	48	79	100	72	42	11
11	38	61	69	56	32	4
0	17	33	38	31	13	0
0	0	4	8	2	0	0

Elke grijswaarde kunnen we ook weergeven door een getal.

Bijvoorbeeld:

- 0 : wit
- 50 : half zwart, half wit
- 100 : zwart

0	4	18	21	15	0	0
3	27	46	52	43	22	0
15	44	71	83	65	38	8
18	48	79	100	72	42	11
11	38	61	69	56	32	4
0	17	33	38	31	13	0
0	0	4	8	2	0	0

Nu kunnen we elk contour bepalen tussen de waarde 0 en 100.

Daarvoor bestaat een methode die eenvoudig in een computerprogramma is te stoppen: het “Marching Squares” algoritme.

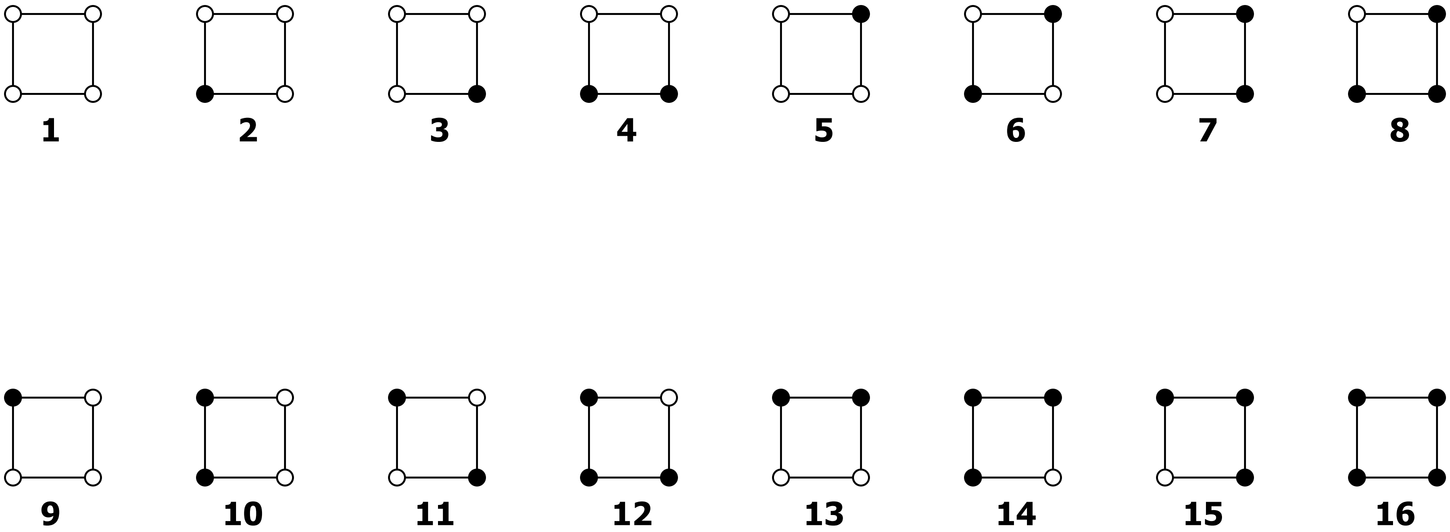
Het Marching Squares algoritme werkt als volgt:

0	4	18	21	15	0	0
3	27	46	52	43	22	0
15	44	71	83	65	38	8
18	48	79	100	72	42	11
11	38	61	69	56	32	4
0	17	33	38	31	13	0
0	0	4	8	2	0	0

1. Plaats een vierkant met de hoekpunten over de vier pixels linksboven in het plaatje
2. Bepaal van elk van de 4 hoekpunten in het vierkant of de waarde van het pixel boven (•) of onder (○) de contourwaarde zit

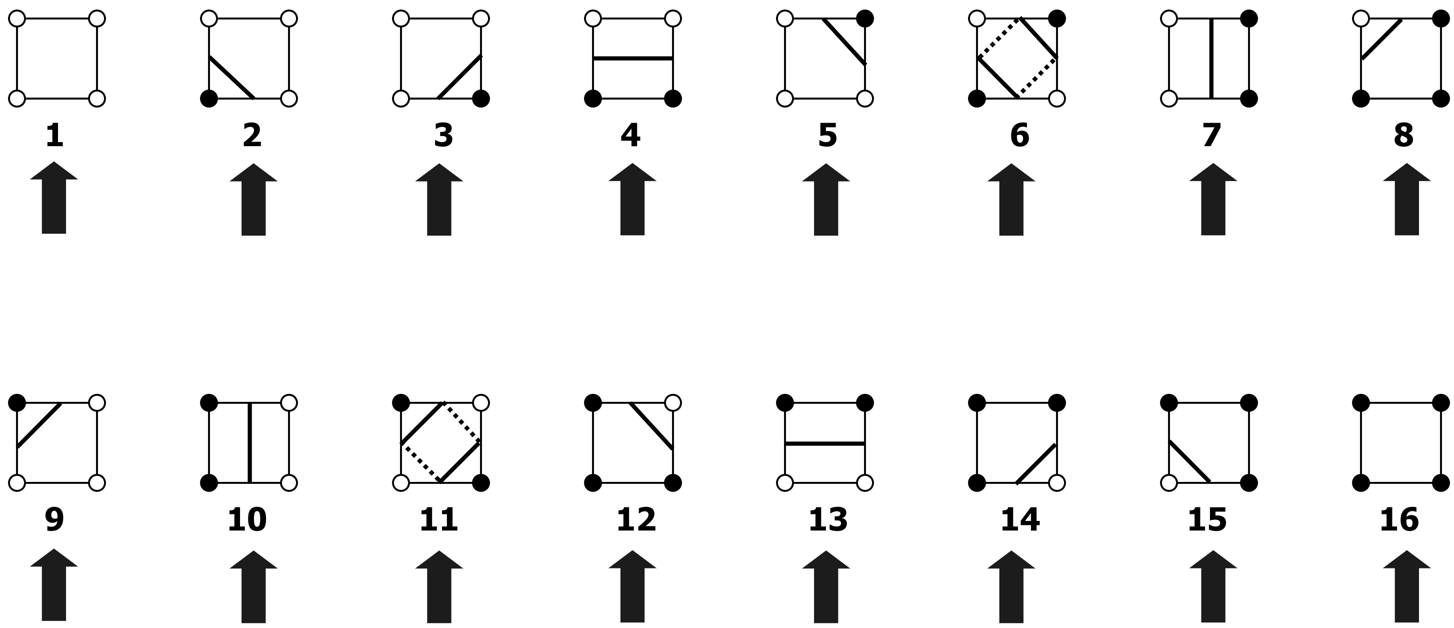
16 mogelijkheden

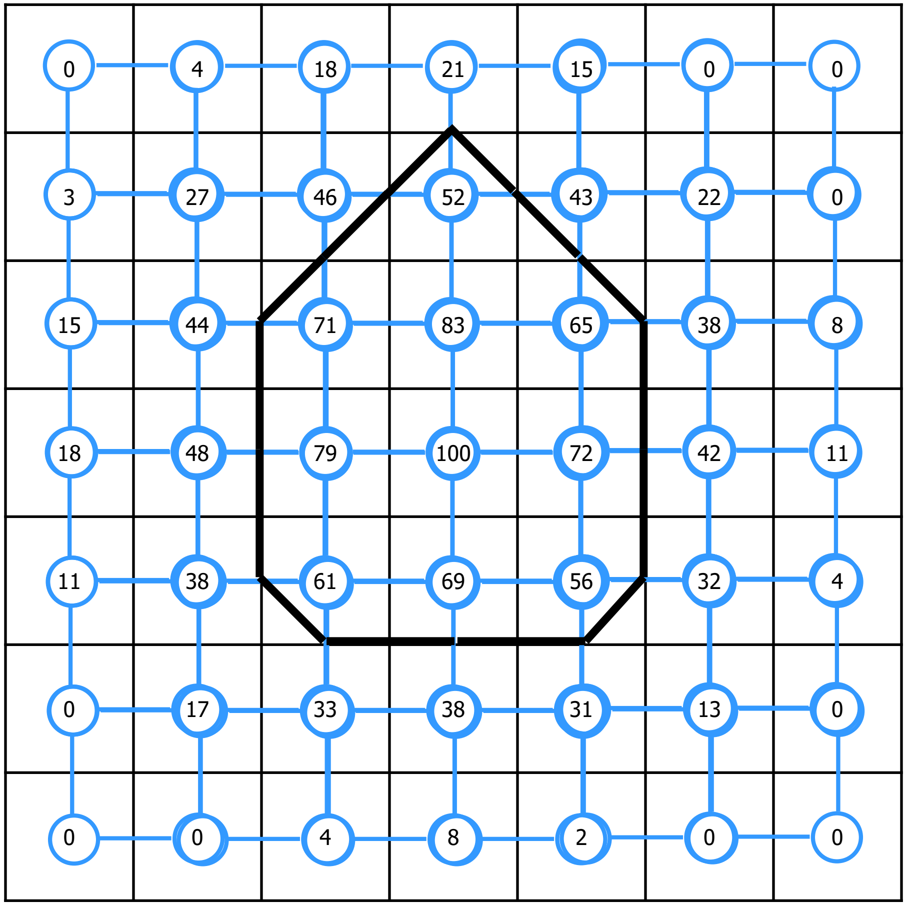
4 pixels, elk 2 mogelijkheden (erboven of eronder),
dus 2^4=16 mogelijkheden:



16 mogelijkheden

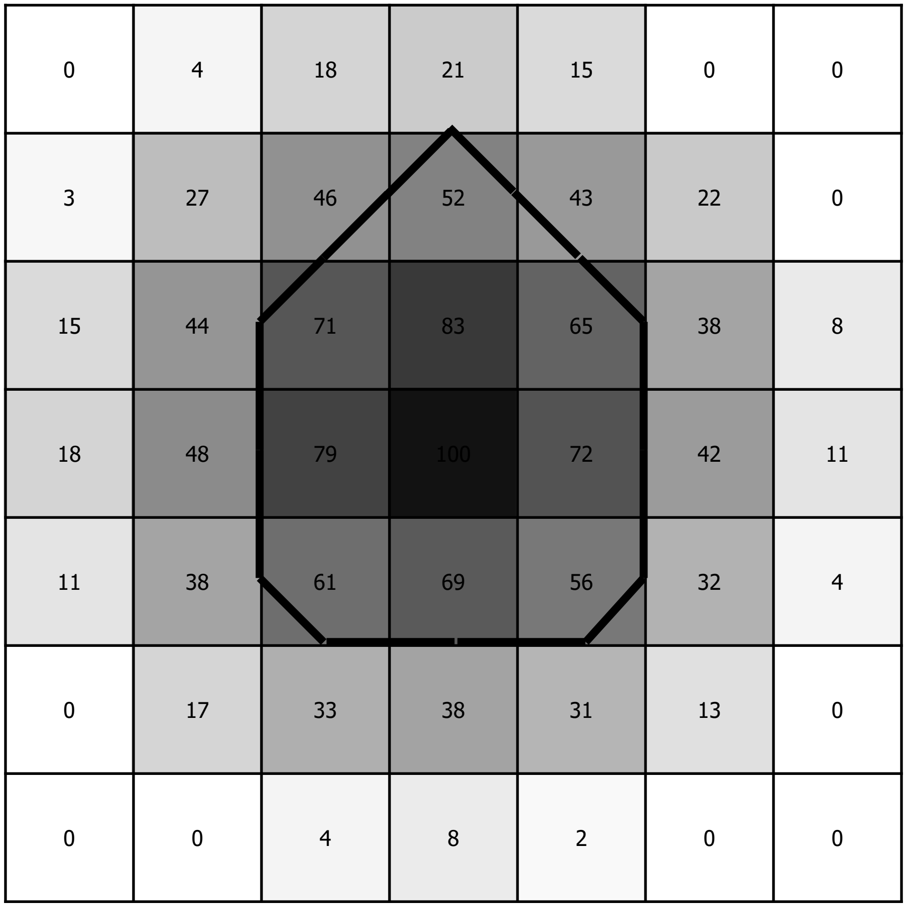
We kunnen heel makkelijk zien waar het contour de zijden van het vierkant doorsnijdt. Doe je mee?





In dit voorbeeld zoeken we het contour met de waarde 50.

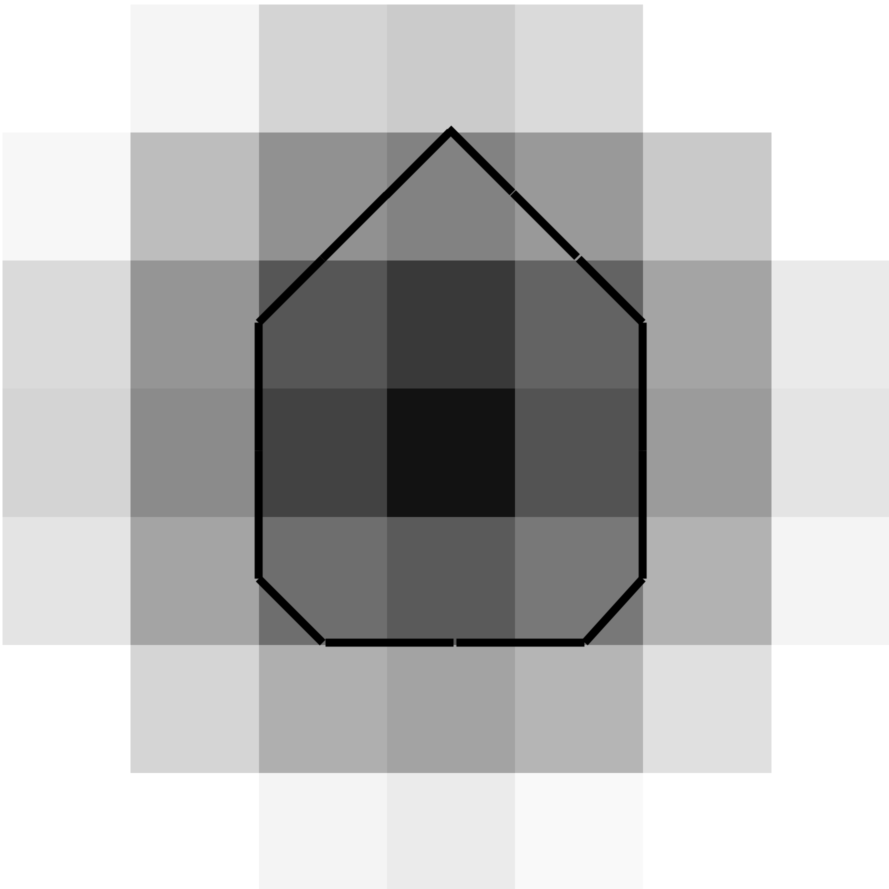
1. Plaats een vierkant met de hoekpunten over de vier pixels linksboven in het plaatje
2. Bepaal van elk van de 4 hoekpunten in het vierkant of de waarde van het pixel boven (•) of onder (◦) de contourwaarde zit
3. Zoek op in een tabel waar het contour het vierkant doorkruist
4. Sla de lijnstukjes op in een lijst en schuif het vierkant naar de volgende 4 pixels
5. Ga naar 2. tot we alle pixels hebben gehad
6. Teken alle lijnstukjes



Dit is het resultaat in het grijswaardenplaatje.

Voor het gemak hebben we de snijpunten van het contour met de vierkanten telkens precies door het midden gekozen.

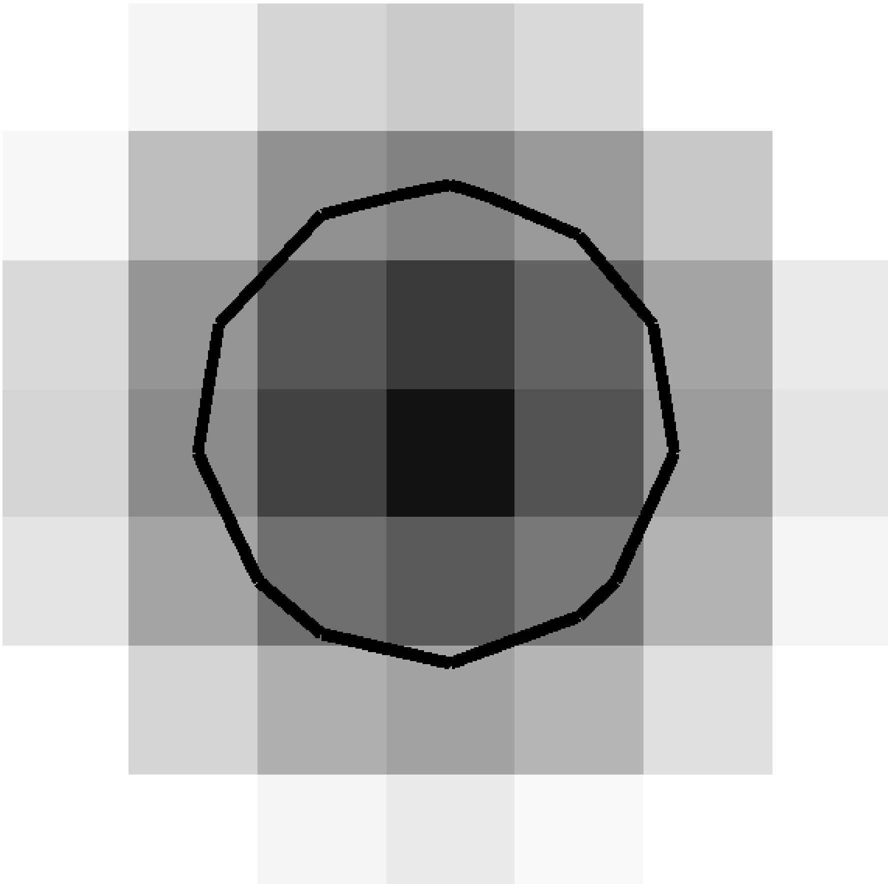
Dat kan ook beter...



Je kunt ook uitrekenen of het snijpunt dichter bij het ene punt zit dan bij het andere.

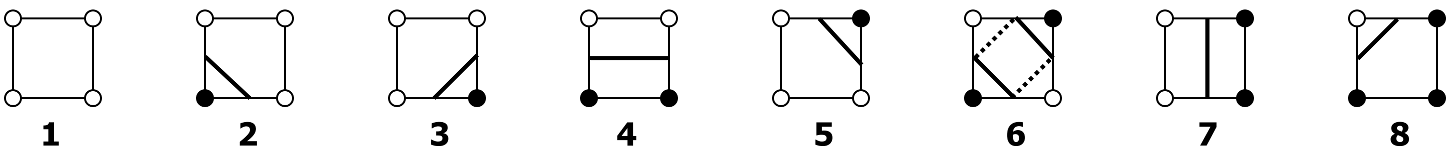
Dat noemen we “lineaire interpolatie”.

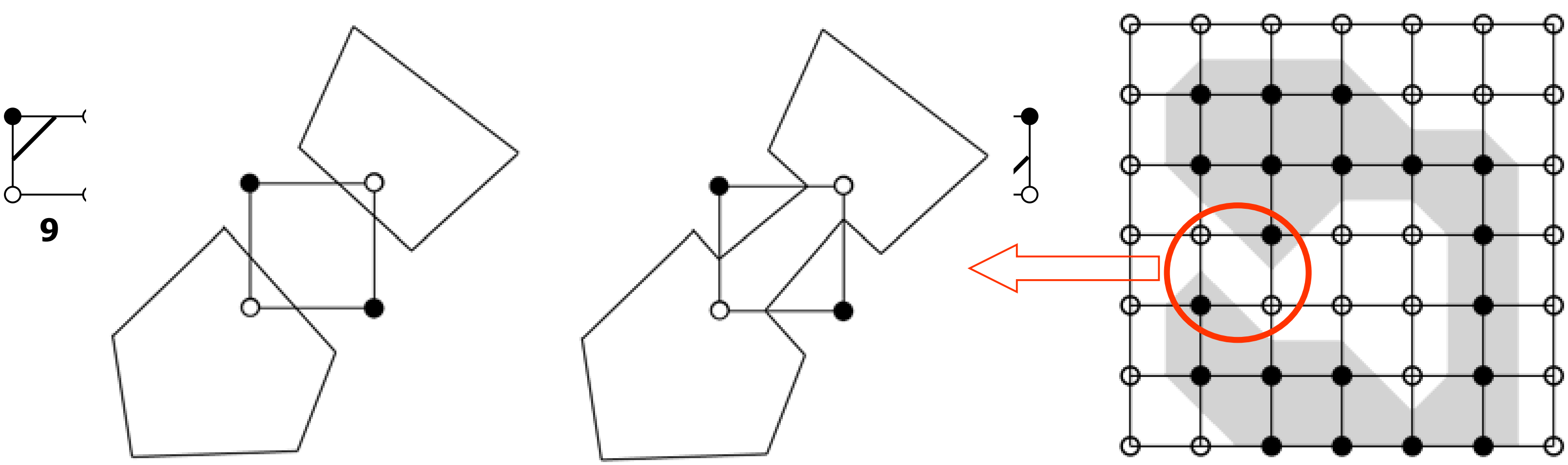
Dan ziet het contour er zo uit...



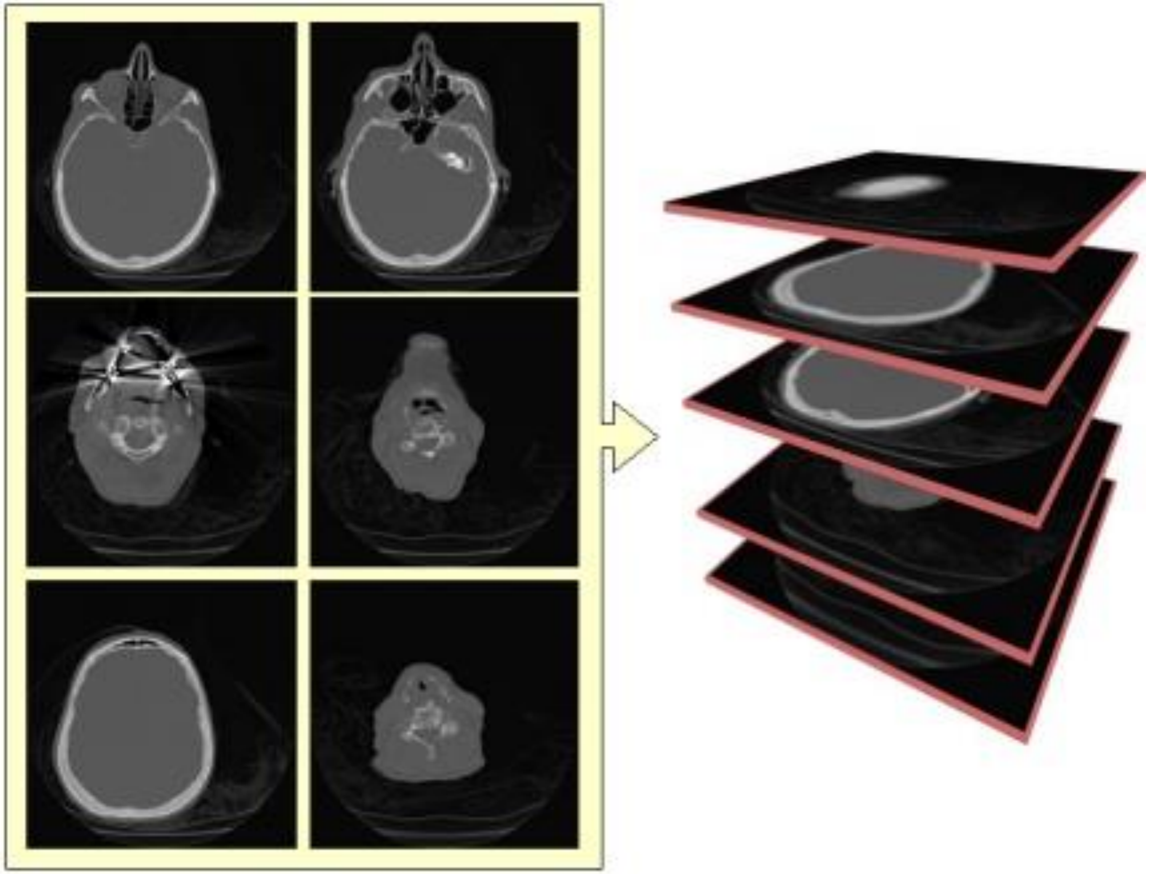
Dat is al een stuk beter, niet?

Twee “moeilijke” gevallen

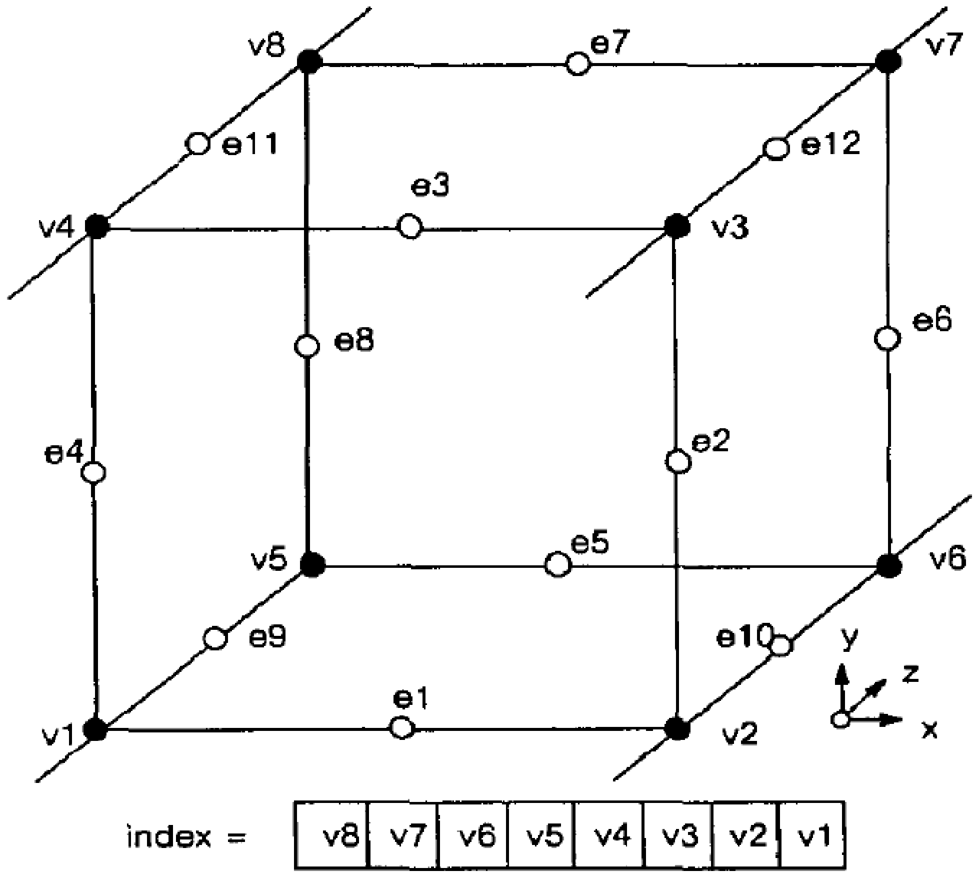




3D: isosurfaces



Marching cubes

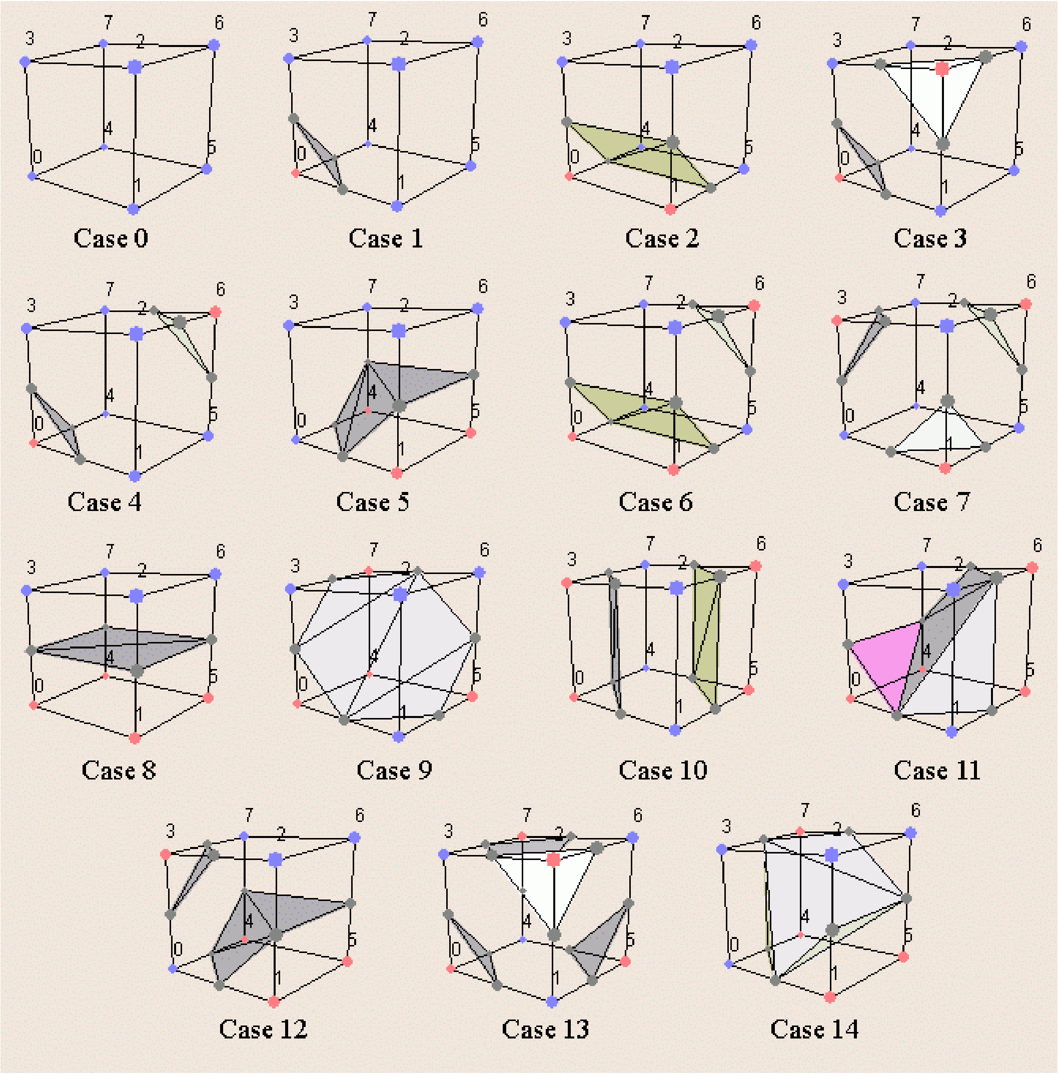


Marching squares/cubes

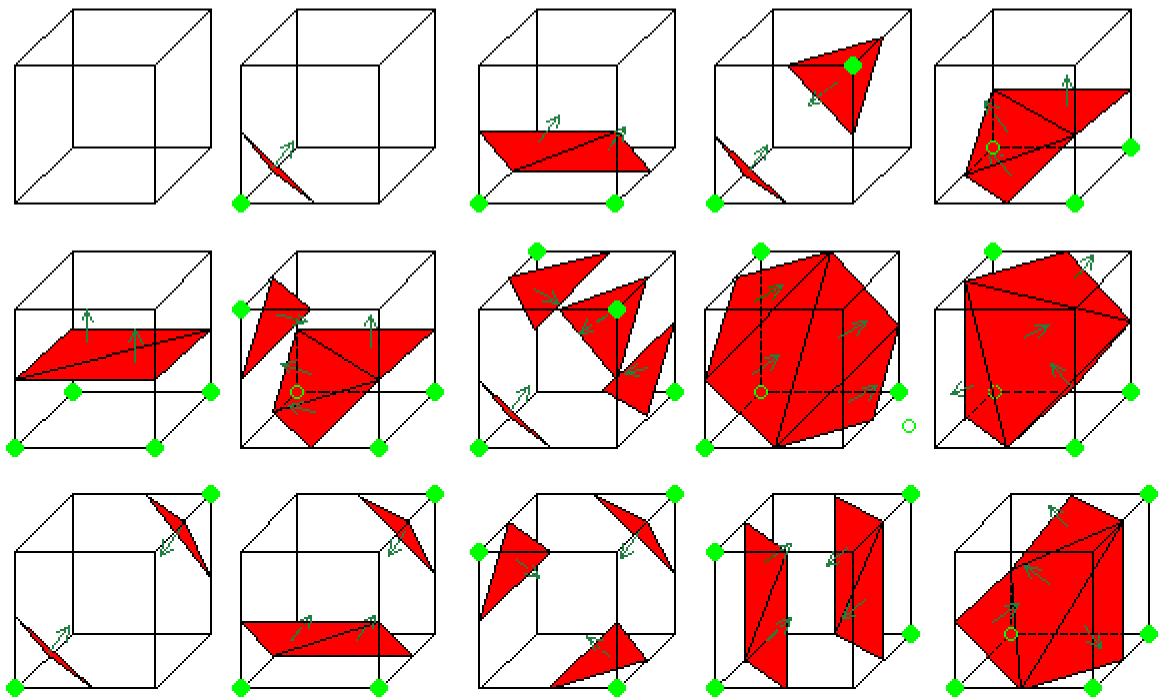
Algorithm:

1. March through the computational domain cell by cell
2. For each cell, determine whether or not the surface passes through it
3. If cell contributes to the surface, compute index into a look-up table using status of corner nodes
4. The index into the look-up table is used to determine how many triangles are needed + connectivity
5. Determine cell intersections by interpolation between intersection points

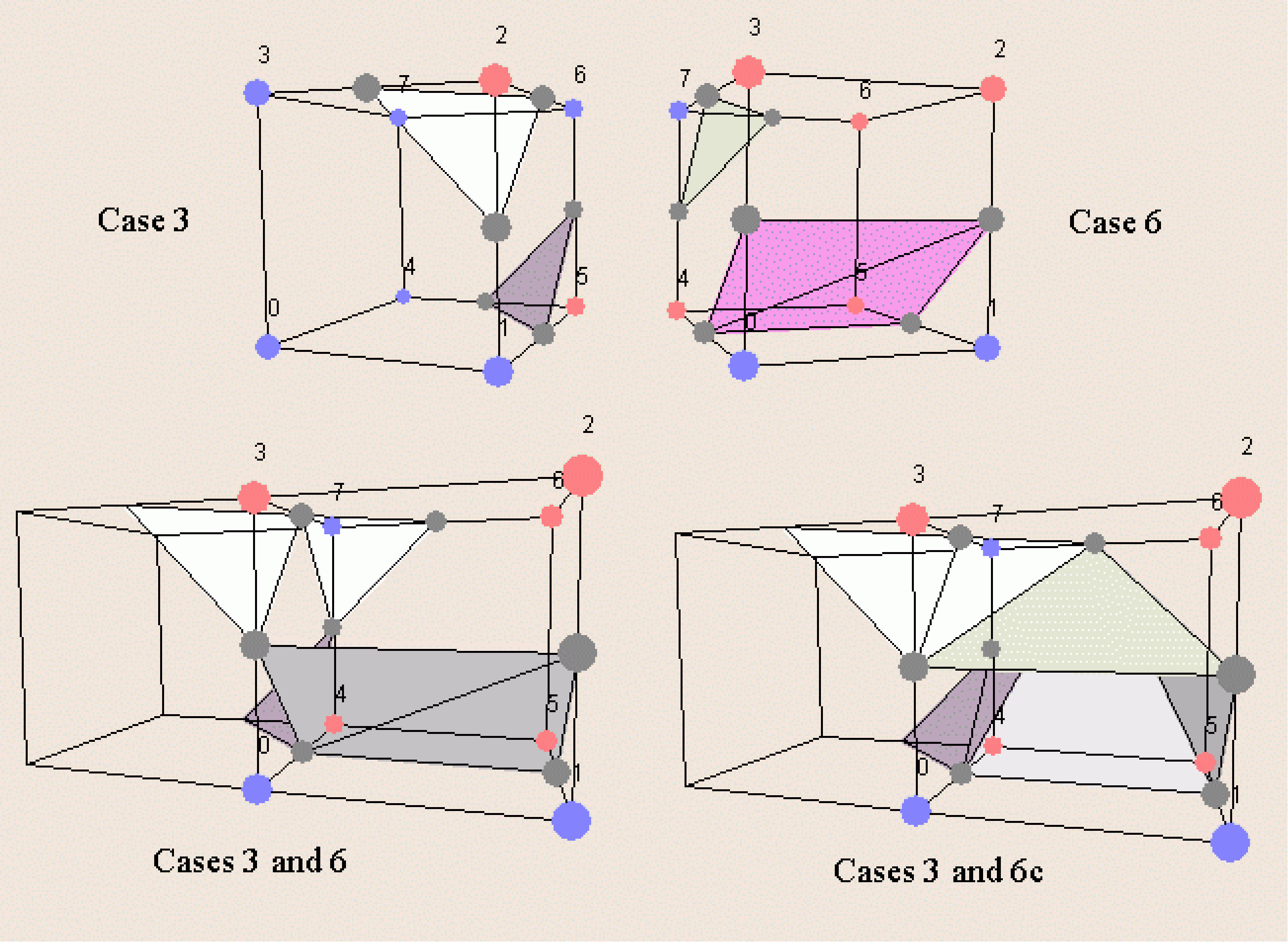
Marching cubes: cases



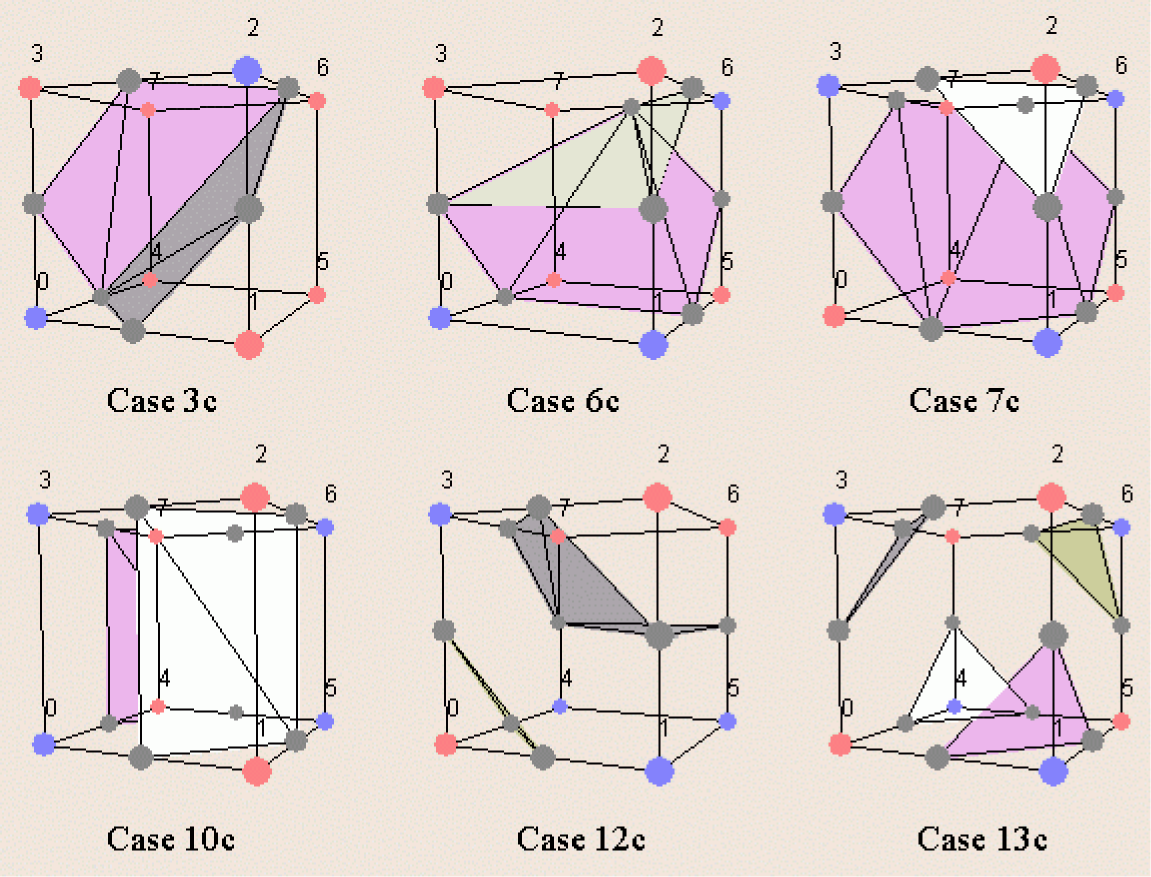
Marching cubes: surface orientation



Marching cubes: ambiguity

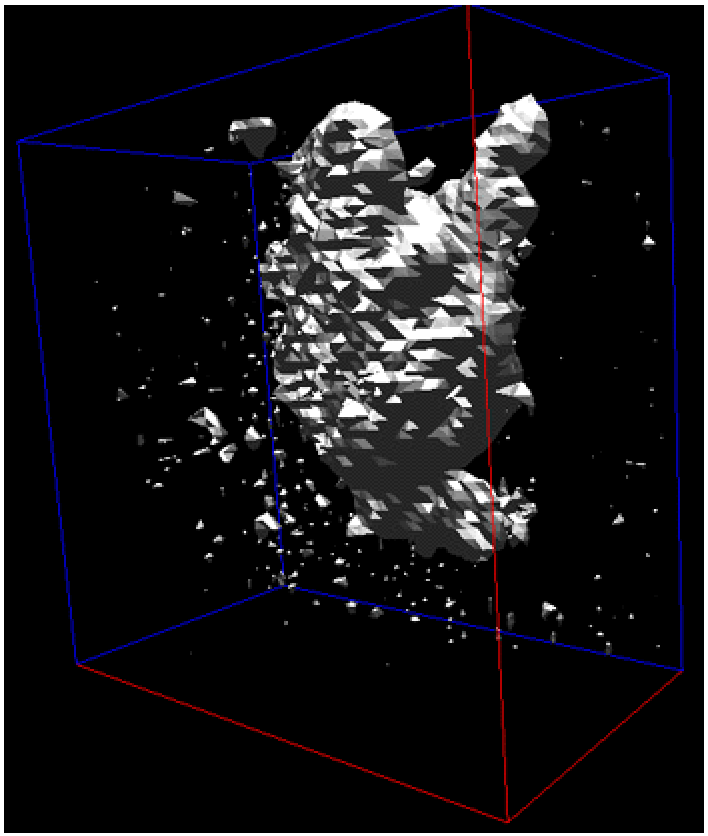


Marching cubes: additional cases

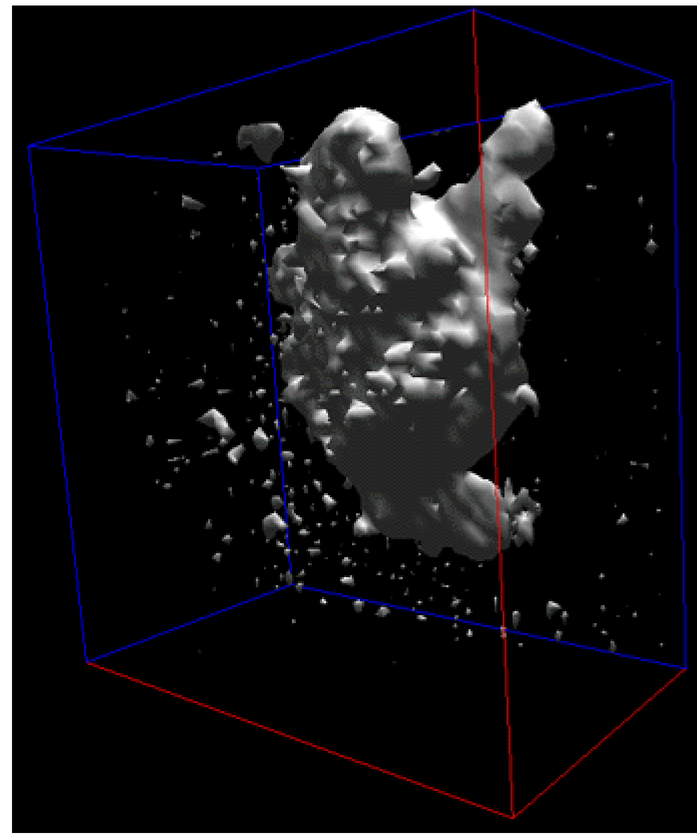


Vertex normals

Zonder vertex normals:



Met vertex normals:



Vertex normals

- Estimate gradient vector at cube vertices using central differencing:

$$G_x(i,j,k) = \frac{D(i+1,j,k) - D(i-1,j,k)}{\Delta x}$$

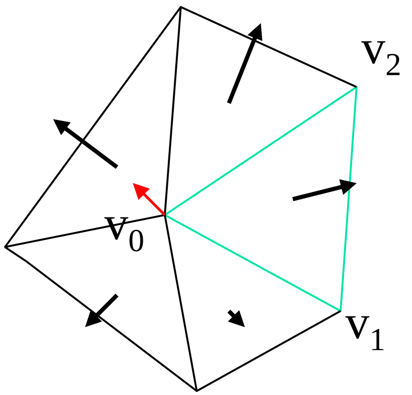
$$G_y(i,j,k) = \frac{D(i,j+1,k) - D(i,j-1,k)}{\Delta y}$$

$$G_z(i,j,k) = \frac{D(i,j,k+1) - D(i,j,k-1)}{\Delta z}$$

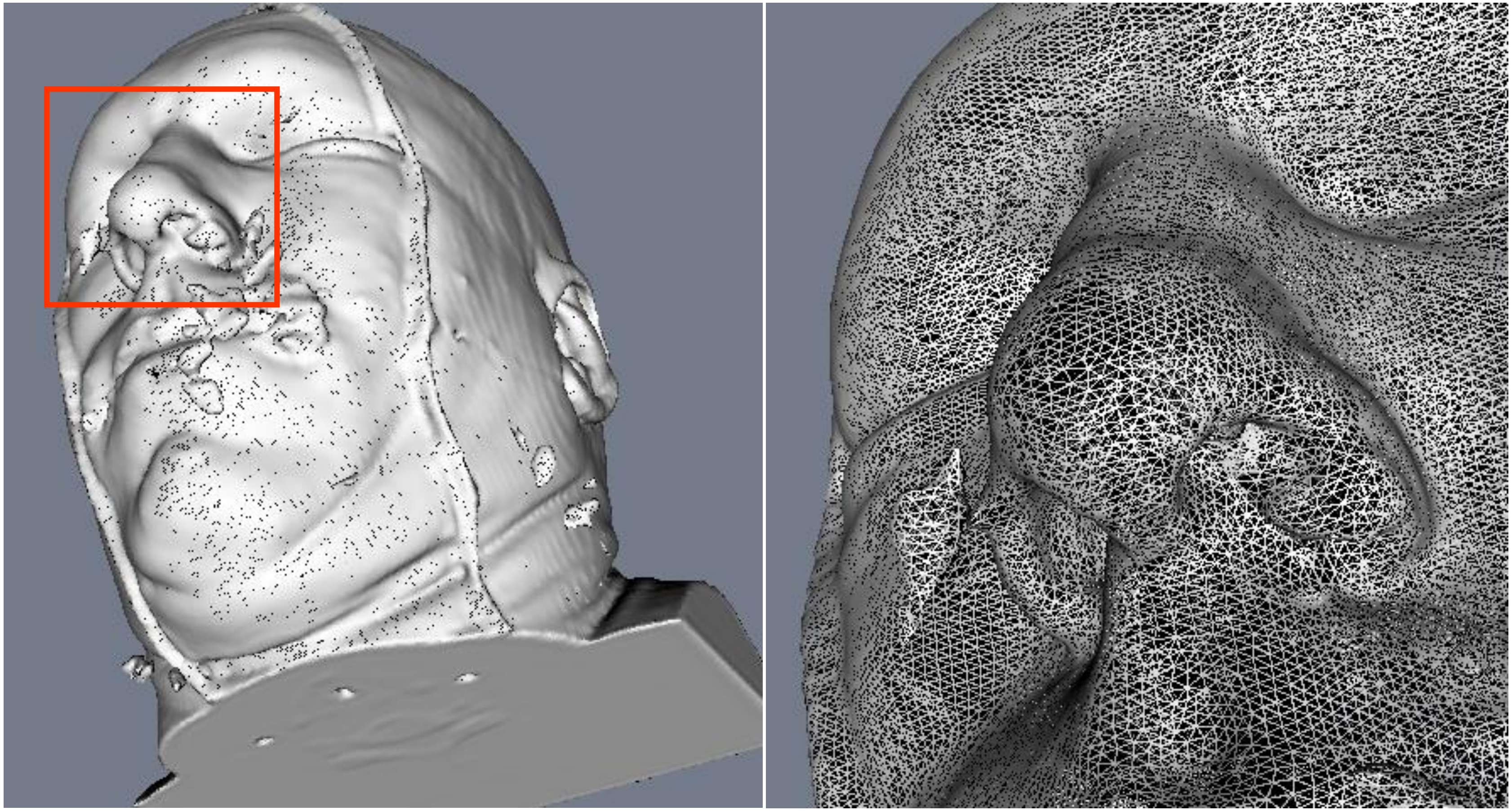
- Average of all face normals
  - Face normal:

    $$(v_0 - v_1) \times (v_1 - v_2)$$

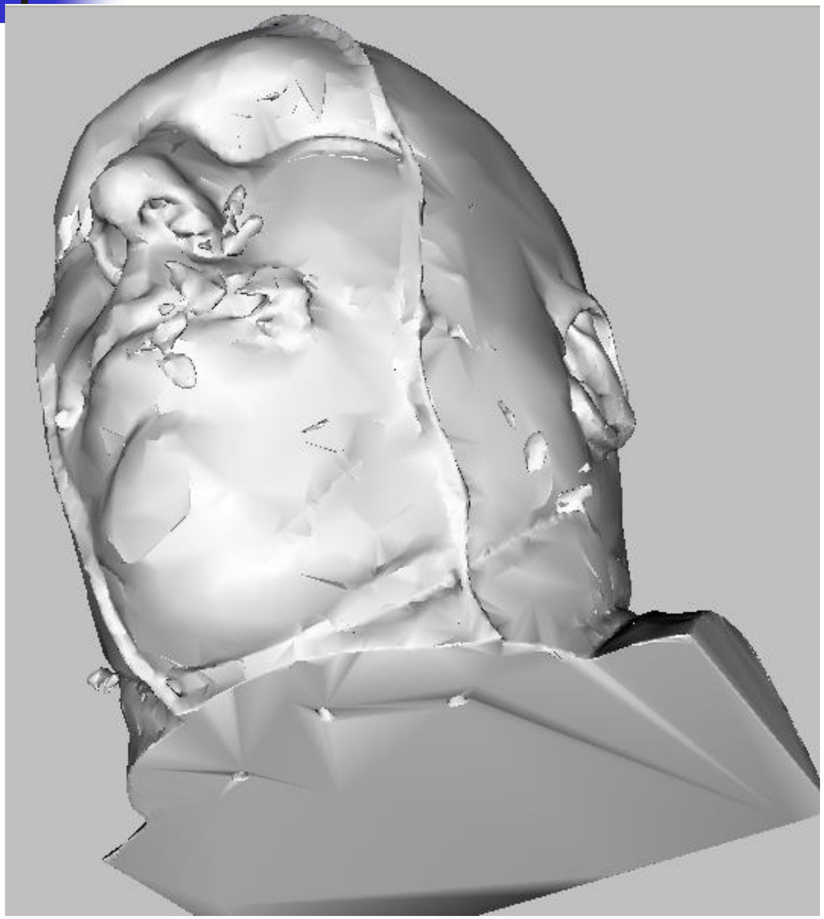
  - Vertex normal is average over all neighbouring face normals

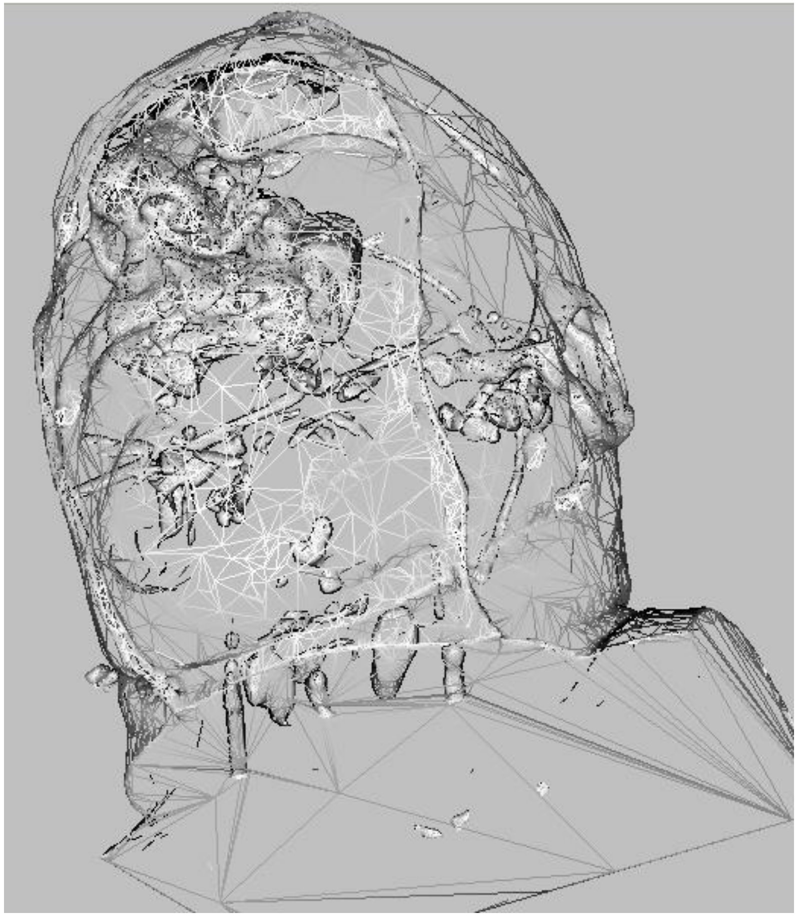


Examples: Visible Man



Examples: Visible Man - decimation

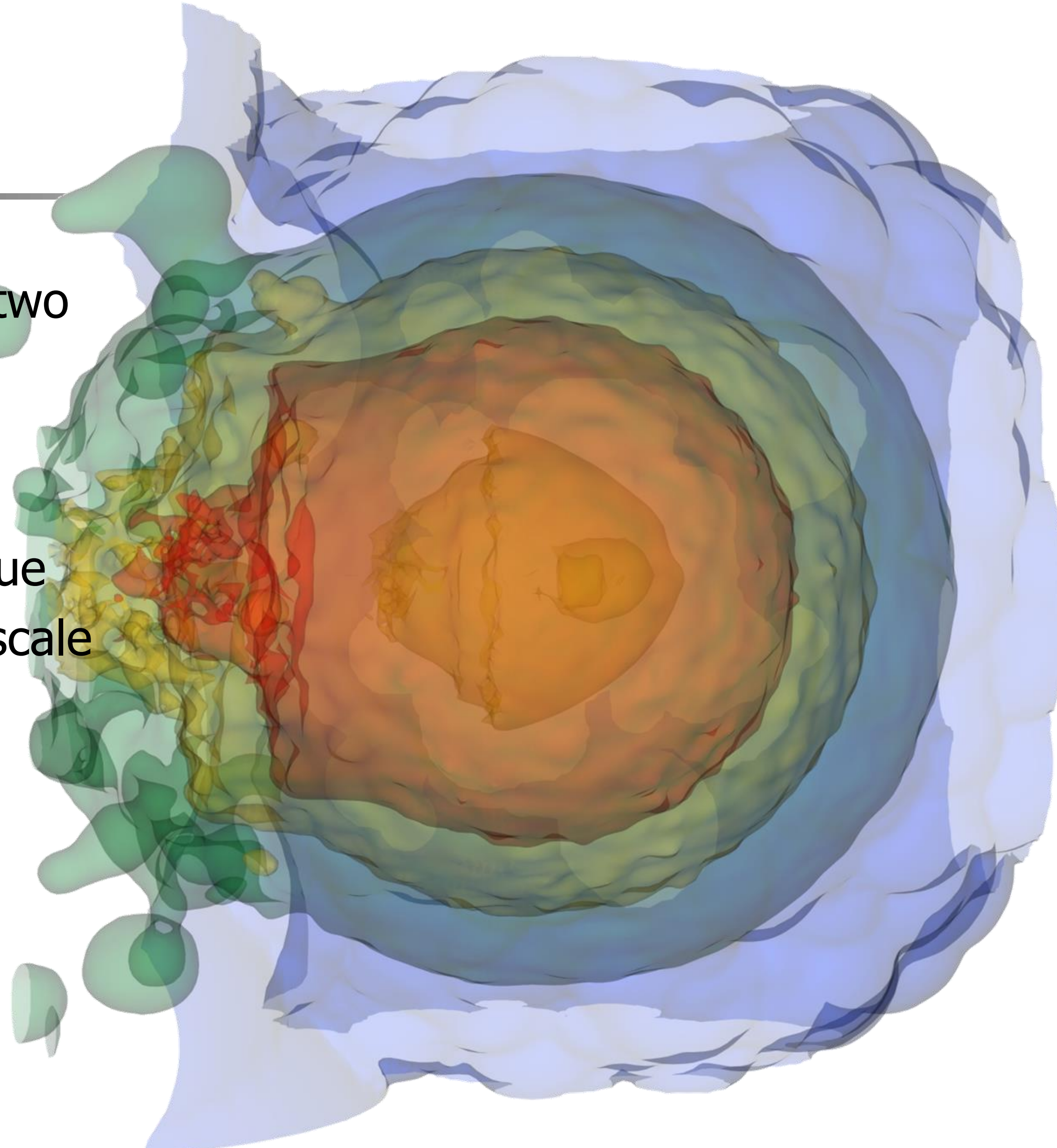




Examples

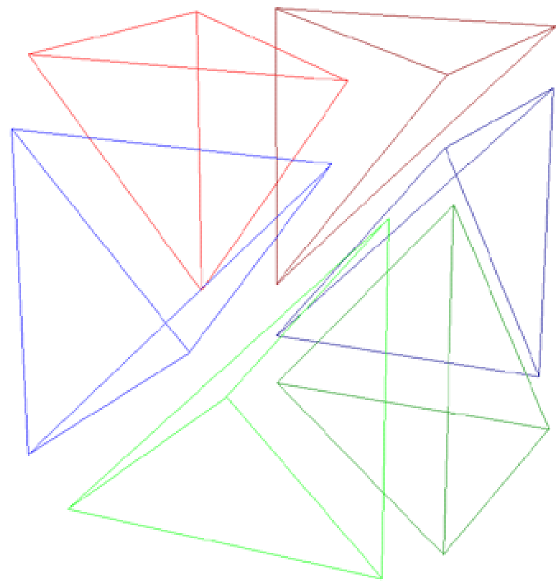
Hydrodynamics of two colliding stars

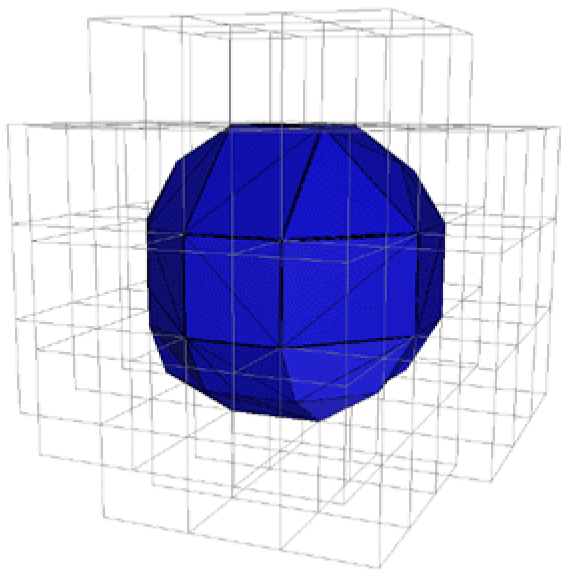
- 10 isosurfaces
- each 10% opaque
- ^{10}log threshold scale



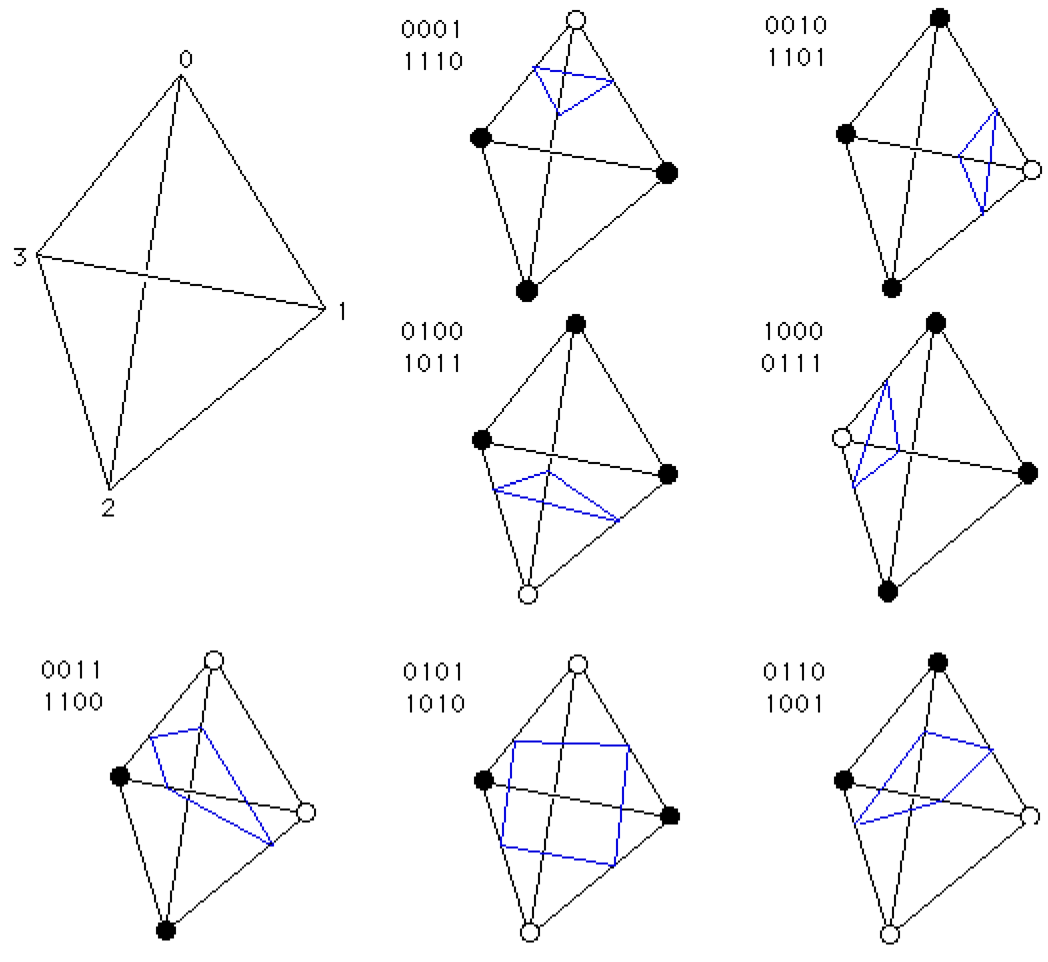
Marching tetrahedra

- Each mesh cell split into 6 tetrahedra
- Does not suffer from ambiguities
- Generated surfaces contain many more triangles

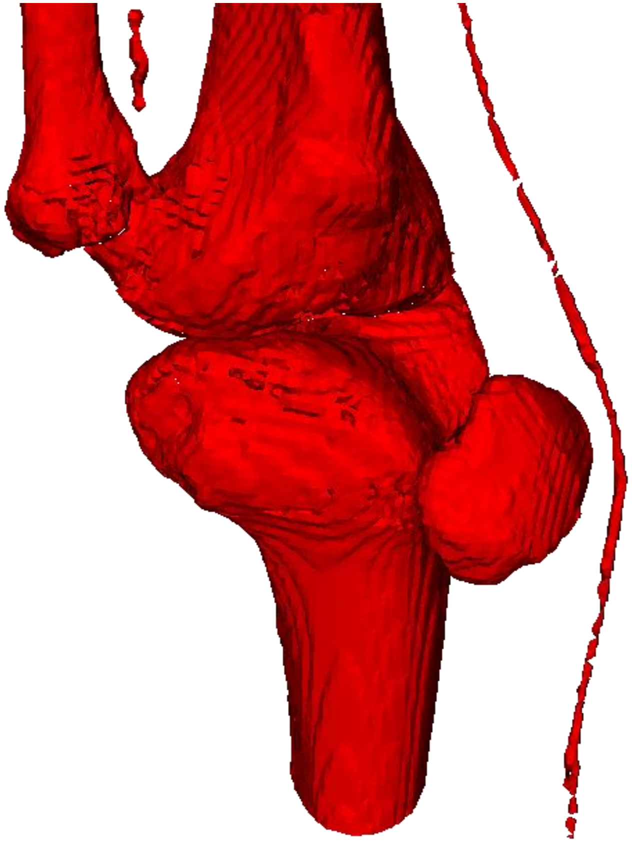




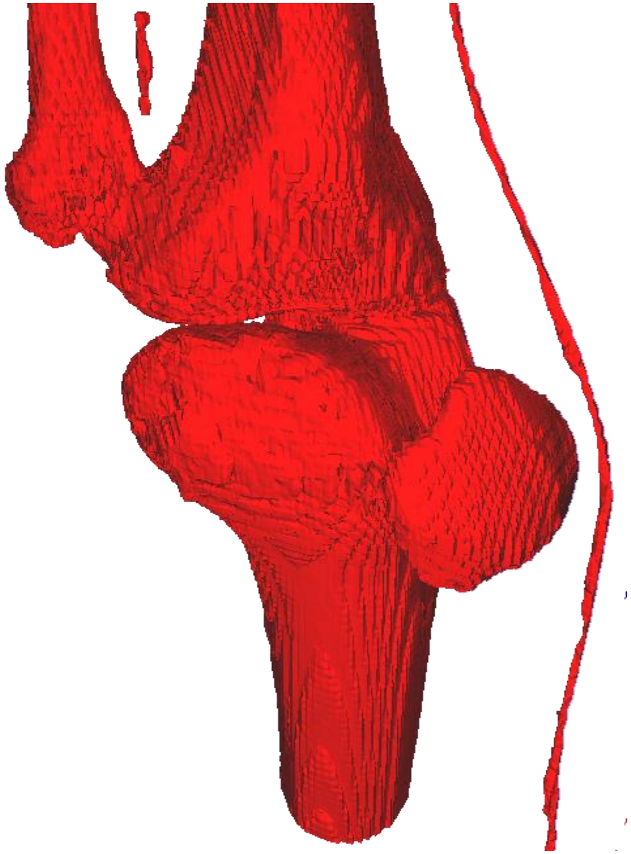
Marching tetrahedra



Vergelijking



Marching cubes
(~76,000 triangles)



Marching tetrahedra
(~236,000 triangles)