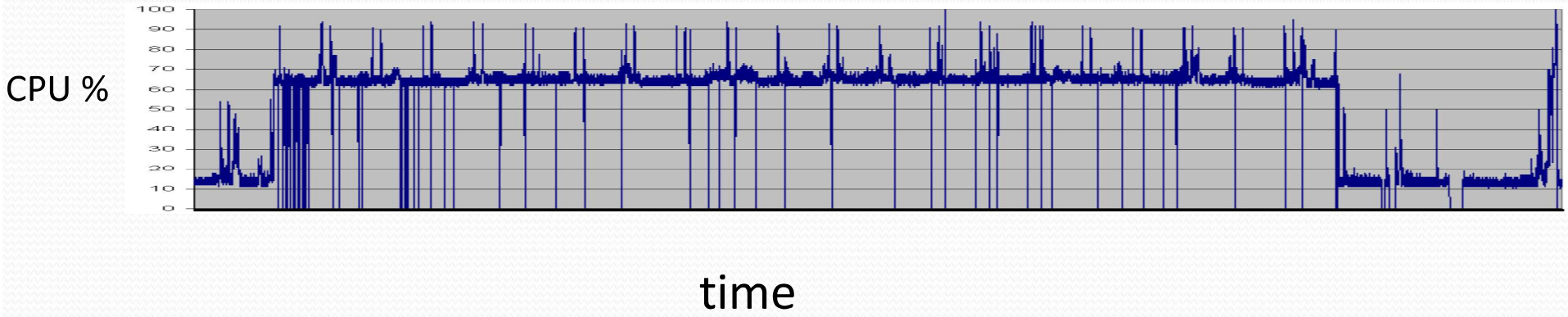
Some examples of complex representations

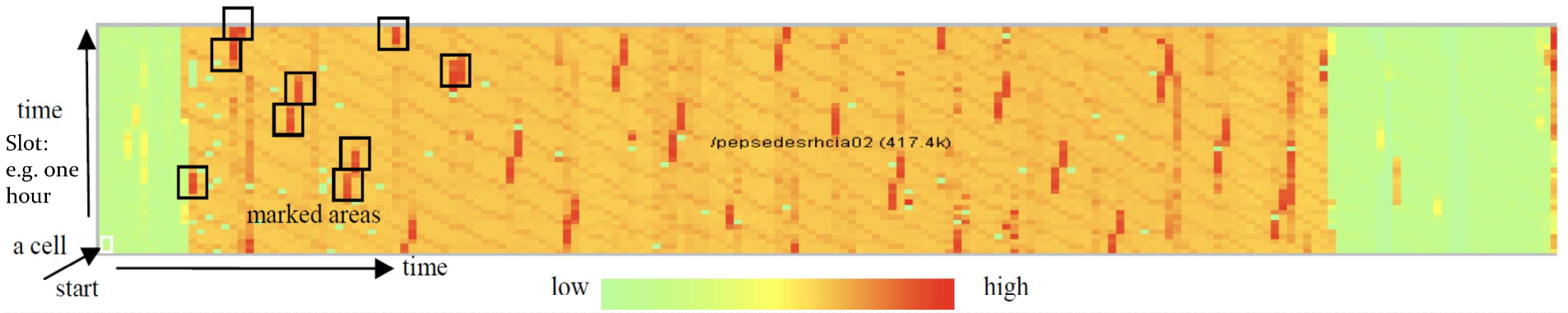
Example 1
Pixel oriented visualizations

Visual Analytics of Anomaly Detection in Large Data Streams
(paper from Daniel Keim group)

- You have to monitor a network composed of 8 systems with 16 servers each
- Each server provides basic information
  - CPU % occupation
  - DISK % occupation
  - MEM % occupation
  - …
  - That corresponds to 128 temporal data streams (overplotting !!)



Pixel oriented visualization overview



28 days (5 min windows), about 8k observations

Each observation takes a pixel

8 pixels (8 systems) per cell

The color codes the CPU %

Monitoring at server level

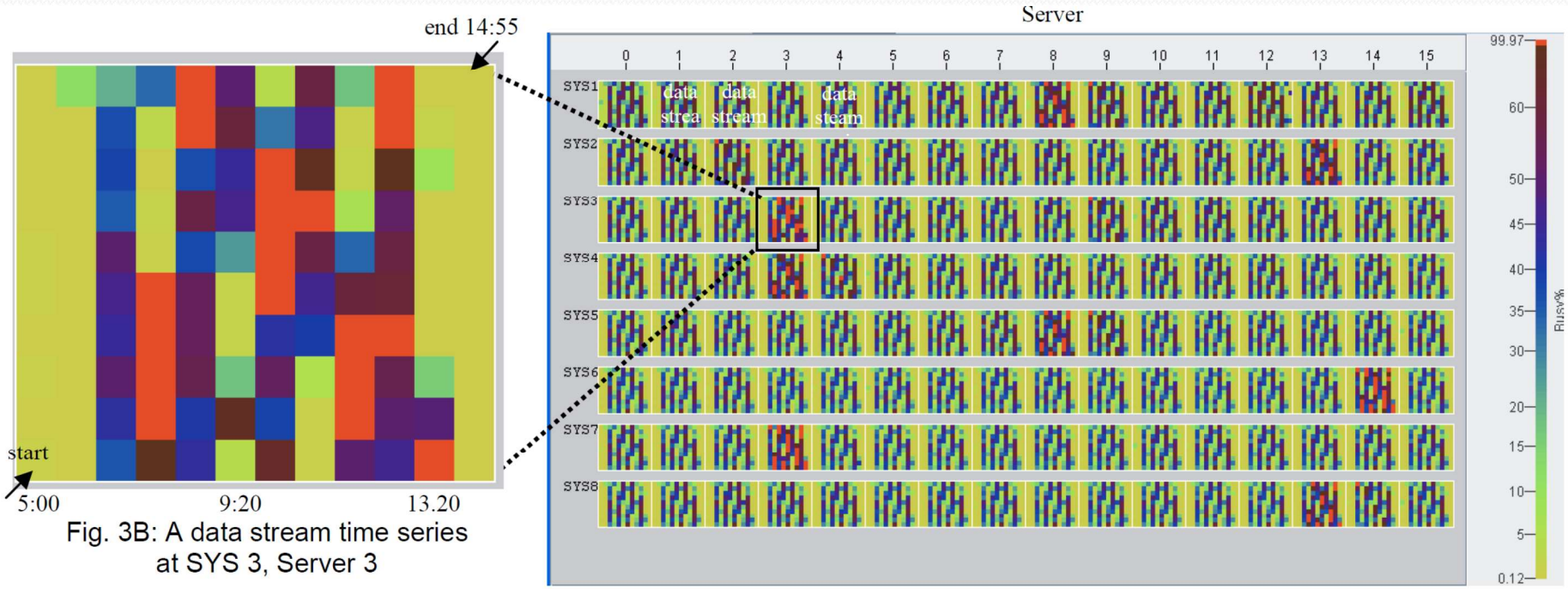


Fig. 3B: A data stream time series at SYS 3, Server 3

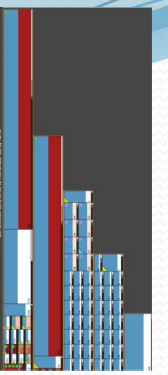
Color is preattentive!

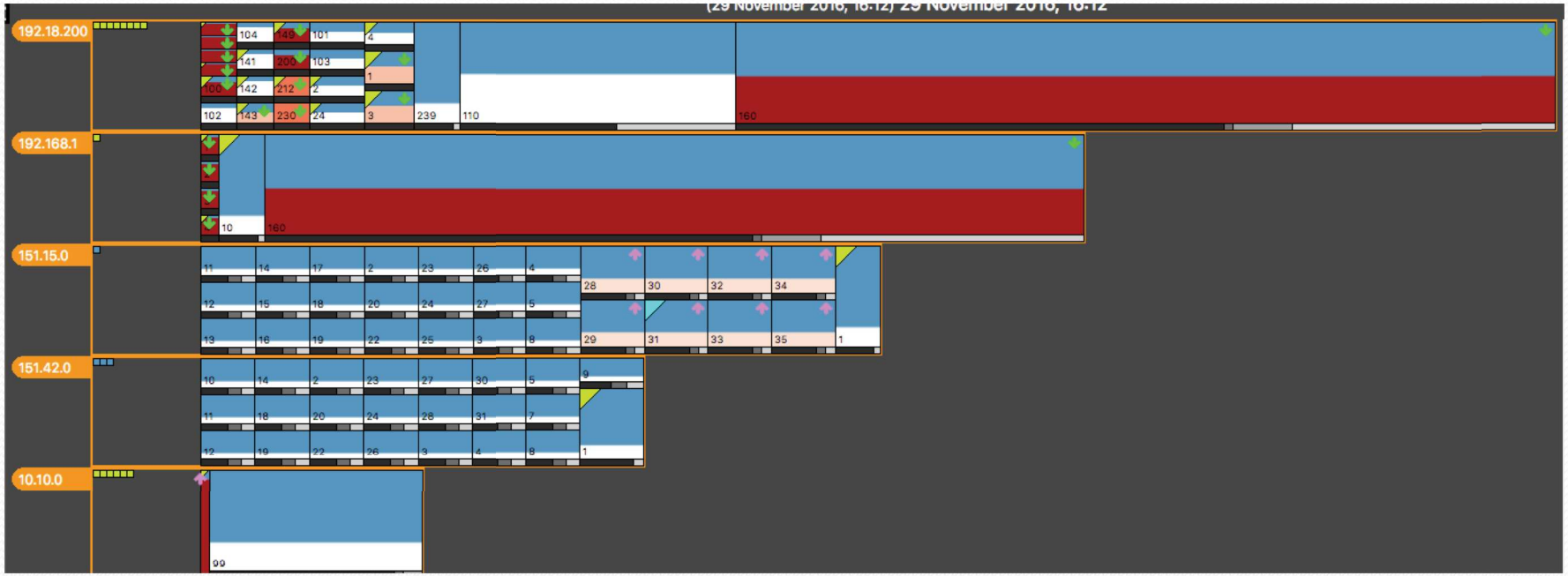
458757626808609928083982698028
747976296262867897187743671947
746588786758967329667287682085

Example 2
Security Domine

Combining/modifying existing representation

Histogram + 1 level treemap



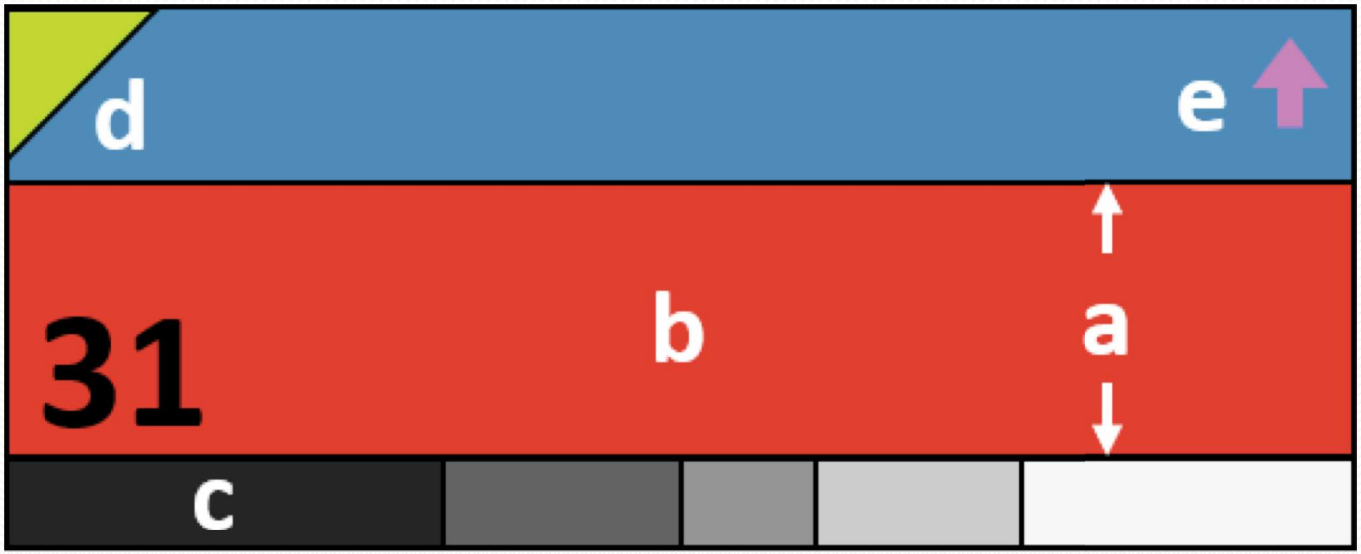


Length = number of vulnerabilities in a sub-network

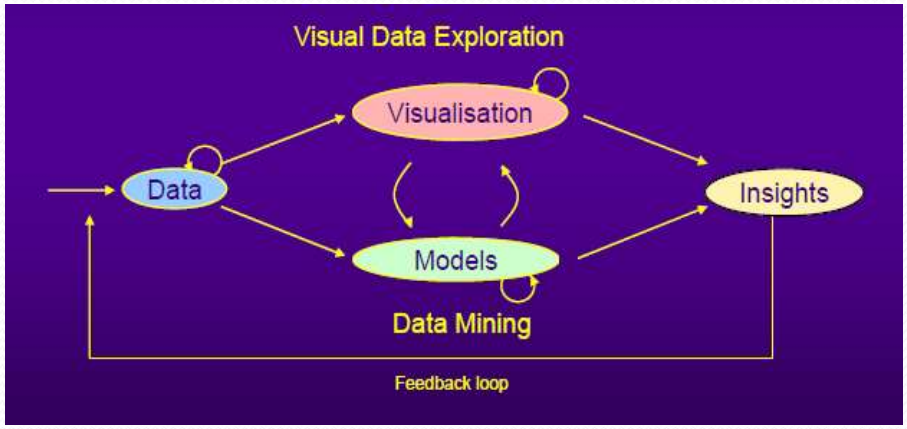
Size = number of vulnerabilities in a node

Special representation for nodes with 0 vulnerabilities

Node encoding



Role of analytics within representation phase



We can classify automatic activities in three main groups

1. Deriving new values from the dataset for ad-hoc visualization
   - This is the less standard and the more creative part of the process
2. Data reduction / data mining
   - Clustering /classification /…
   - Sampling
   - Dimensionality reduction (coming soon…)
3. Visualization improvement
   - Data distribution
   - Perceptual issues
   - Cognitive issues

Example 3
Information Retrieval Domine

Deriving new values from the dataset for ad-hoc visualization (you visualize DERIVED data)

The Context

- Information retrieval (IR) evaluation is not a theoretical issue
- In the last 20 years, large-scale evaluation campaigns have the goal of assessing performance of IR engines
  - TREC [Text REtrieval Conference] (USA) and CLEF [Cross-Language Evaluation Forum](Europe)
  - Hundreds of research groups
  - Producing a huge amount of valuable data to be analyzed, mined, and understood

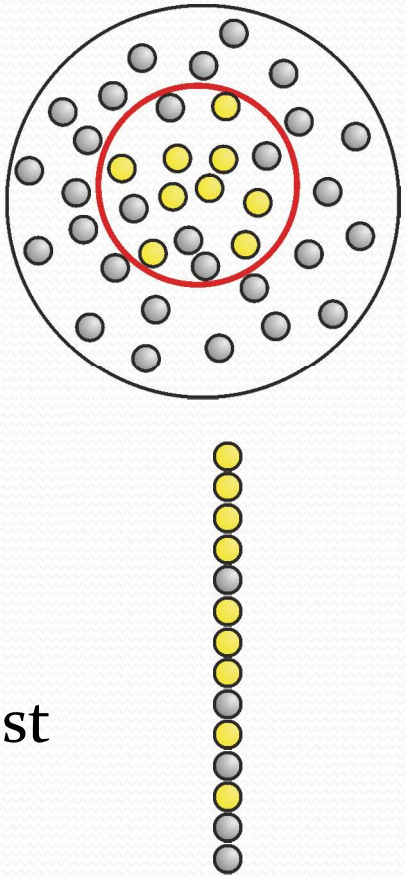
The Cranfield methodology

- Shared experimental collections :
  - To create comparable experiments
  - To evaluate their performance

- An experimental collection is a triple:
  - D is a set of documents
  - Q is a set of topics simulating actual user information needs (i.e., queries)
  - J is a (hand made!) set of relevance judgments, assigning a value to each document

Different experiment goals

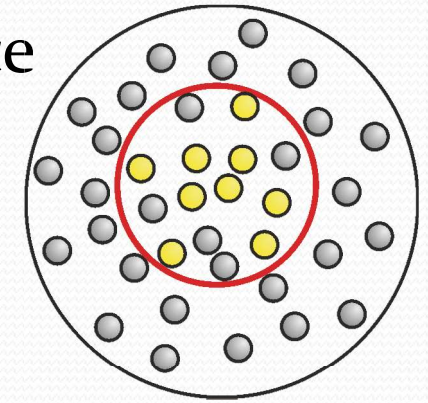
Given a topic q_i and a set of relevant and not relevant documents.



- Catch 'em all ! (the relevant ones)
  - Precision (only relevant ones) P=9/14≈64%
  - Recall (as many as possible) R=9/9 =100%
- Sort 'em all!
  - Rank the result to present to the user the most relevant documents in top position

Ranking!

- We focus on assessing the ranking quality in the challenging situation in which the relevance judgment is not binary
  - RANK: = 3 = 2 = 1 = 0



- Our goal is to provide precise indications on:
  - Misplaced documents (in the ranked list)
  - Quantitative problems rising from such misplacements

Modeling the problem

- We use a judgment function that assigns a value [0..k] to each document (0 non relevant, k highly relevant)
- We model the retrieved result with a ranked vector V of n elements
- Typical values: k=3, n=200
- The ground truth GT(V) function returns the relevance values

V	GT(V)
id1	3
id2	1
id3	2
id4	3
id5	2
id6	2
id7	3
…	…

Modeling the problem(cont.)

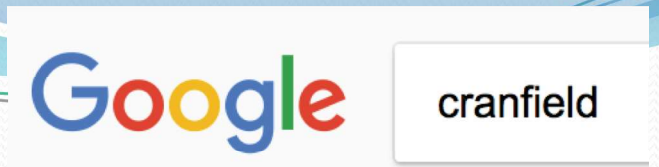
- For the end user, the value of a retrieved document is associated with its position in the vector (the lower the position the less likely the user will read it)
- We model it with a discounting function DF that progressively reduces the relevance of a document GT(V[i]) as i increases

$$DF(V[i]) = \begin{cases} GT(V[i]), & if\ i \leq x \\ GT(V[i])/\log_x(i), & if\ i > x \end{cases}$$

x=2 impatient users … x=10 patient users

GT(V)
3
1
2
3
2
2
3
…

You know this stuff…



Welcome to Cranfield University
https://www.cranfield.ac.uk/ ▾ Traduci questa pagina
Cranfield University unlocks the potential of people and organisations. We deliver research, postgraduate education and professional development.
Off-campus resources · Aerospace · How to apply

Cranfield School of Management - Cranfield University
https://www.cranfield.ac.uk/som ▾ Traduci questa pagina
Cranfield School of Management is one of the oldest business schools in ...

Università di Cranfield - Wikipedia
https://it.wikipedia.org/wiki/Università_di_Cranfield ▾
La **Cranfield** University è un'università esclusivamente post-laurea inglese basata sulla ricerca universitaria. I suoi corsi sono programmi universitari di secondo ...

Cranfield - Wikipedia
https://it.wikipedia.org/wiki/Cranfield ▾
Cranfield è un paese di 5.443 abitanti della contea del Bedfordshire, in Inghilterra. Altri progetti[modifica | modifica wikitesto]. Altri progetti. Wikimedia Commons.

Cranfield University - Wikipedia
https://en.wikipedia.org/wiki/Cranfield_University ▾ Traduci questa pagina
Cranfield University is a British postgraduate and research-based public university specialising in science, engineering, technology and management.

Cranfield - Wikipedia
https://en.wikipedia.org/wiki/Cranfield ▾ Traduci questa pagina
Cranfield is a village and civil parish in north west Bedfordshire, England, between Bedford and Milton Keynes. It has a population of 4,909, increasing to 5,369 ...

Modeling the problem(cont.)

- The overall quality of a result is evaluated using a discount cumulative gain function:

$$DCG(V,i) = \sum_{j=1}^{i} DF(V[j])$$

  - That estimates the information gained by a user that examines the first **i** documents of V

  - It is a summary measure and hides details

Modeling the problem(cont.)

- We consider the vector O, the optimal permutation of V that produces the highest DCG for each i

- We compute:
  - R_Pos, the relative position of documents in V with respect to their optimal position in O
  - Delta_gain(i), the difference of information gain DCG between V[i] and O[i]

Visualizing misplaced elements

The actual result

GT(V)	DF	DCG
3	3,00	3,00
1	1,00	4,00
2	1,26	5,26
3	1,50	6,76
2	0,86	7,62
2	0,77	8,40
3	1,07	9,47
2	0,67	10,13
0	0,00	10,13
1	0,30	10,43
0	0,00	10,43
3	0,84	11,27

OK

ABOVE

BELOW

The optimal result

GT(O)	DF	DCG
3	3,00	3,00
3	3,00	6,00
3	1,89	7,89
3	1,50	9,39
2	0,86	10,25
2	0,77	11,03
2	0,71	11,74
2	0,67	12,41
1	0,32	12,72
1	0,30	13,02
0	0,00	13,02
0	0,00	13,02

Misplaced elements and their numerical influence on DCG

The actual result

GT(V)	DF	DCG	DELTA GAIN
3	3,00	3,00	0,00
1	1,00	4,00	-2,00
2	1,26	5,26	-0,63
3	1,50	6,76	0,00
2	0,86	7,62	0,00
2	0,77	8,40	0,00
3	1,07	9,47	0,36
2	0,67	10,13	0,00
0	0,00	10,13	-0,32
1	0,30	10,43	0,00
0	0,00	10,43	0,00
3	0,84	11,27	0,84

OK

LOSS

GAIN

The optimal result

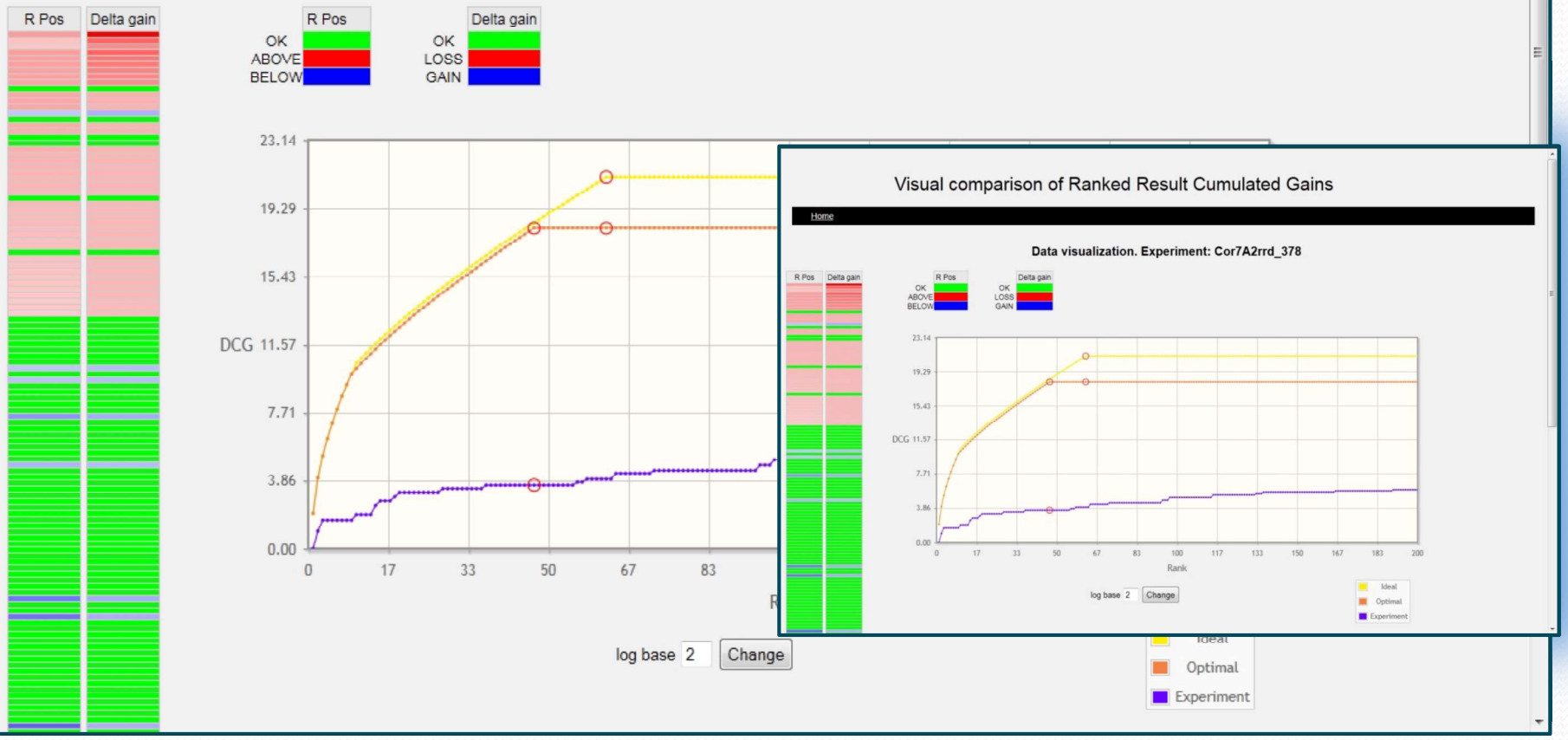
GT(O)	DF	DCG
3	3,00	3,00
3	3,00	6,00
3	1,89	7,89
3	1,50	9,39
2	0,86	10,25
2	0,77	11,03
2	0,71	11,74
2	0,67	12,41
1	0,32	12,72
1	0,30	13,02
0	0,00	13,02
0	0,00	13,02

promise

Visual comparison of Ranked Result Cumulated Gains

Home

Data visualization. Experiment: Cor7A2rrd_378



Visual comparison of Ranked Result Cumulated Gains
Home
Data visualization. Experiment: Cor7A2rrd_378

Representation=

Data (Data position and rank) + Derived data:

V	GT(V)
id1	3
id2	1
id3	2
id4	3
id5	2
id6	2
id7	3
...	...

Misplacement (color)
Delta Gain (color)
Computed using
DCG and R_Pos

GT(V)	DF	DCG	DELTA GAIN
3	3,00	3,00	0,00
1	1,00	4,00	-2,00
2	1,26	5,26	-0,63
3	1,50	6,76	0,00
2	0,86	7,62	0,00
2	0,77	8,40	0,00
3	1,07	9,47	0,36
2	0,67	10,13	0,00
0	0,00	10,13	-0,32
1	0,30	10,43	0,00
0	0,00	10,43	0,00
3	0,84	11,27	0,84

Example 4
Text analysis

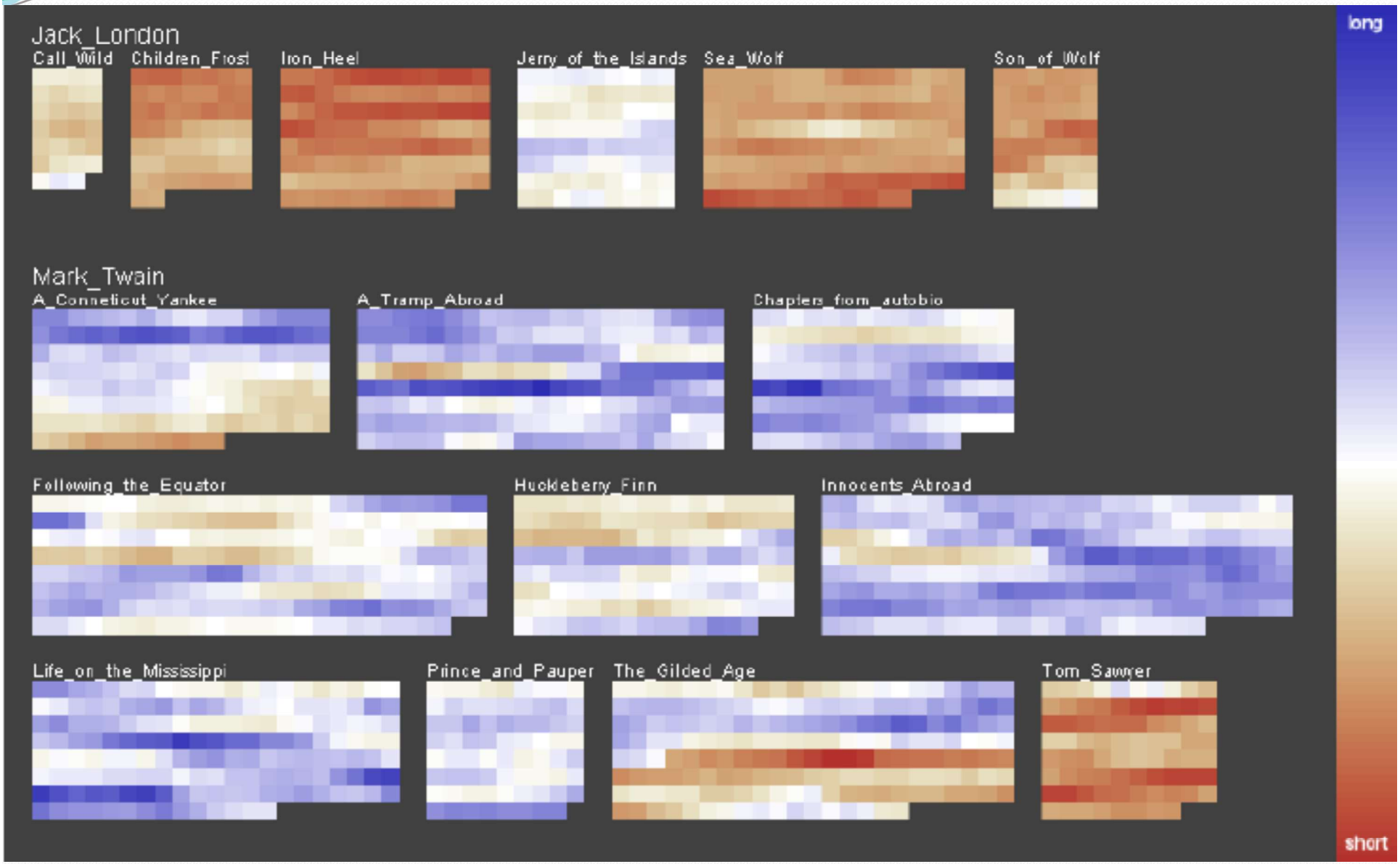
Deriving new values from the dataset for ad-hoc visualization (you visualize DERIVED data)

Deriving new values from the dataset for ad-hoc visualization

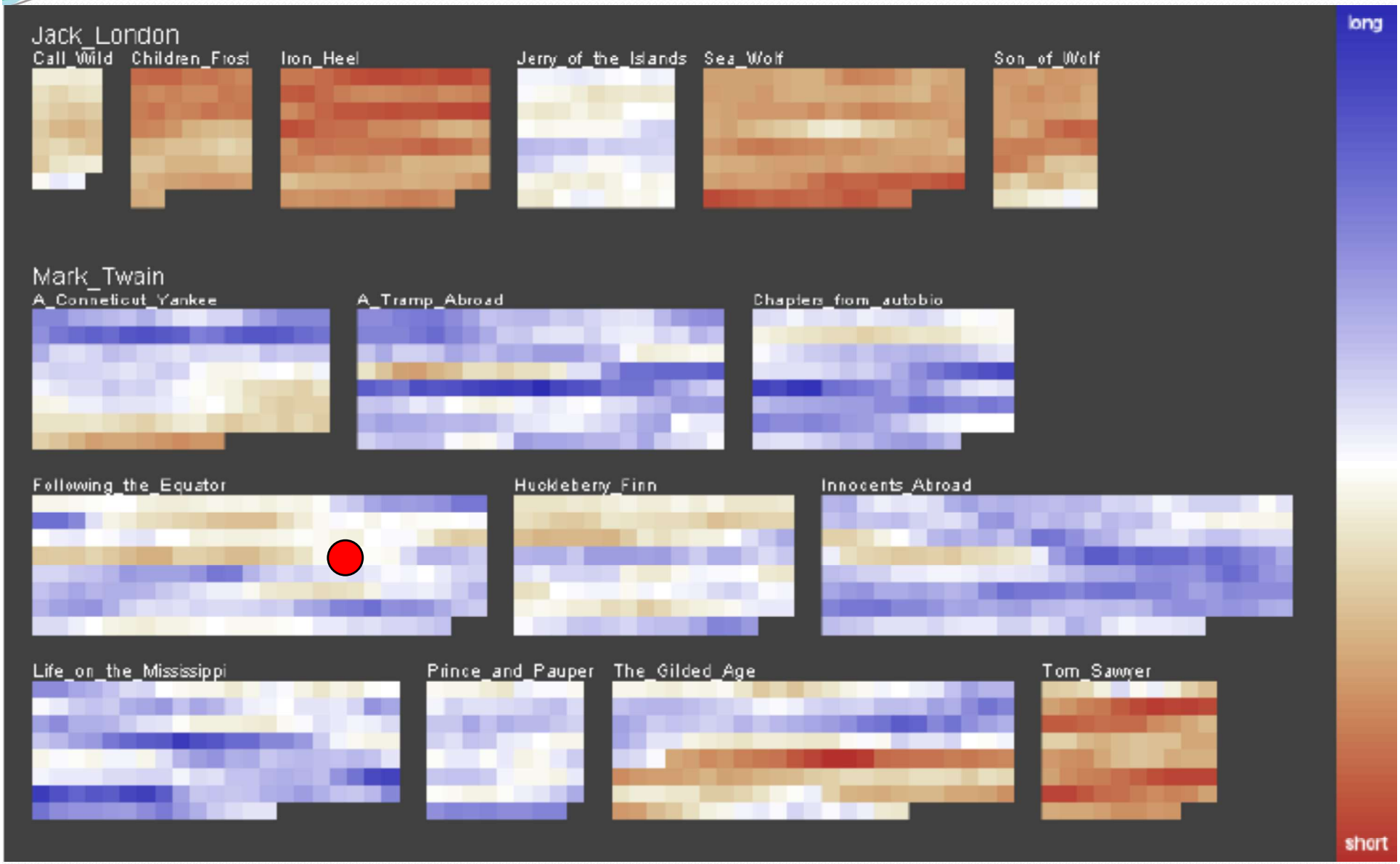
- How to visually compare J. London and M. Twain books ?
- [D. A. Keim and D. Oelke. Literature Fingerprinting: A New Method for Visual Literary Analysis. 2007 IEEE Symp. on Visual Analytics Science and Technology (VAST '07)]

1. Split the book in several text blocks (e.g., pages, paragraph, sentences)
2. Measure, for each text block, a relevant feature (e.g., average sentence length, word usage, etc.)
3. Associate the relevant feature to a visual attribute (e.g., color)
4. Visualize it

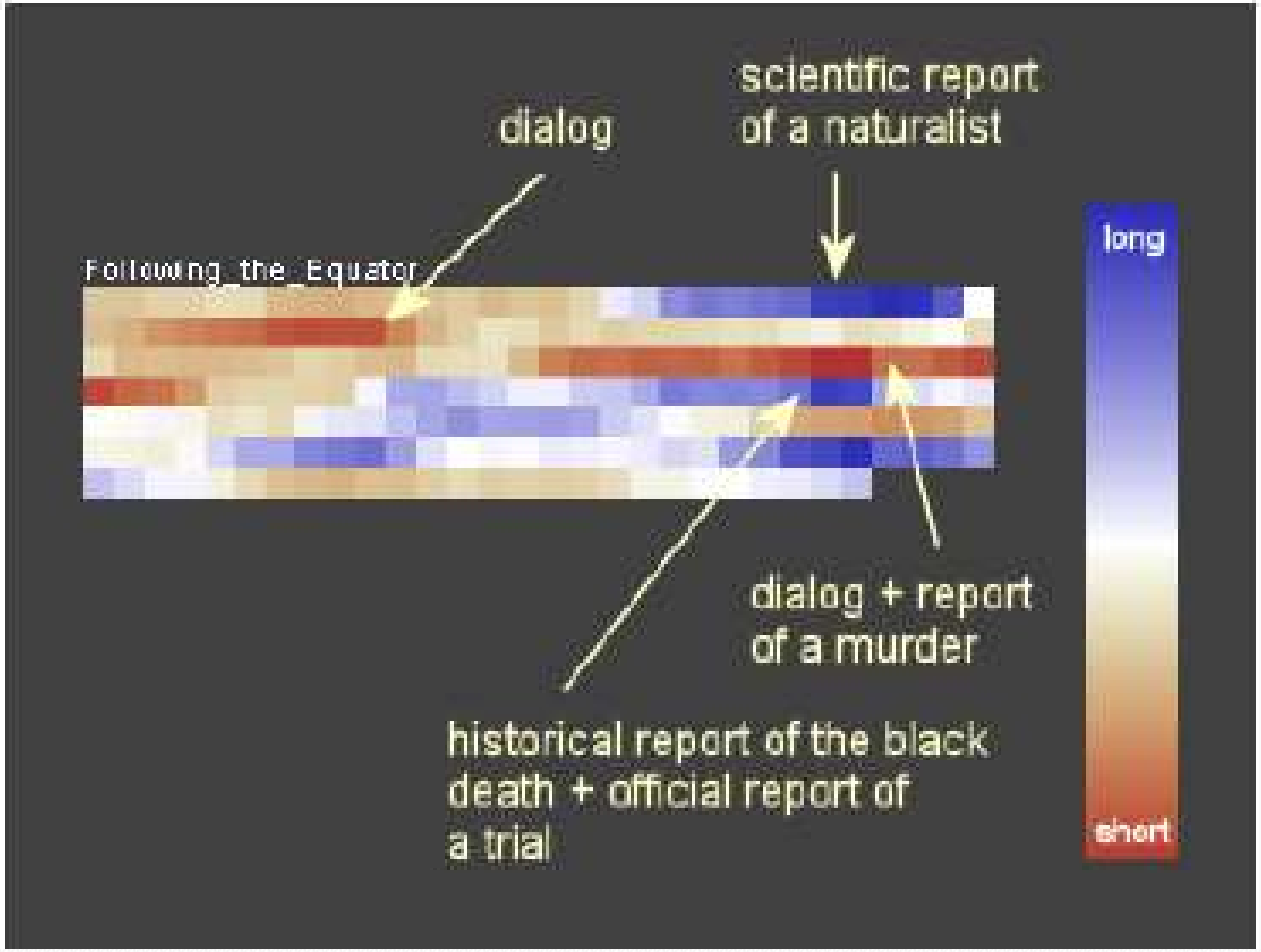
J.London vs M.Twain average sentence lengths



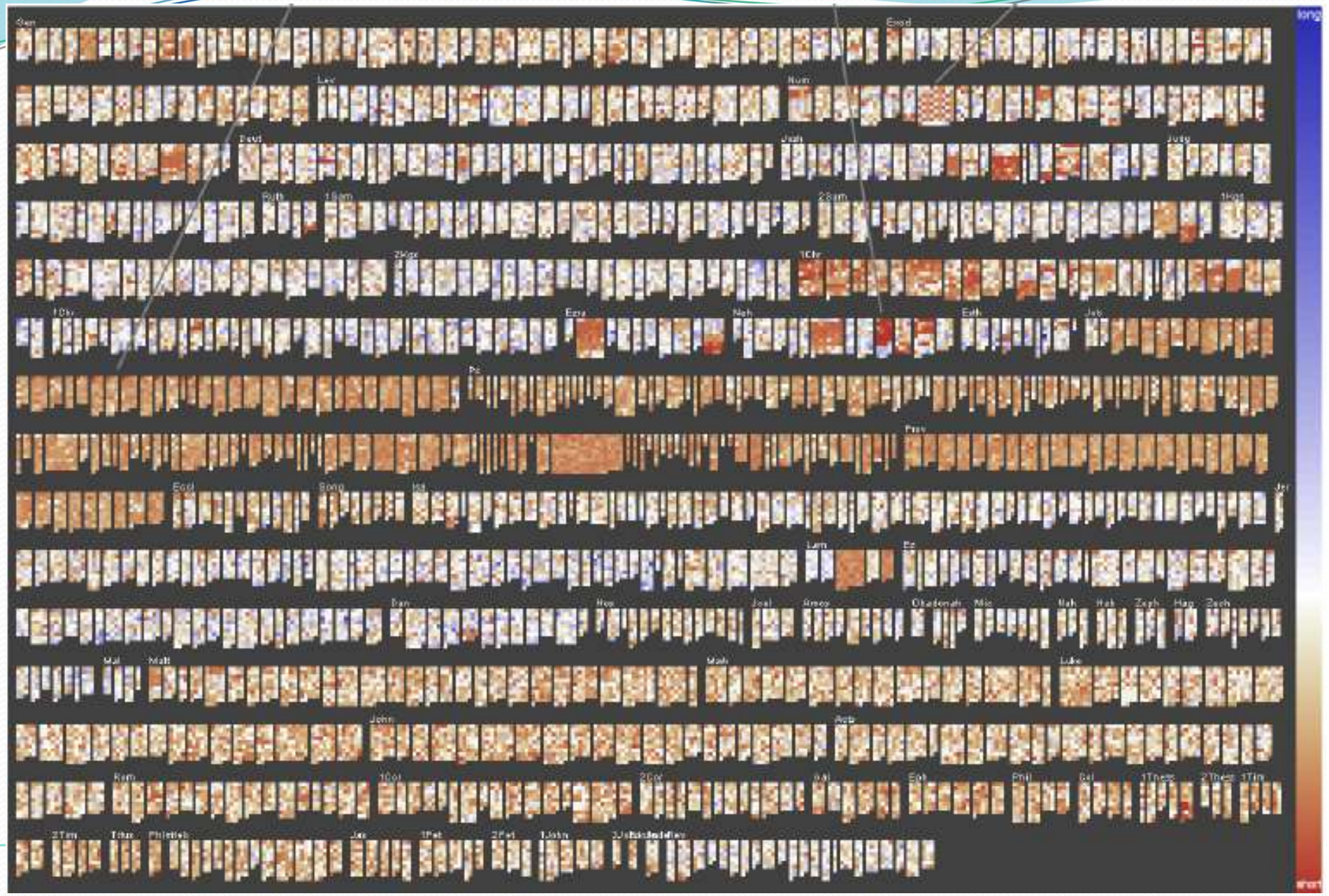
User interaction (a non uniform book?)



Details of a book



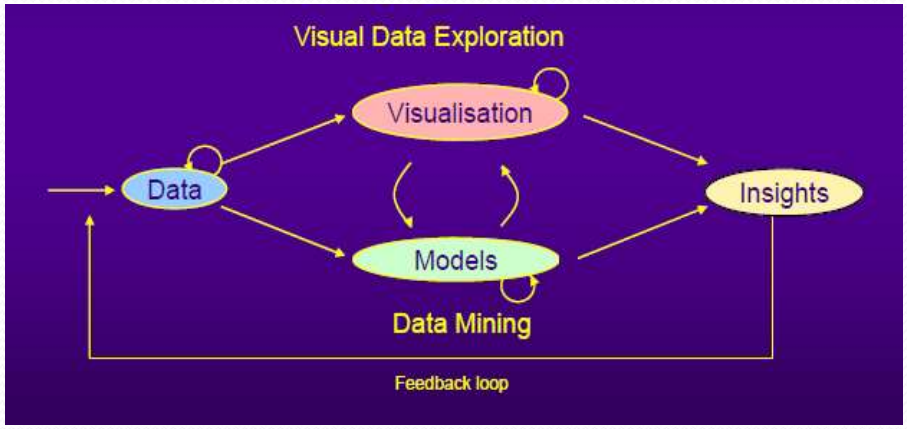
What about the Bible?



Example 5
Density maps

Improving the visualization using data distribution

Role of analytics within representation phase



We can classify automatic activities in three main groups

1. Deriving new values from the dataset for ad-hoc visualization
   - This is the less standard and the more creative part of the process
2. Data reduction / data mining
   - Clustering /classification /…
   - Sampling
   - Dimensionality reduction
3. Visualization improvement
   - Data distribution
   - Perceptual issues
   - Cognitive issues

Problem

- A very basic Infovis activity:

  mapping data values to a visual attribute (e.g., color, size, thickness, etc.)

$$D = \{ d_1, \ldots, d_{N_{Dv}} \}$$

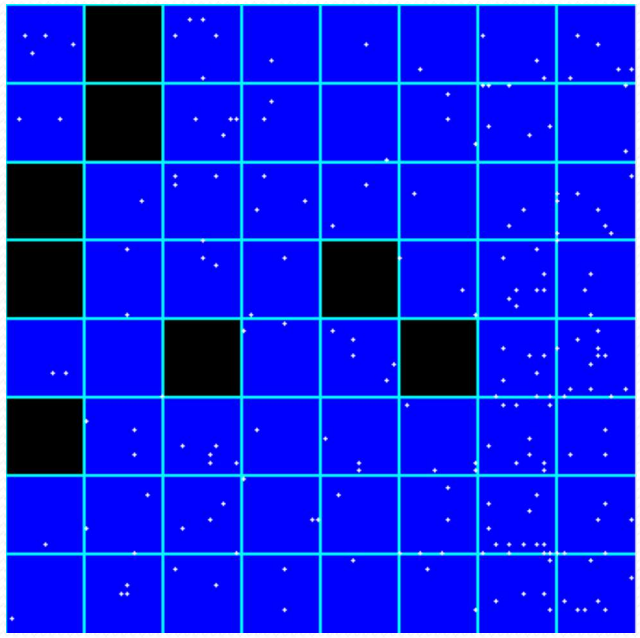
$$VisualAttribute = \{ C_1, \ldots, C_{N_L} \}$$

We discuss the matter in the context of
2D scatter plot density maps

Density maps

4 data items are plotted on the same pixel:**d=4**

empty pixel



8x8 pixels

we can **map** the density values to a 256 levels gray or color scale

In the example we borrow the Keim&Kriegel [KK95] color scale, presenting a monotonically increasing brightness

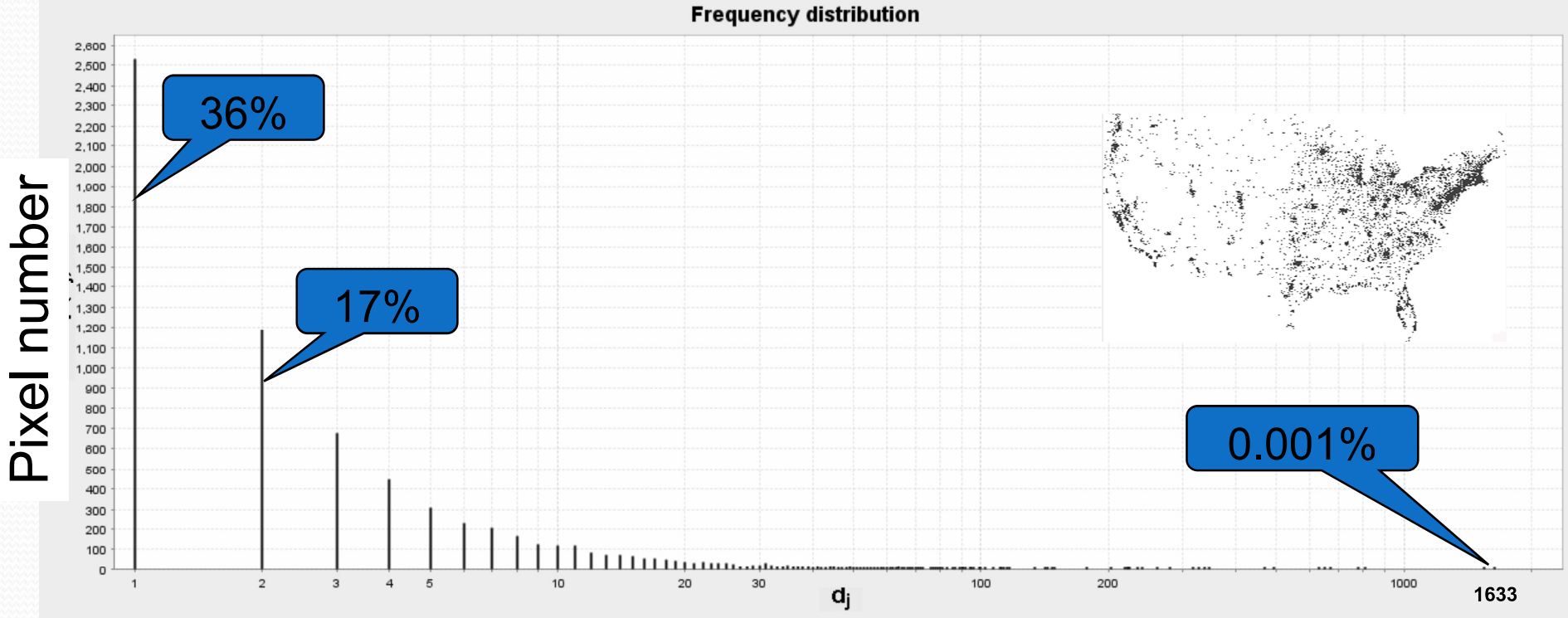
The case study (Infovis contest 2005)

The case study (Infovis contest 2005)

- About 60,000 USA companies plotted on a 800x450 (360,000 pixels) scatter plot
- N_{DV}=126 distinct density values (collisions on the same pixel) ranging on [1..1,633]
- N_{AP}=7,042 active pixels (i.e., hosting at least one company):
  - 2526 pixels (36%) host exactly one company (d=1)
  - 1182 pixels (17%) host two companies (d=2)
  - …
  - 1 pixel (0.0001 %) hosts 1633 companies (d=1633)

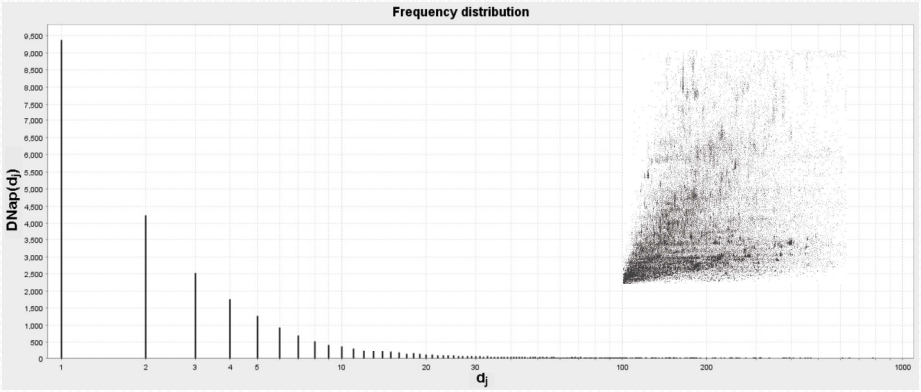
What is the source of the problem?

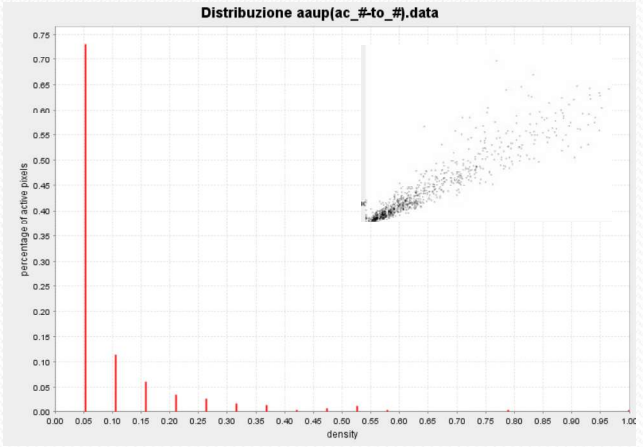
- The choice of the right **mapping** is crucial, because of density frequency distribution presents very skewed behaviour

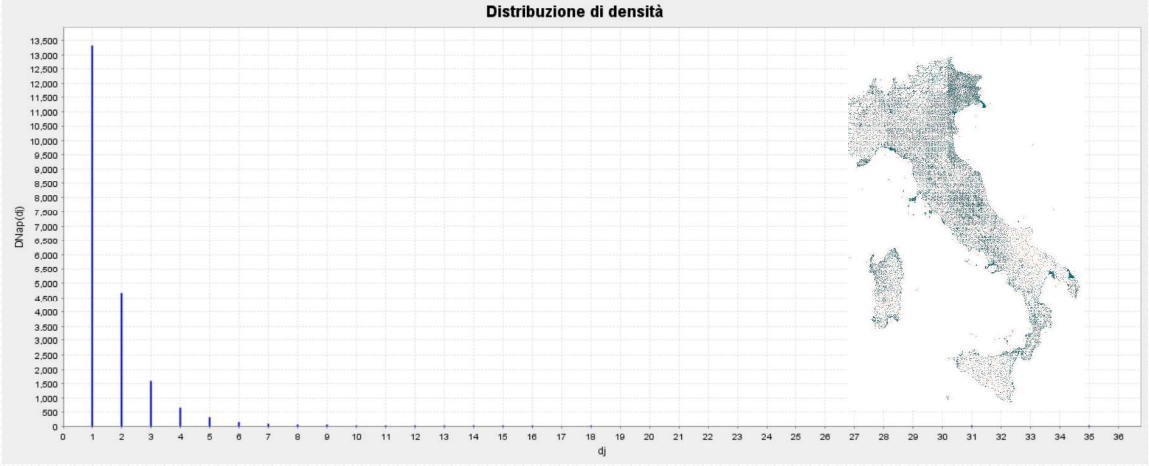


A (quick) parenthesis

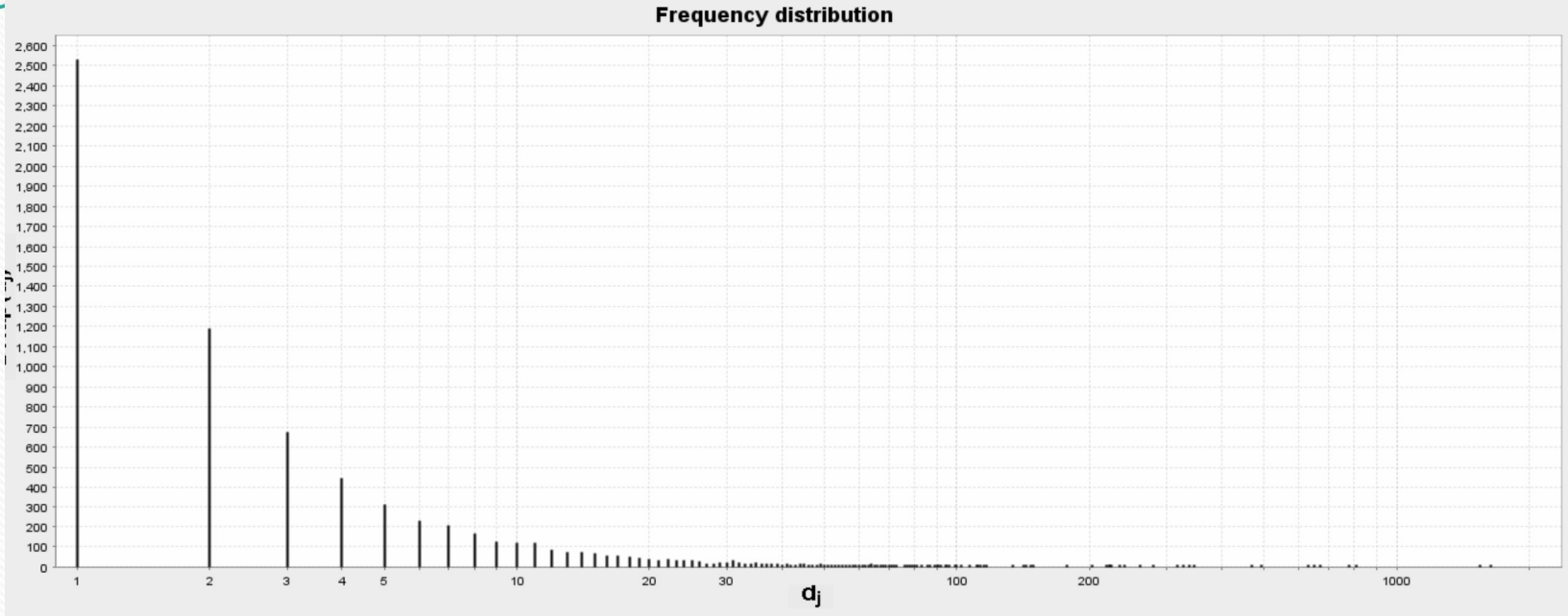
- This is not by chance: very often 2D scatter plots exhibit this density frequency distribution







The mapping



N_{DV}=126 different data densities = { 1, 2, … , 1,633 }

?

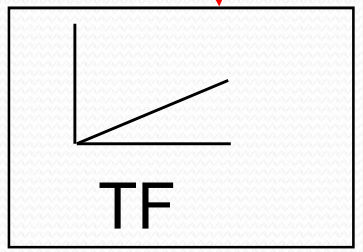
256 Color Codes = { 0,1, 2, … , 255}

Outline

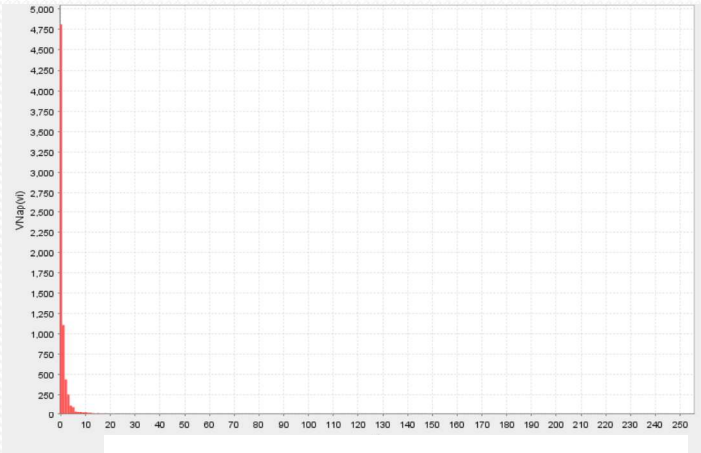
- The problem
- Available solutions
  - Linear mapping
  - Non linear mappings
- Our proposal
- Metrics and discussion
- Conclusions and future work

Linear mapping

$$ColorCode(d) = Round\left[255\,\frac{d - d_{min}}{d_{max} - d_{min}}\right]$$



- Straightforward solution
- Useless in this situation

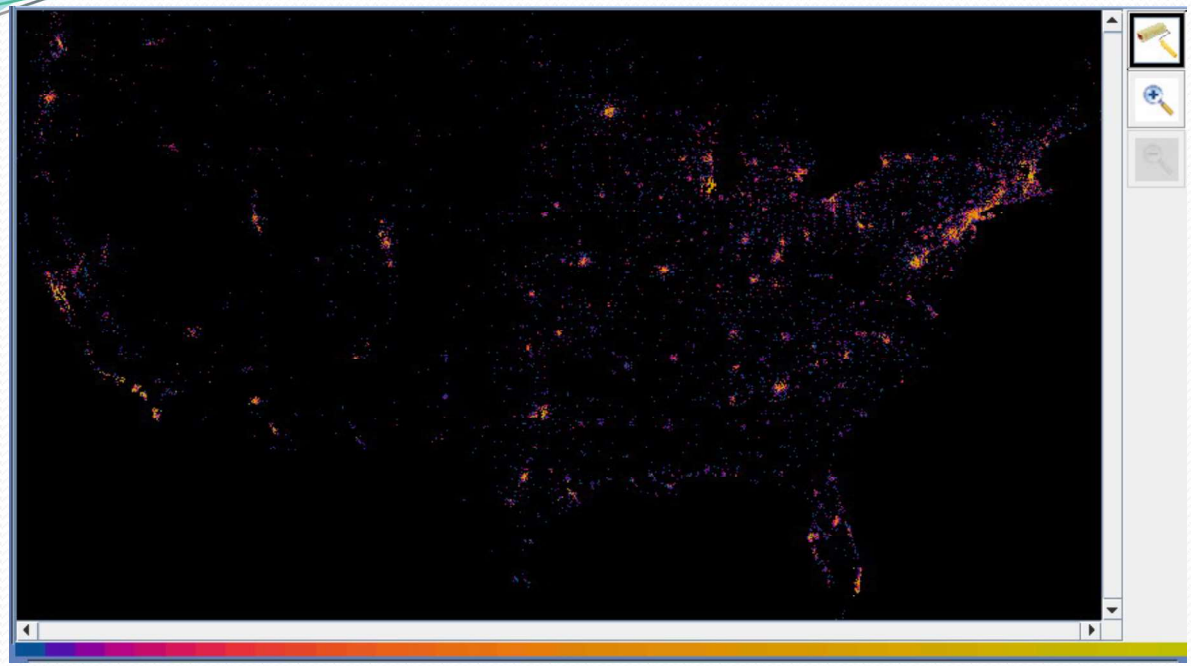


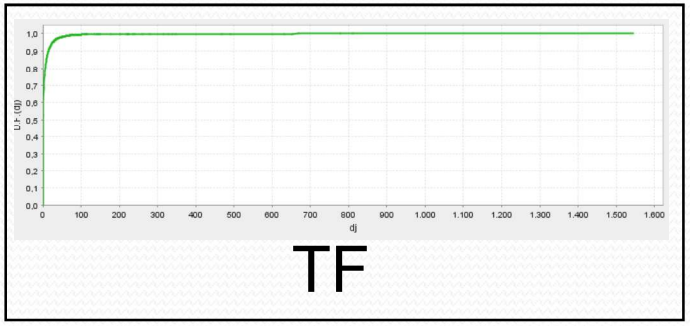
Color code frequency distribution

- Most pixels share very low color codes
- Few color codes are used (46 out of 256)
- **Different** low density values are represented by the **same** color code:
  densities in [1..10] are mapped on codes {1,2}

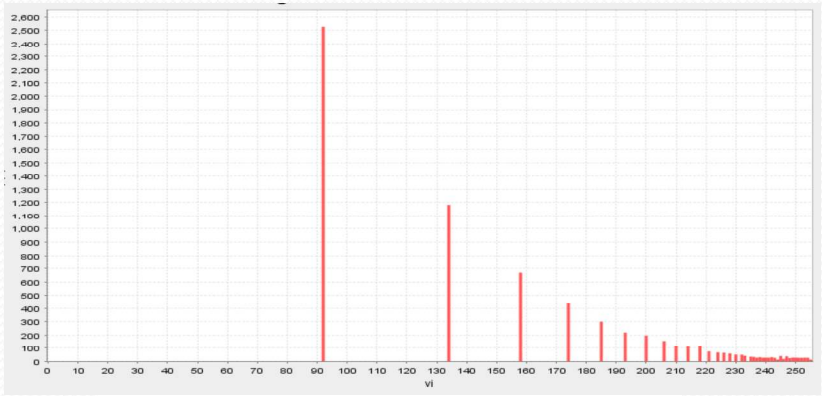
Density function mapping

$$ColorCode(d_j) = Round\left[255 \sum_{i=1}^{j} \frac{DN_{AP}(d_i)}{N_{AP}}\right]$$





- Hermann et al. [HMM00]
- Quite similar to histogram equalization
- Better than linear mapping



Color code frequency distribution

- Few color codes are used (39 out of 256)
- Lowest color code unnecessarily high
- Codes ranging only on [91..255]
- **Different** high density values are represented by the **same** color code: densities in [48..1,633] -> [250,255]

- The problem
- Available solutions
- Our proposal
  - Rationale
  - We have explored two different cases
    - One to one mapping
    - Uniform scale mapping
  - Visual comparison

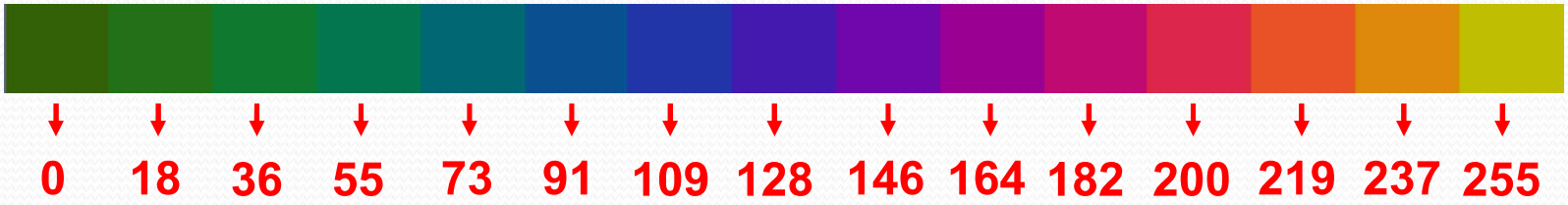
Rationale

We take into account that:

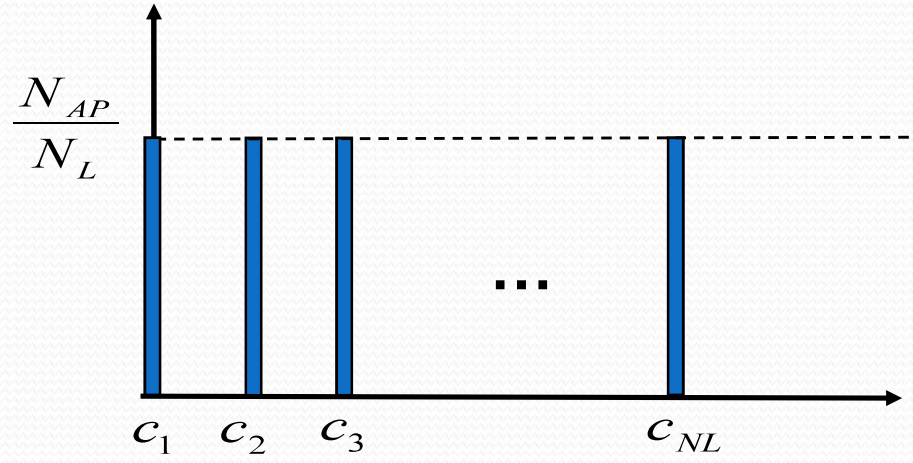
1. densities and color codes are discrete and finite
2. too close color codes are hardly distinguishable!
3. we need some objective quality metrics to validate/drive the mapping are needed

Uniform scale mapping

We use a reduced color scale, e.g. with 15 codes (N_L=15)JND!!



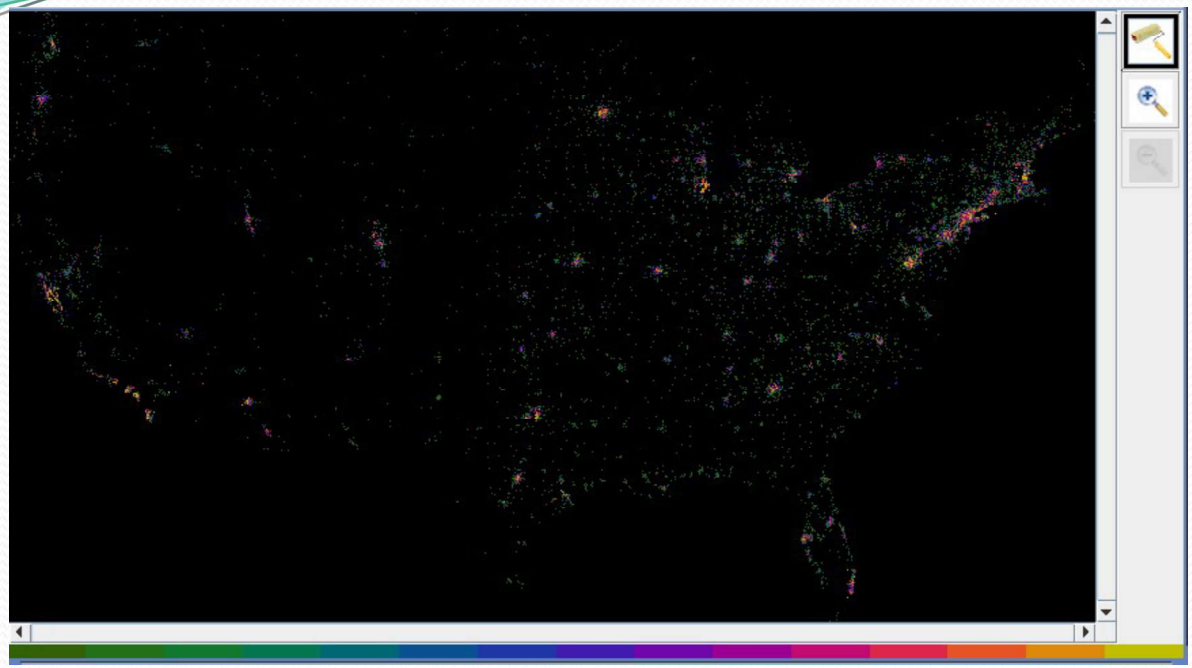
This implies that **different** density values will be **necessarily** represented by the **same** color code: to reduce the degradation the mapping is performed through an algorithm that tries to assign to each code the same number of pixels



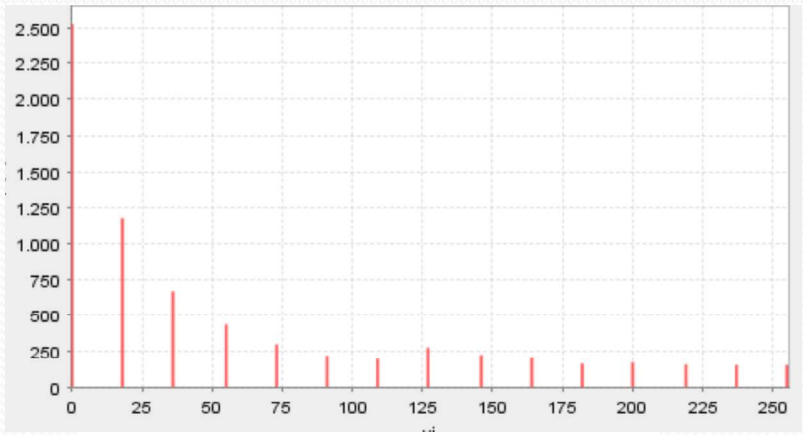
Target color code frequency distribution

uniform scale mapping

$$ColorCode(d_j) = DistributePixels$$



Because of densities are discrete the algorithm cannot ensure the N_{AP}/N_L value and through a peak analysis it minimizes the variance



Color code frequency distribution

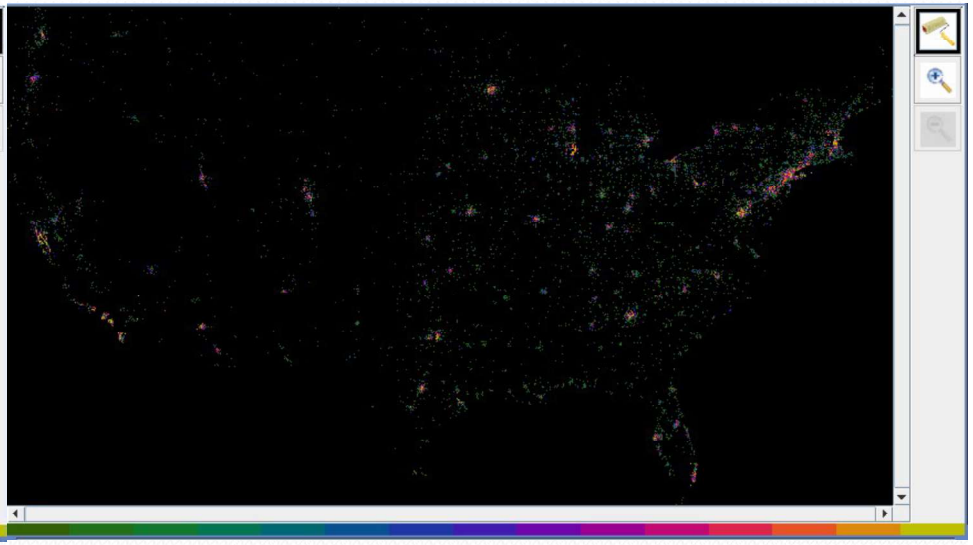
- Full color scale usage [0..255]
- All the color codes are used
- Maximum color code separation

Visual comparison

Linear mapping

Density function mapping

One to one mapping



Uniform scale mapping

Metrics

1. **C**olo**S**cale**U**sage: How many color codes is the mapping using?
2. **C**olor**S**cale**A**ctive**R**ange: What part of the color scale range is the mapping using?
3. **C**olor**S**eparation: How distinguishable are the used color codes?

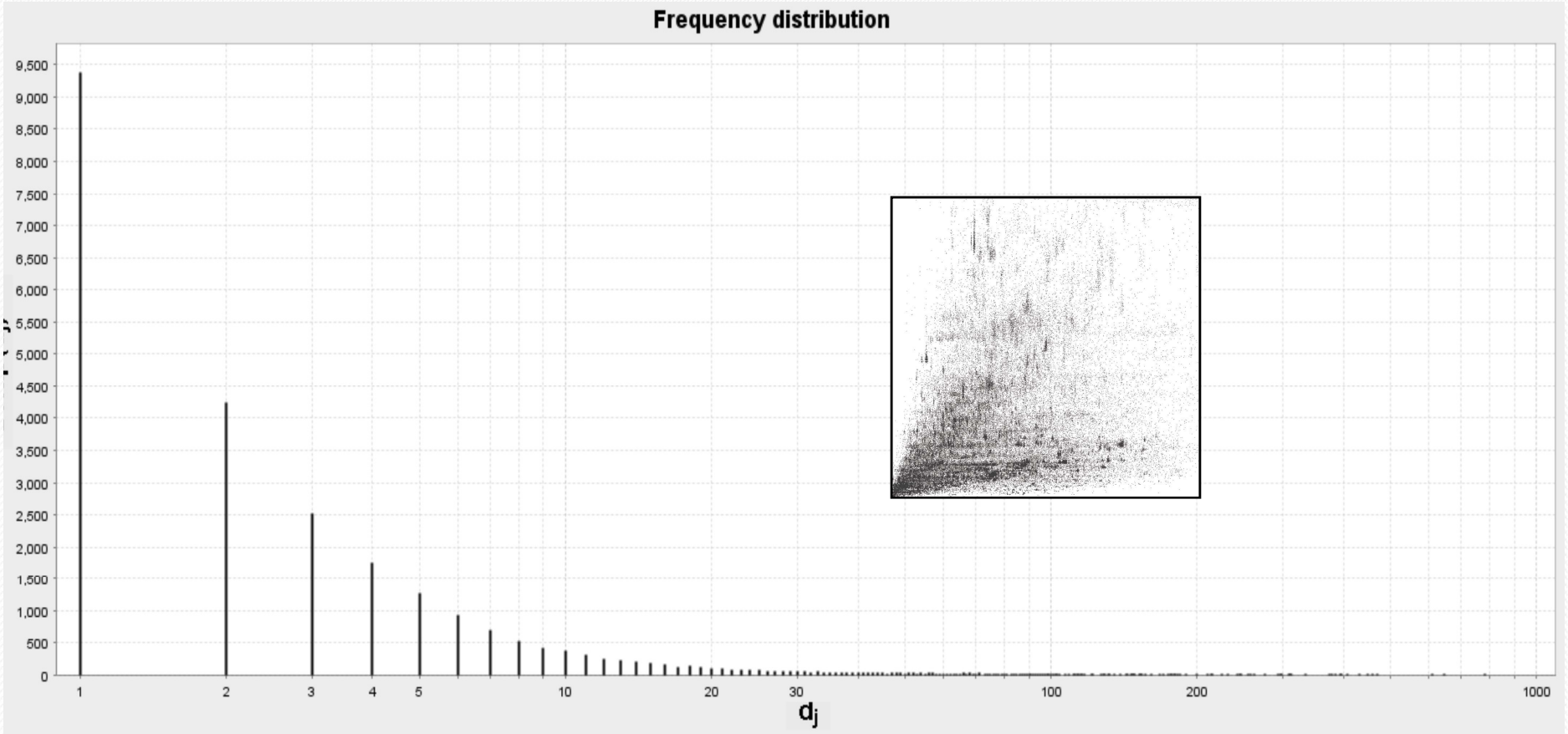
$$CSU = \frac{N_{UL}}{\min(N_{DV}, N_L)}$$

$$CsAR = \frac{C_{N_{UL}} - C_1}{C_{\max} - C_{\min}}$$

$$CS = \frac{\sum_{i=2}^{N_{UL}} (c_i - c_{i-1})}{N_{UL} - 1}$$

Postal parcels plotted by weight (x) and volume (y)

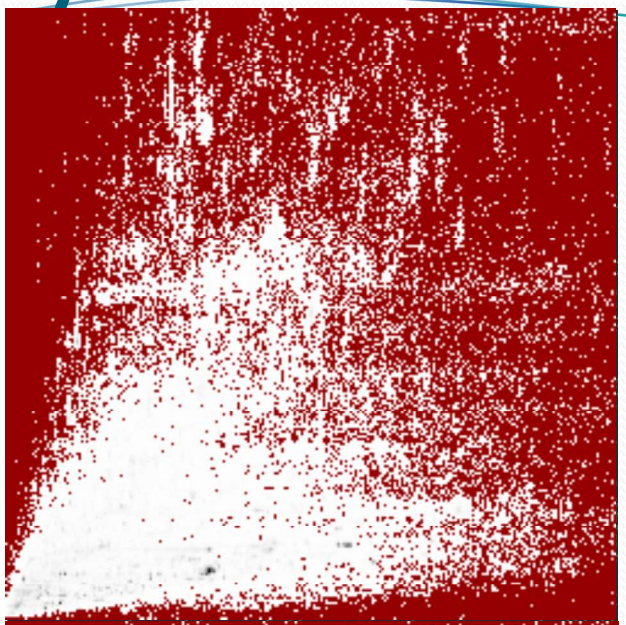
A new dataset



Grey scale

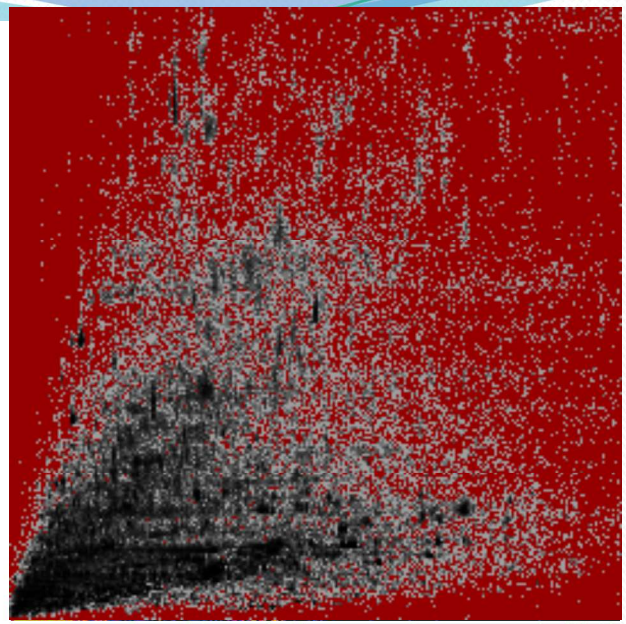
Linear

CSU=0.53
CsAR=1
CS=2.83



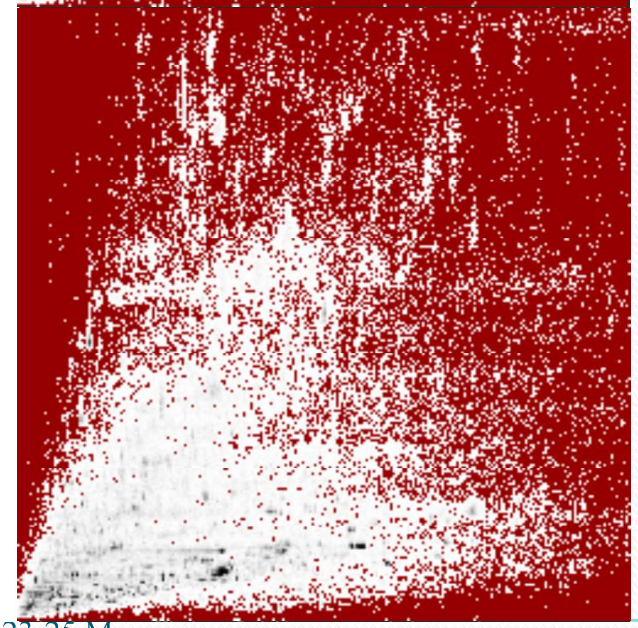
Density Function

CSU=0.18
CsAR=0.62
CS=5.23



One 2 One

CSU=1
CsAR=1
CS=1.47



Uniform color sc.

CSU=1
CsAR=1
CS=8.79