Classification-Based Browsing for Image Databases

Erwan Loisant
erwan@loisant.org

Tokyo Metropolitan University, Graduate School of Engineering

Image Search

The number of digital images is quickly increasing:

- World Wide Web
- Digital camera
- Mobile phones

Image Search

The number of digital images is quickly increasing:

- World Wide Web
- Digital camera
- Mobile phones

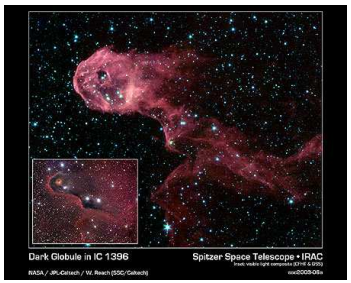
Such image collections need to be **organised** *in order to be* **retrieved** *efficiently*

Applications

- Private photograph collections (on a regular computer or a mobile device)
- Image providers catalogues
- Specialised image banks (astronomy. . .)

PHOTOGRAPH

ECOGRAPHY (MEDICAL IMAGE)



SPACE PHOTOGRAPH

SATELLITE IMAGE

Image Specificities

- *Multimedia* suggests the use of different kinds of data together...
- ...However we use it for *non-classical* data

Image Specificities

- *Multimedia* suggests the use of different kinds of data together...
- ...However we use it for *non-classical* data

Multimedia Data

- Can not be normalised
- Variable information density
- Information depends on the observer

Problem & Goal

Problem:

- Images are difficult to describe.
- Most people do not want to take the time to annotate their images.
- Proposals based on retrieval are difficult for users.

Problem & Goal

Problem:

- Images are difficult to describe.
- Most people do not want to take the time to annotate their images.
- Proposals based on retrieval are difficult for users.

Goal

Build a proposal of **image search by navigation** to complement image retrieval. It should be:

- **Easy to use:** the user interaction should be mainly mouse clicks.
- **Fast and responsive:** the user should not have to wait for the results.
- **Scalable:** it should be applicable to large collections.

Part I

State of The Art

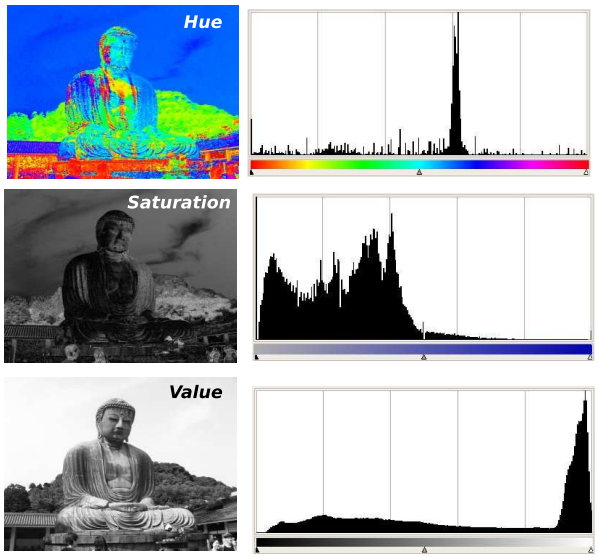
Outline

Outline

Content Information vs. Annotations

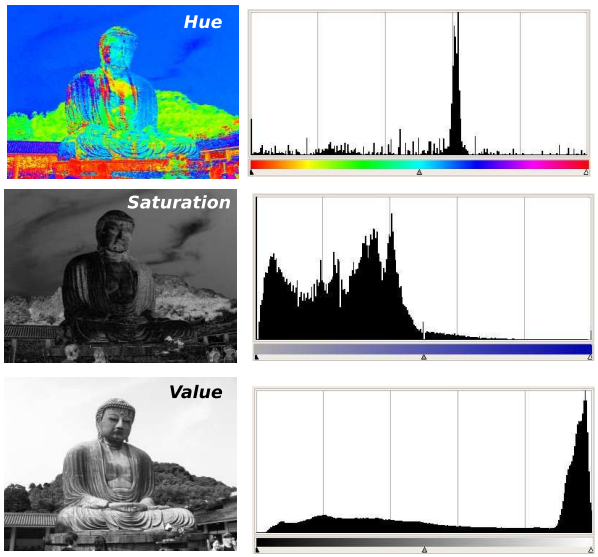
Content Information vs. Annotations

Content Information



Content Information vs. Annotations

Content Information

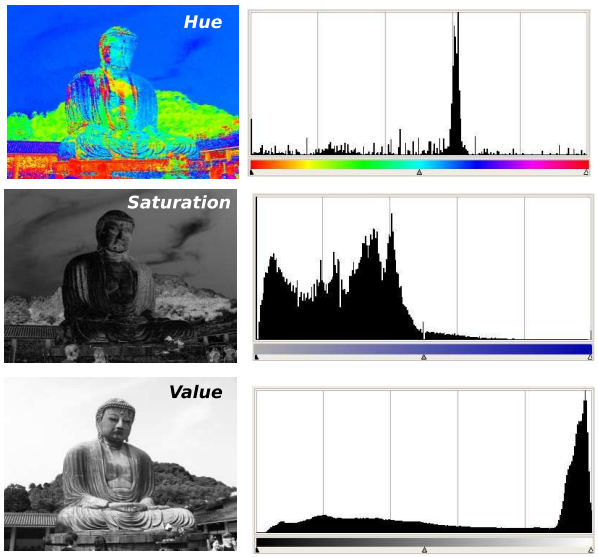


Annotations

- Buddha
- Kamakura
- Japan
- Monument

Content Information vs. Annotations

Content Information



◯ Objective, automatic
× Semantic gap

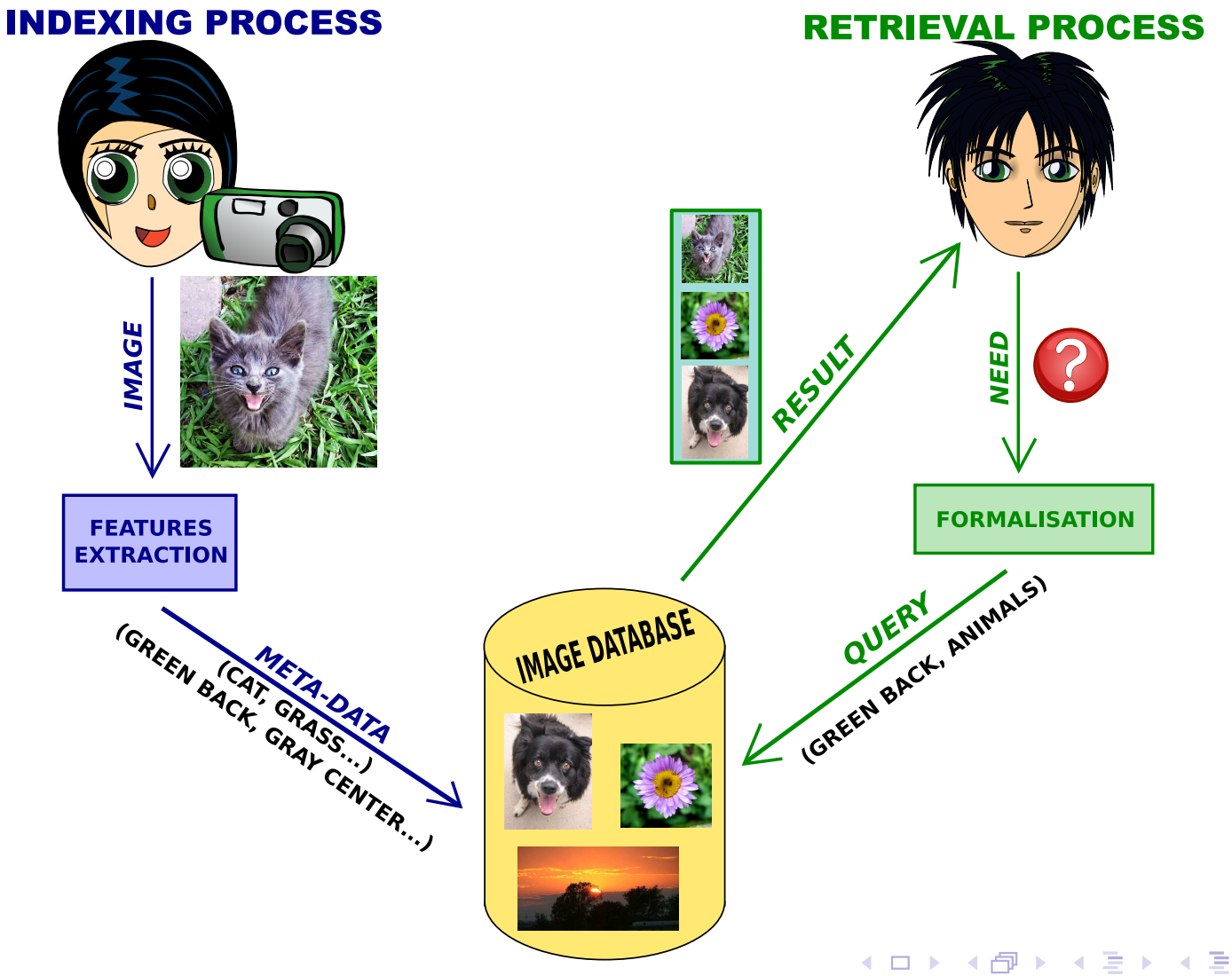
Annotations

- Buddha
- Kamakura
- Japan
- Monument

◯ Good semantics
× Subjective, manual
(→ cost)

Outline

Image Retrieval in a Nutshell



Defining the Search

Different kinds of search methods (not an exhaustive list):

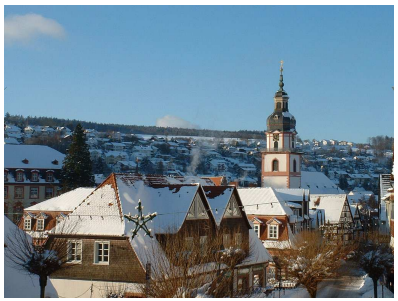
- Retrieval
  - Formal queries
  - Similarity search
  - Queries with relevance feedback
- Navigation

Formal Queries (Chabot, by Ögle & Stonebraker)

```
RETRIEVE (q.all) FROM q IN photocd_bib WHERE
    q.shoot_date$>$"Jan 1 2000" and
    MeetsCriteria("MostlyBlue",q.histogram) and
    MeetsCriteria("SomeWhite",q.histogram)
```

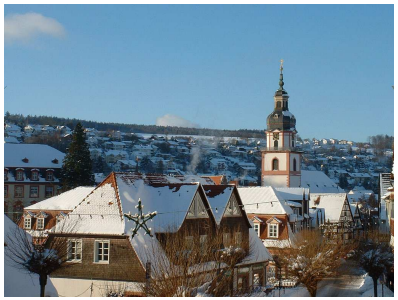
Formal Queries (Chabot, by Ögle & Stonebraker)

```
RETRIEVE (q.all) FROM q IN photocd_bib WHERE
    q.shoot_date$>$"Jan 1 2000" and
    MeetsCriteria("MostlyBlue",q.histogram) and
    MeetsCriteria("SomeWhite",q.histogram)
```

Formal Queries (Chabot, by Ögle & Stonebraker)

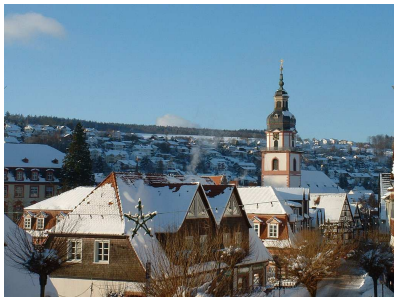
```
RETRIEVE (q.all) FROM q IN photocd_bib WHERE
    q.shoot_date$>$"Jan 1 2000" and
    MeetsCriteria("MostlyBlue",q.histogram) and
    MeetsCriteria("SomeWhite",q.histogram)
```

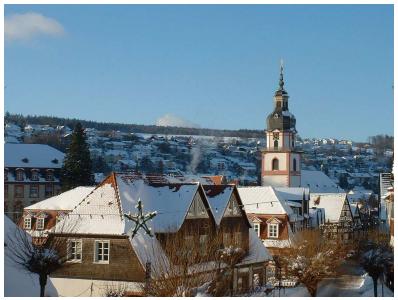
Problem: who is able to write this kind of query?

Similarity Search - Sample Image

Similarity Search - Sample Image



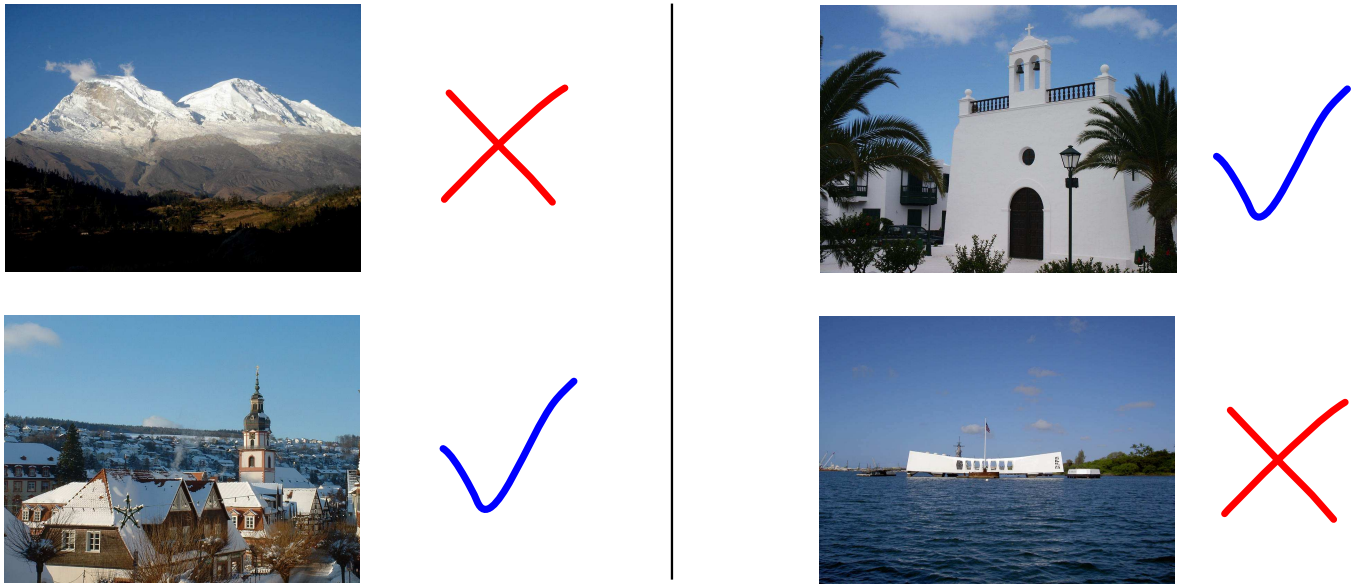
Similarity Search - Sample Image



Problem: usually, the user has no sample image.

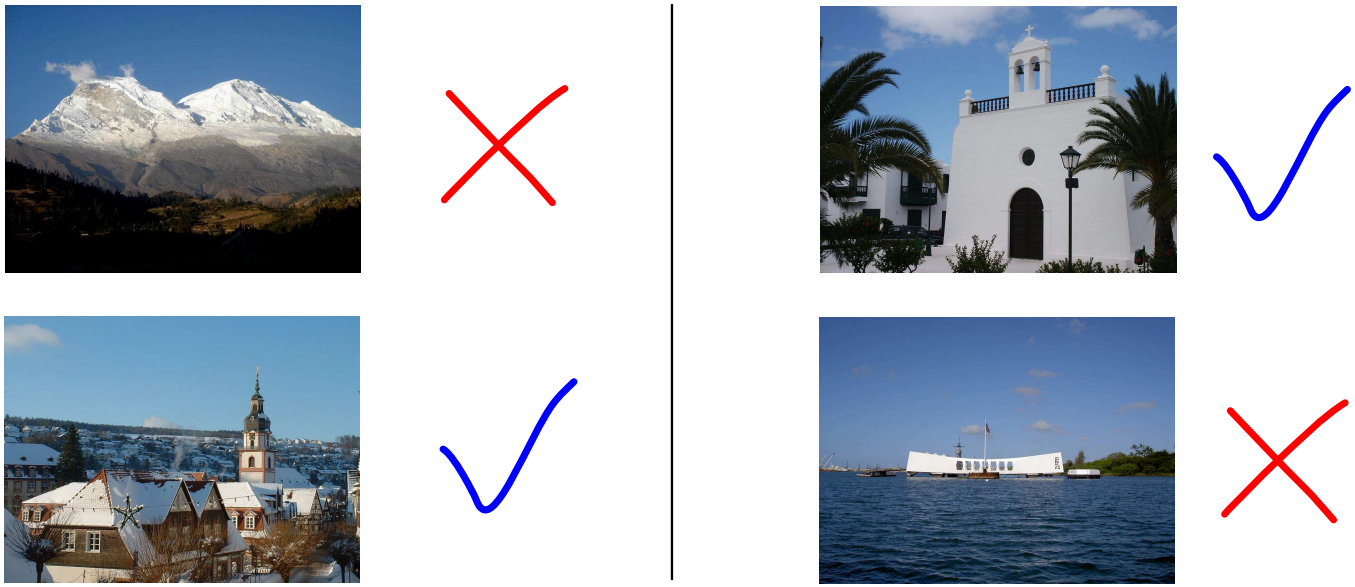
Relevance Feedback

The user marks "good" images and "bad" images.



Relevance Feedback

The user marks "good" images and "bad" images.



Problem: write a query that accepts all examples and reject all counter-example is known to be **NP-complete** .

Image Retrieval (Reminder)

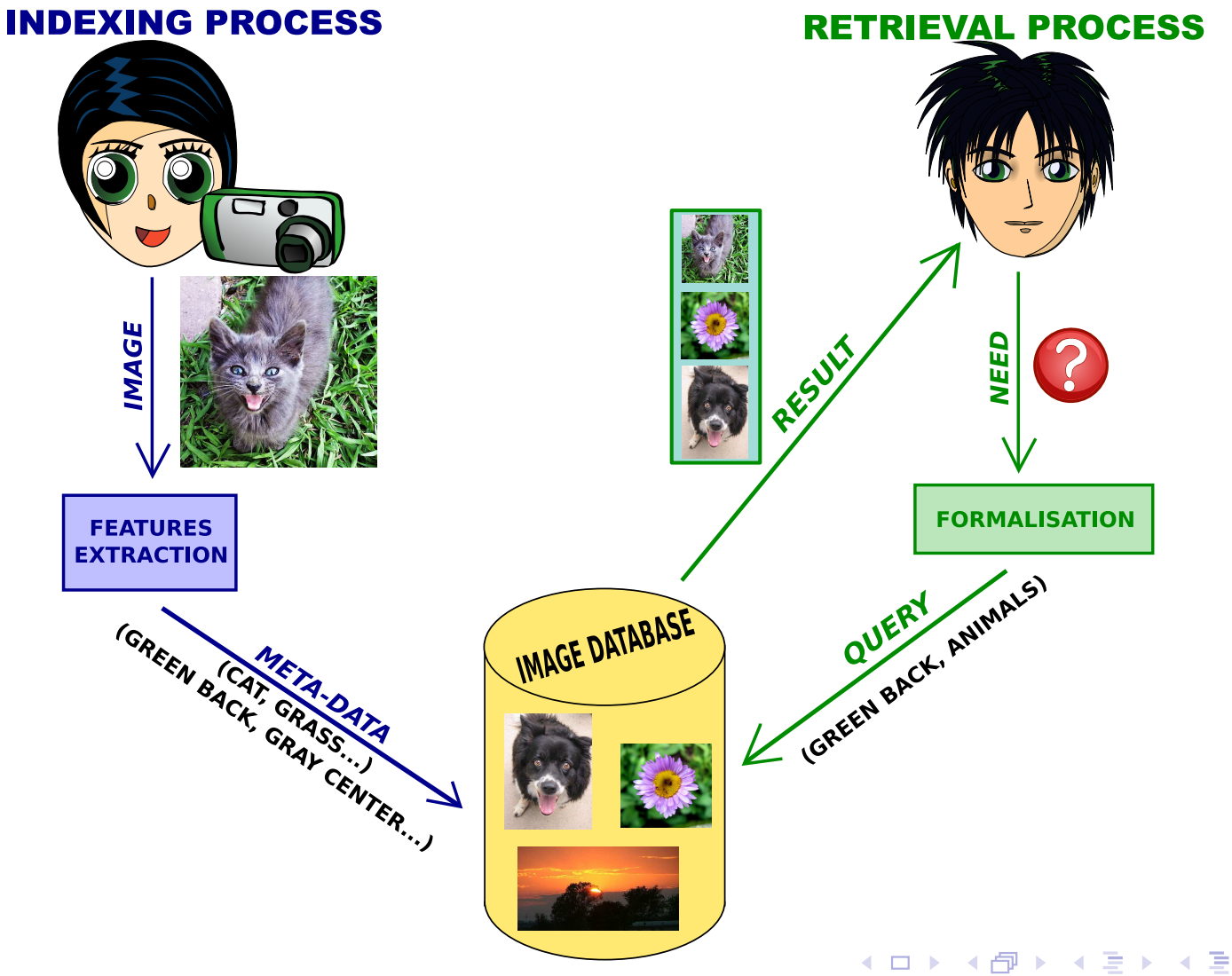
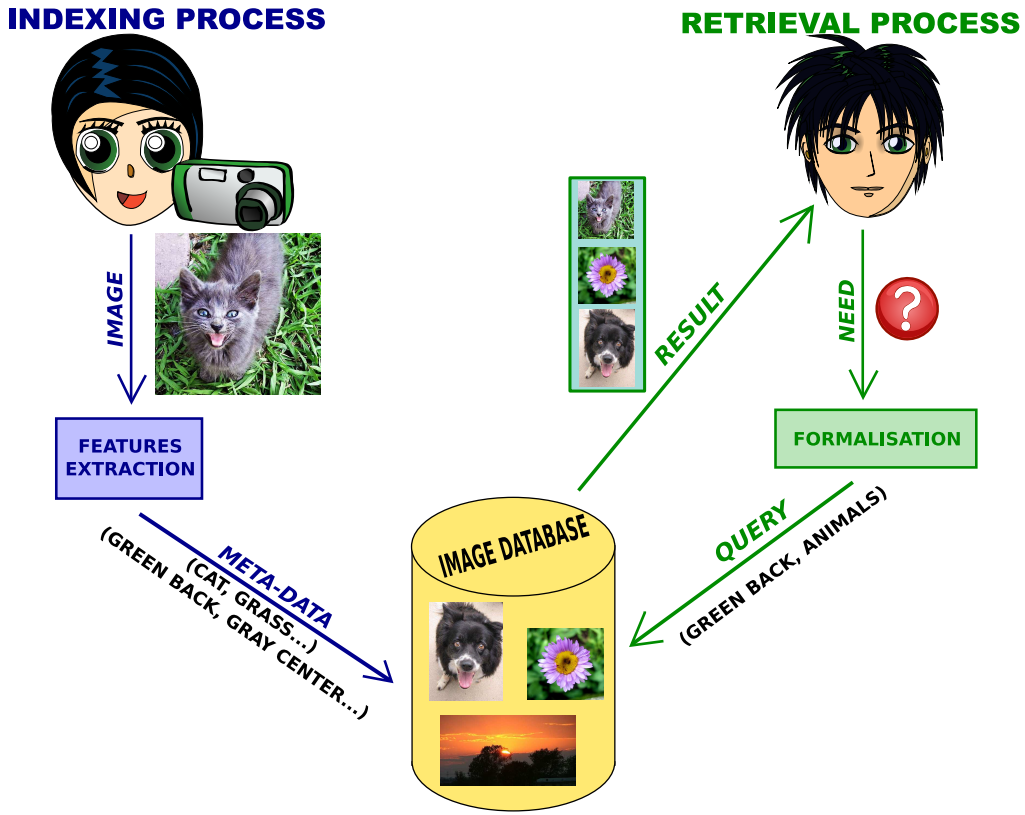
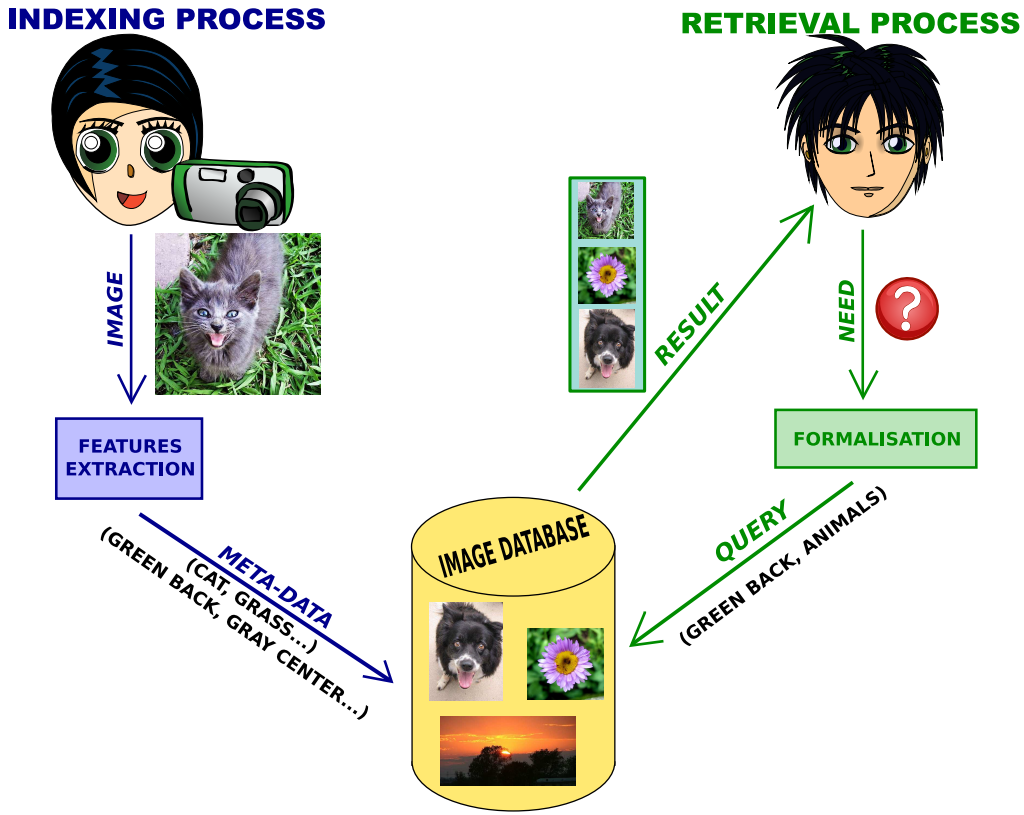


Image Retrieval (Reminder)



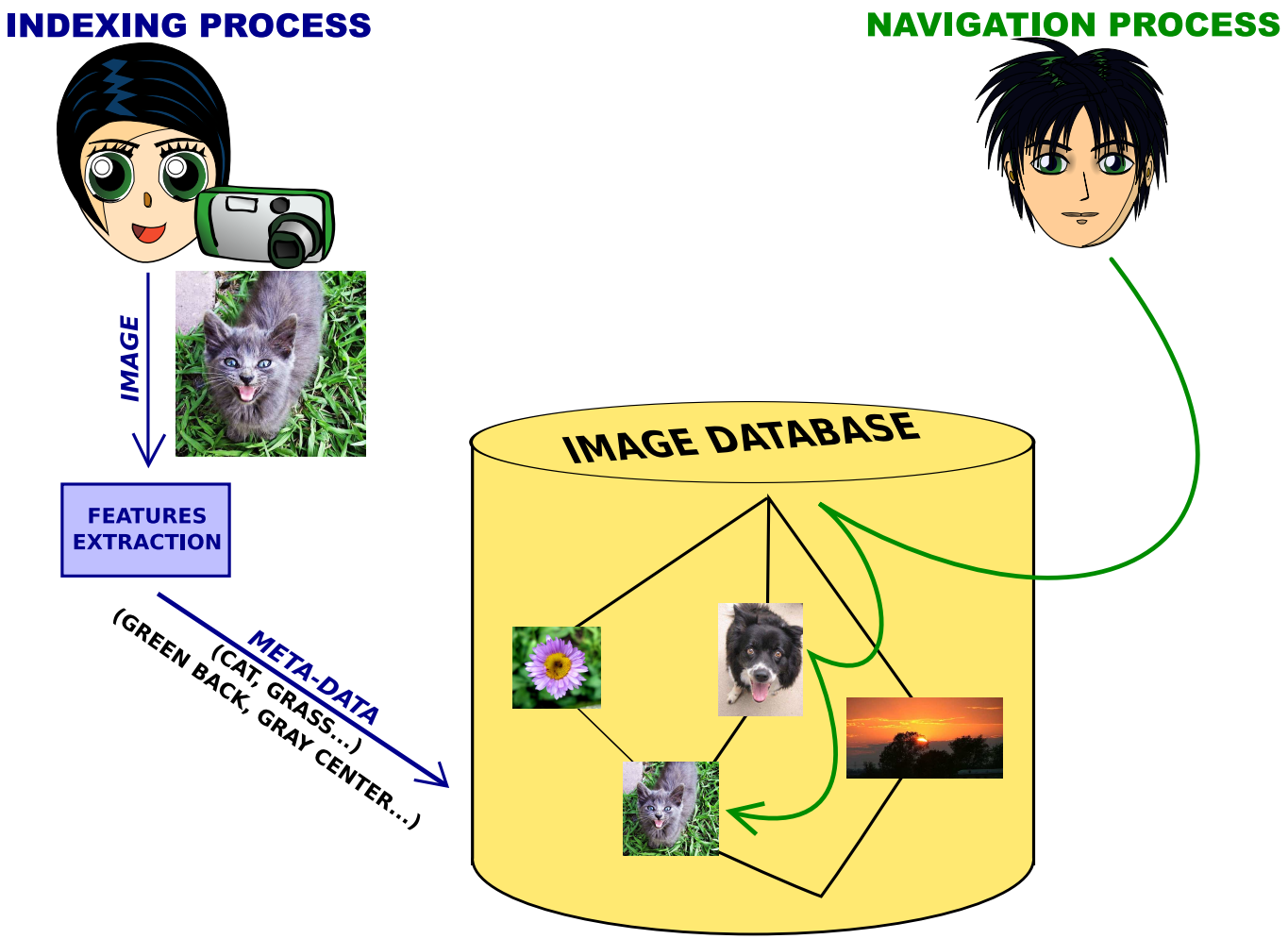
- *PROBLEM 1*: Formalisation is not trivial.
- *PROBLEM 2*: Query processing can be costly.

Image Retrieval (Reminder)

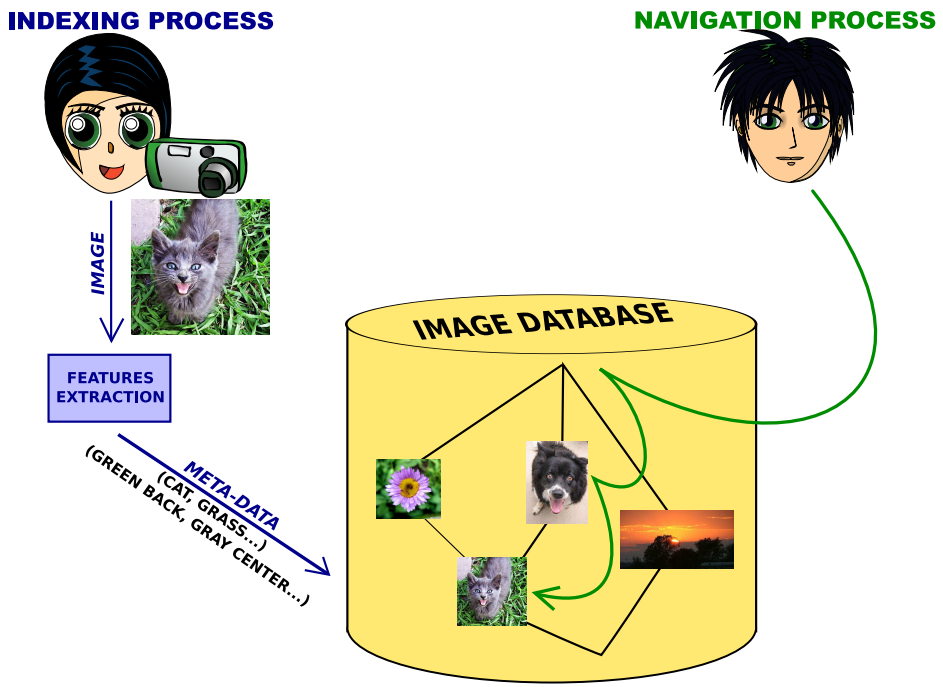


- *Assumption*: There are **few insertions**.
- *Assumption*: There are **a lot of searches** (at best $O(log(n))$).

Why Navigation?



Why Navigation?



The cost of insertion is higher, but:

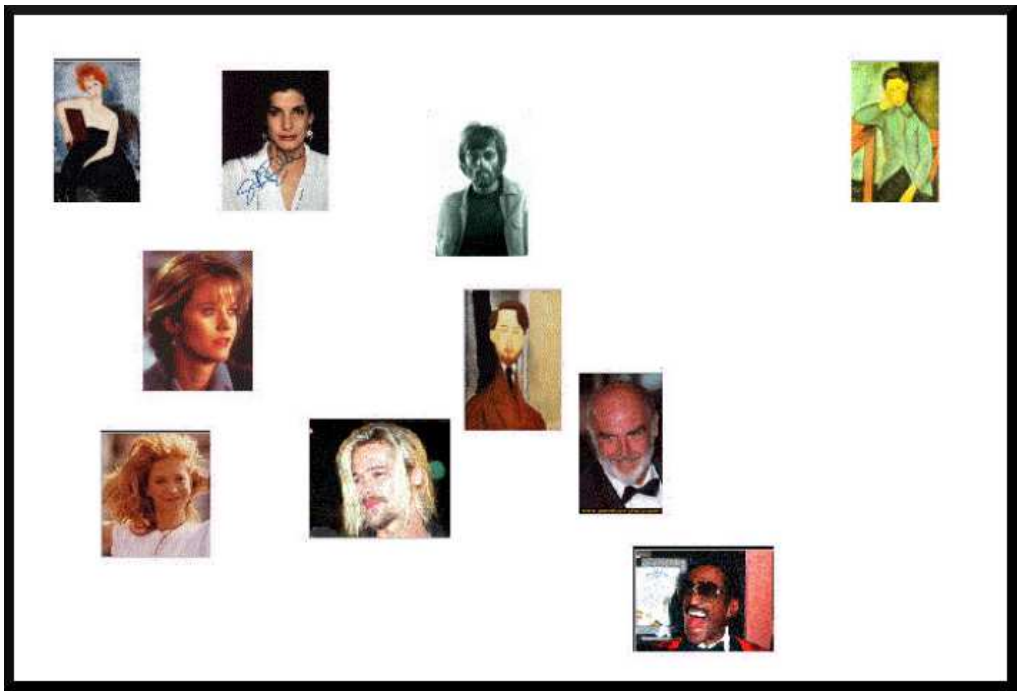
- *NO FORMALISATION PHASE*: Easy for the user.
- *NO QUERY PROCESS PHASE*: Quick and responsive.

Previous Work linked to Navigation

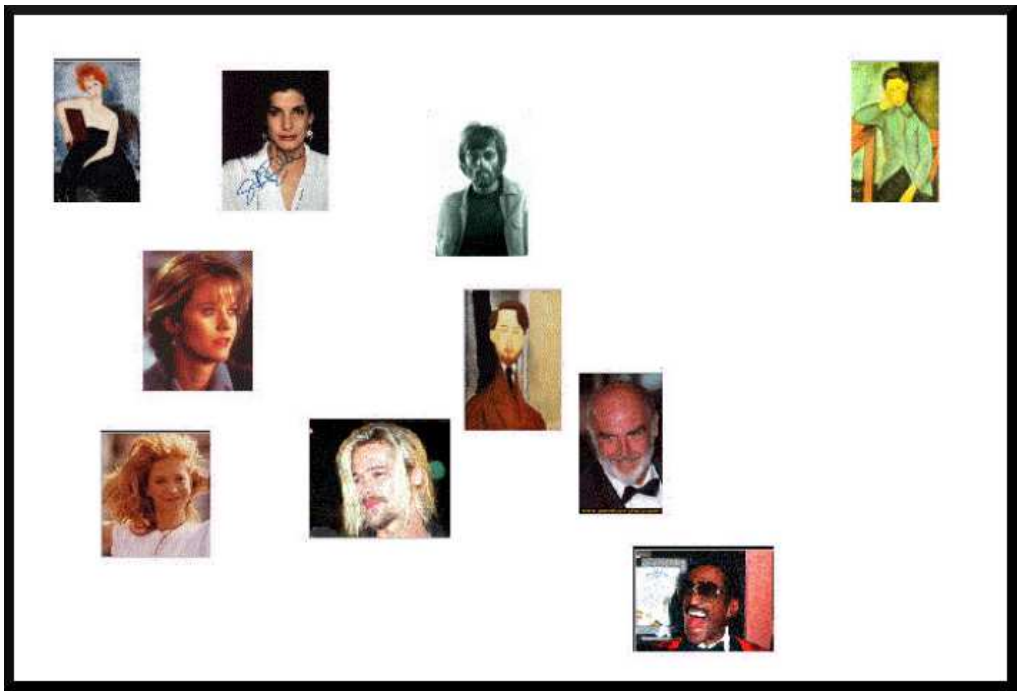
There are few proposals on navigation. It can still be compared to:

- Show the images on a map, according to location only (Geobloggers. . .): *Limited to place.*
- Combine the place and the time space into *events*: *Use only two dimensions is limited.*
- Organise the images on a virtual space according to content similarity (Santini's El Niño. . .): *Based on retrieval → long processing time.*

A Modigliani's Painting with Similar Images (Santini)



A Modigliani's Painting with Similar Images (Santini)



PROBLEM: Based on retrieval, with a costly query process.

Part II

Efficient Structures for Navigating an Image Collecti

Outline

Outline

Goal

Provide a system for navigation-based image search.

- Define a metamodel to describe images.
- Define a navigation structure for the user.

Goal

Provide a system for navigation-based image search.

- Define a metamodel to describe images.
- Define a navigation structure for the user.

Objectives

- Metamodel:
  - **Low cost:** the metrics should be fast to process.
  - **Match human perception:** the metrics should have a meaning for a human observer.
- Navigation:
  - **Easy to use:** the user interaction should be mainly mouse clicks.
  - **Fast and responsive:** the user should not have to wait for the results.

Work Domain

We restrict our study to "classical" photographs:

- Rectangular, with no transparency channel
- Colour images
- Excluding specific images, such as:
  - Medical images
  - Satellite images
  - Photographs of the space

Meta-Model Overview

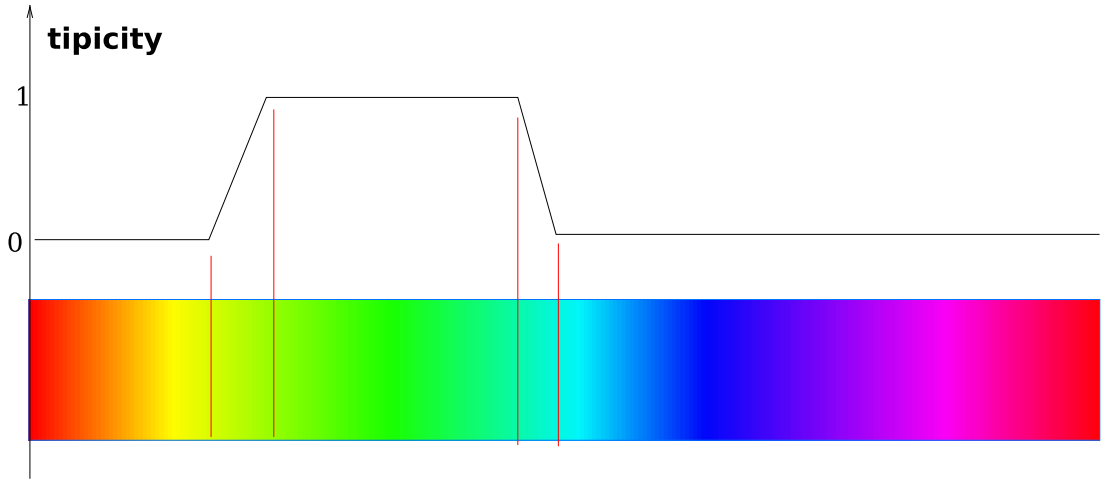
Our model is based on content information:

- Dominant colours for each part of a syntactical division (similar to the use of histograms, a fast to calculate metric)
- General geometrical measures (orientation, elongation and size)

Merits

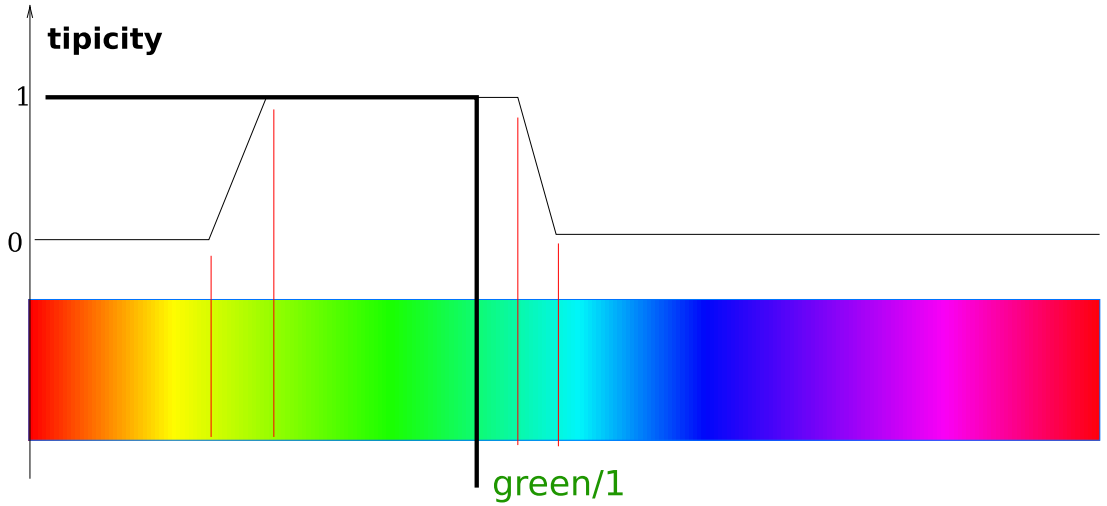
- **Fuzzy logic** $\longrightarrow$ good representation of the human perception.
- **Syntactical division** $\longrightarrow$ more efficient than segmentation but still offers a good semantic separation.

Fuzzy Logic (the green label)



A fuzzy subset is defined by a *trapeze*. A fuzzy label is a description associated to a tipicity.

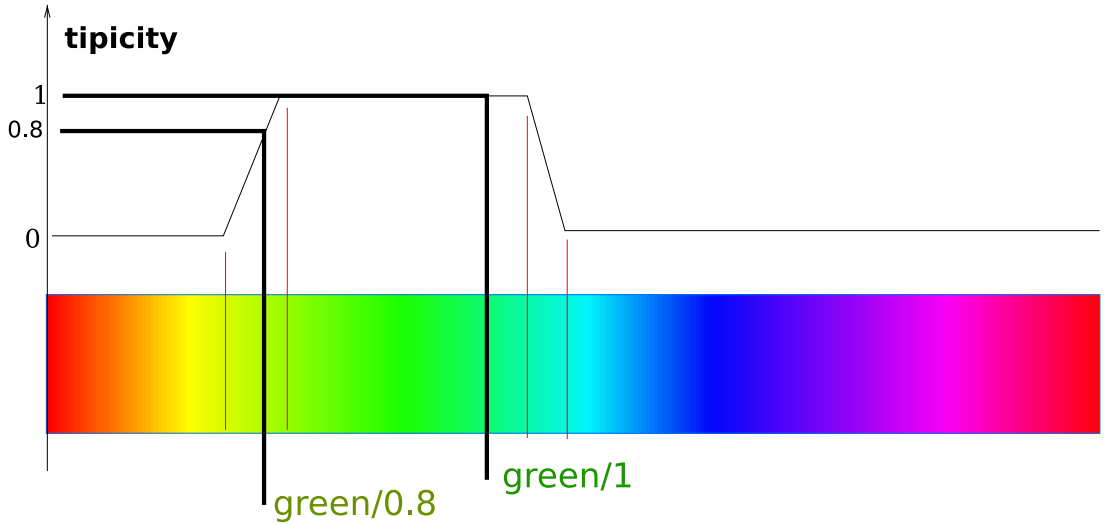
Fuzzy Logic (the green label)



A fuzzy subset is defined by a *trapeze*. A fuzzy label is a description associated to a tipicity.

- $green/1 \longrightarrow$ a green pixel.

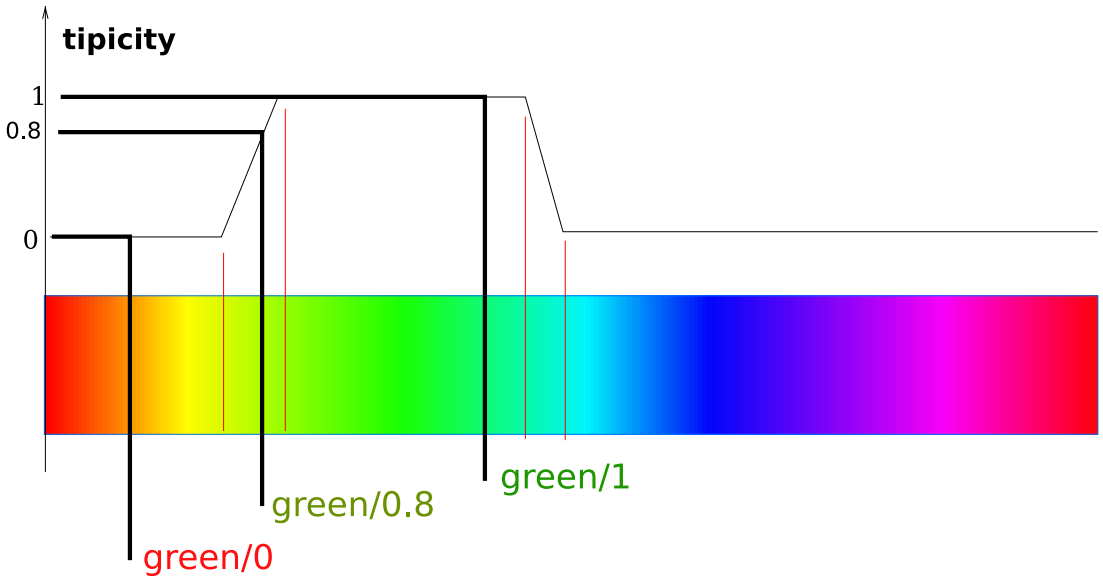
Fuzzy Logic (the green label)



A fuzzy subset is defined by a *trapeze*. A fuzzy label is a description associated to a tipicity.

- $green/1 \rightarrow$ a green pixel.
- $green/0.8 \rightarrow$ an *almost* green pixel.

Fuzzy Logic (the green label)



A fuzzy subset is defined by a *trapeze*. A fuzzy label is a description associated to a tipicity.

- $green/1 \longrightarrow$ a green pixel.
- $green/0.8 \longrightarrow$ an *almost* green pixel.
- $green/0 \longrightarrow$ a *not* green pixel.

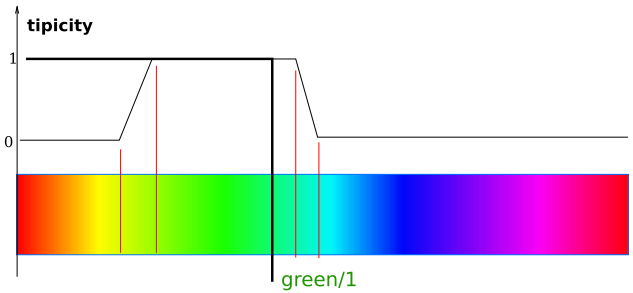
Segmentation of the Colour Space

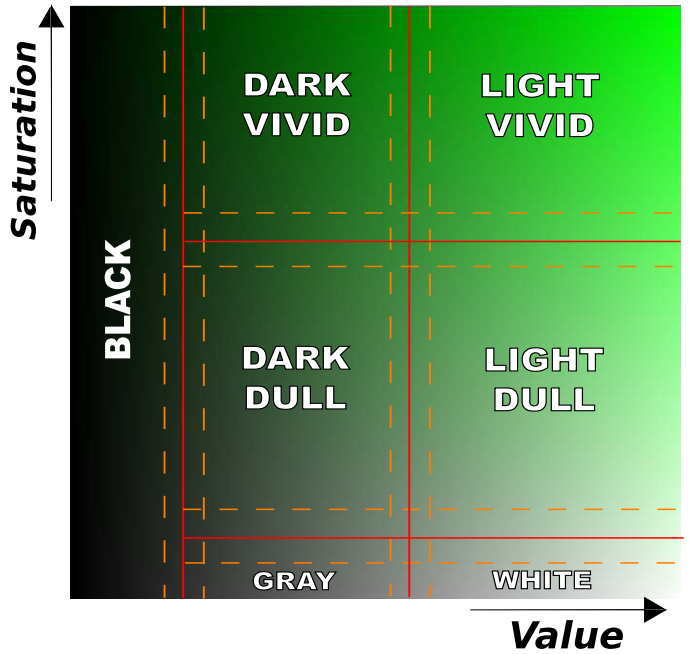
- We chose the HSV (Hue, Saturation, Value) model, each property makes sense for human observers
- 7 fuzzy subsets are defined on hue
- For a given hue, the saturation/value plan is divided into black, white, grey and 4 coloured parts

Examples of labels

- Black
- Light Desaturated Red *(i.e. "pink")*
- Dark Saturated Blue

Saturation and Value



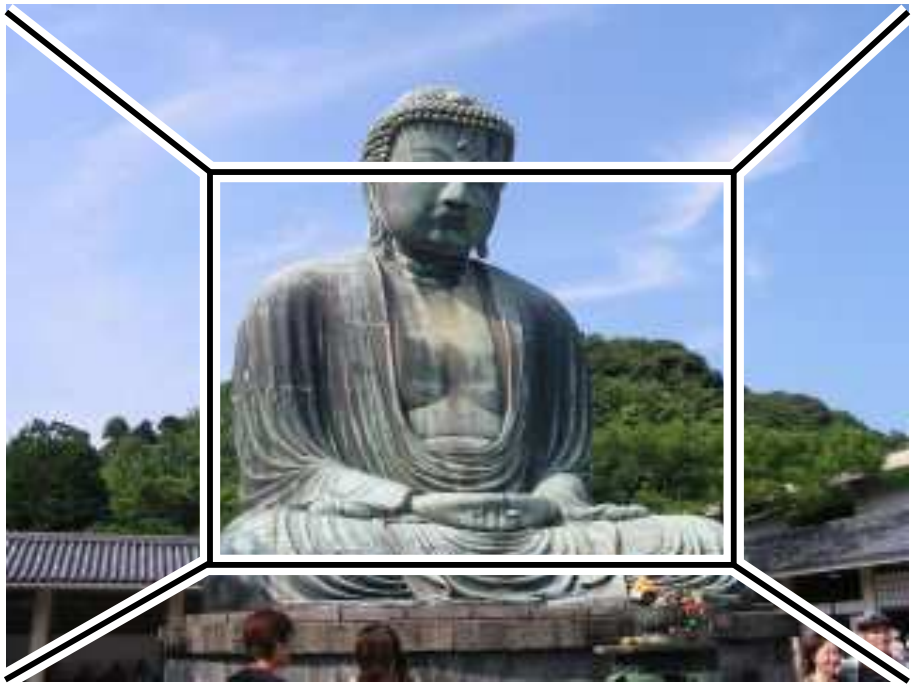


Heuristic Segmentation

For performance reasons, we use a *heuristic segmentation* rather than a real segmentation algorithm.

Heuristic Segmentation

For performance reasons, we use a *heuristic segmentation* rather than a real segmentation algorithm.

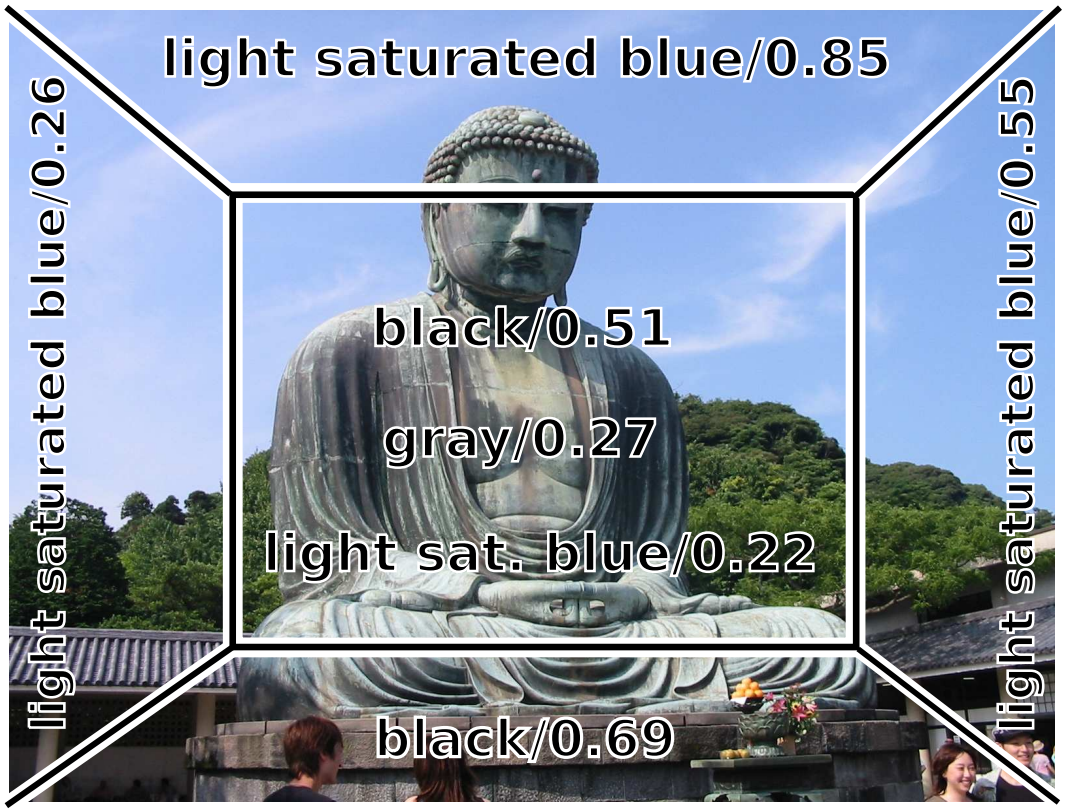


Used Properties

- $\mathcal{D}_{area} = \{tiny, small, medium, large, huge\}$
- $\mathcal{D}_{orientation} = \{portrait, square, landscape\}$
- $\mathcal{D}_{elongation} = \{none, standard, panoramic, elongated\}$
- $\mathcal{D}_{colour} = \mathcal{C} \times \{top, bottom, left, right, centre\}$

$\mathcal{C}$ being the set of colours defined as:
$\mathcal{C} = \{white, gray, black\}$
$\bigcup(\{red, orange, \ldots, cyan, magenta\} \times \{vivid, light\} \times \{dark, light\})$

Properties of the Big Buddha Photograph



+ landscape/1, standard/1, large/0.70, medium/0.30

Conclusions on the Metamodel

The metamodel we defined:

- **Has a low cost:** histograms are fast to calculate.
- **Match human perception:** the fuzzy division of the colour space and the syntactical division of the images gives good semantic measures.

Galois' Lattice

A *lattice* is a graph structure:

- Directed
- With no cycle
- Featuring a unique minimal node (no parents) and a unique maximal node (no child)

Galois' Lattice

A Galois' lattice is a special kind of lattice built from a binary relationship.

- Each node is a couple (images, properties) (I, P)
- If (I_2, P_2) is a child node of (I_1, P_1), then $I_2 \subset I_1$ and $P_1 \subset P_2$

Why choose Galois' Lattices?

Benefits for the user:

- The links between the nodes are semantic
  $\longrightarrow$ **understandable for the user**.
- the browsing is optimal (no calculation)
  $\longrightarrow$ **fast and responsive**.
- The user can go back very easily
  $\longrightarrow$ **helps to correct the users' mistakes.**

Benefits for the system:

- Can be used both as index and retrieval.
- Intrinsically a multi-dimensional classification technique.
- Insensitive to correlations.

Note: the high complexity of the construction algorithm limits us in the collection size (not adaptated to the world wide web).

Example (1/2)

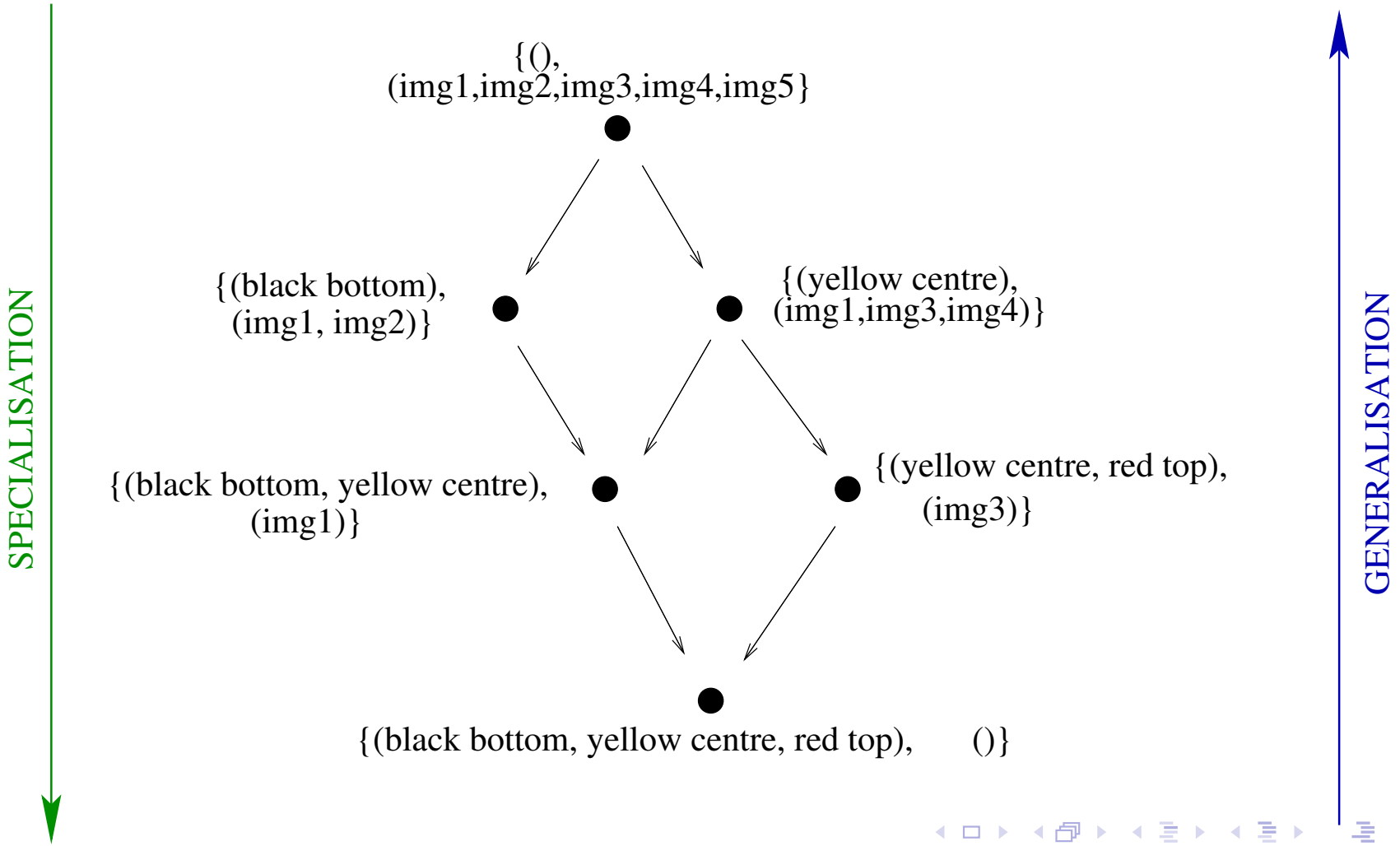
Consider the following images/properties relationship:

$$\begin{aligned} R = \{ \quad & (img_1, blackbottom), (img_1, yellowcentre), \\ & (img_2, blackbottom), \\ & (img_3, yellowcentre), (img3, redtop), \\ & (img4, yellowcentre) \qquad \} \end{aligned}$$

	*img*1	*img*2	*img*3	*img*4
blackbottom	1	1	0	0
yellowcentre	1	0	1	1
redtop	0	0	1	0

Example (2/2)

The associated Galois' lattice is:



A Bigger Lattice

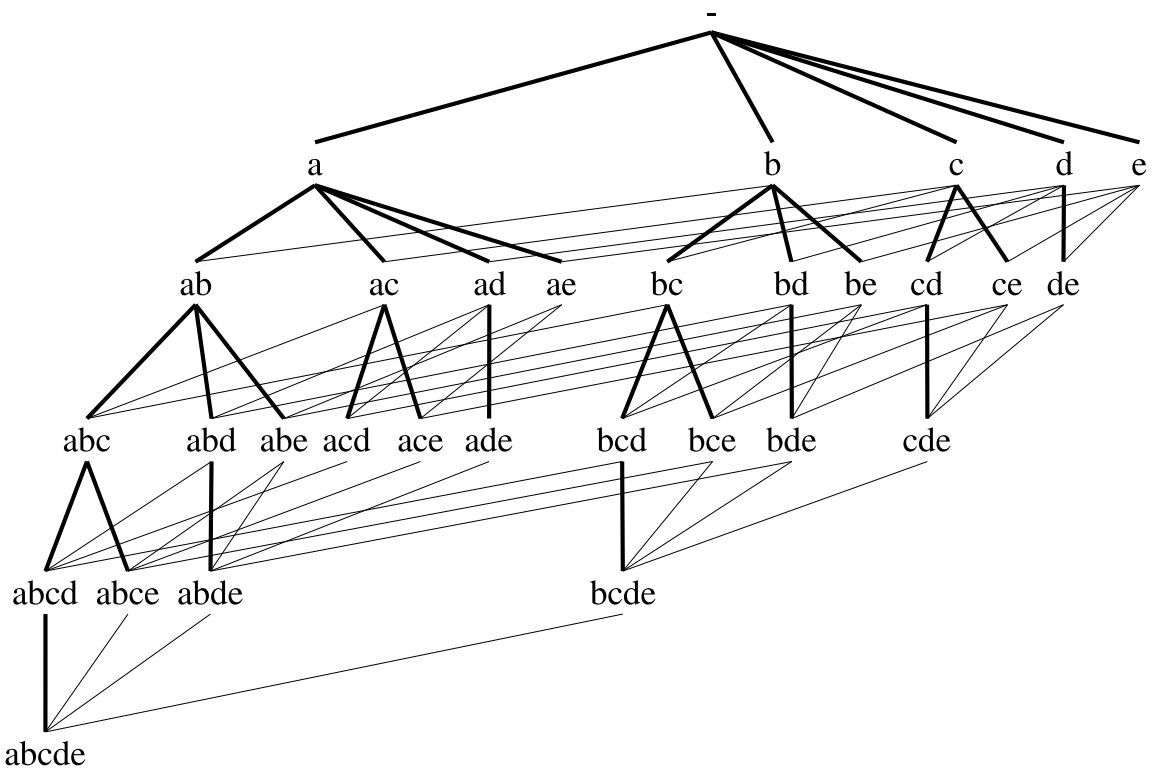


Figure: A bigger Galois' lattice

Constructing a Galois' Lattice

- We need an incremental algorithm
  - Godin et. al published such an algorithm in 1995

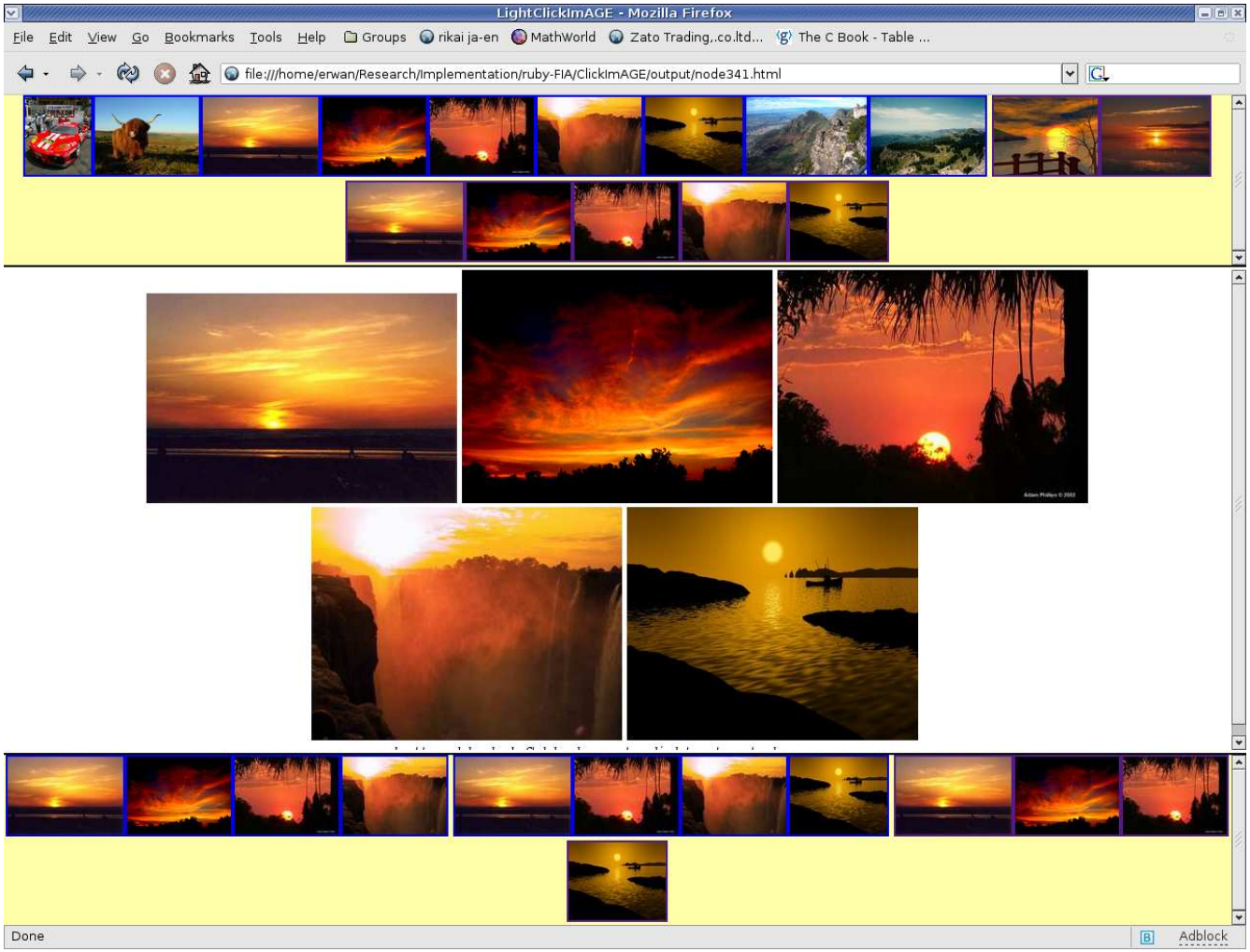
Complexity:

- Time complexity *in the worst case*: exponential (e^n) with respect to the total number of images
- Empirical average time complexity:
  - Add an element: $o(n)$ with respect to the initial number of images
  - From scratch: $o(n^2)$ with respect to the total number of images
  - For large dataset: the limit is known to be $n.log(n)$.

Hypermedia Representation (User-Interface)

- The lattice is built on an images-descriptions binary relationship;
- For each node, a detailed representation and a quick representation is defined;
- Each node is represented as an hypermedia page (XHTML)
- The user can browse the resulting set of web pages to find the image he/she is looking for

User Interface

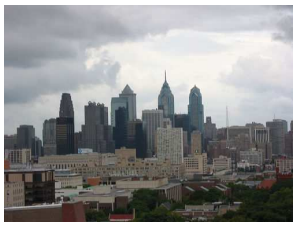


Data Set

About 5,000 images, extracted from Flickr (http://flickr.com) using some of the most popular tags: *art*, *city*, *flower*, *party*, *sunsets*, *travel*, *birthday*, *dog*, *snow*, and *nature*.

Data Set

About 5,000 images, extracted from Flickr (http://flickr.com) using some of the most popular tags: *art*, *city*, *flower*, *party*, *sunsets*, *travel*, *birthday*, *dog*, *snow*, and *nature*.



CITY

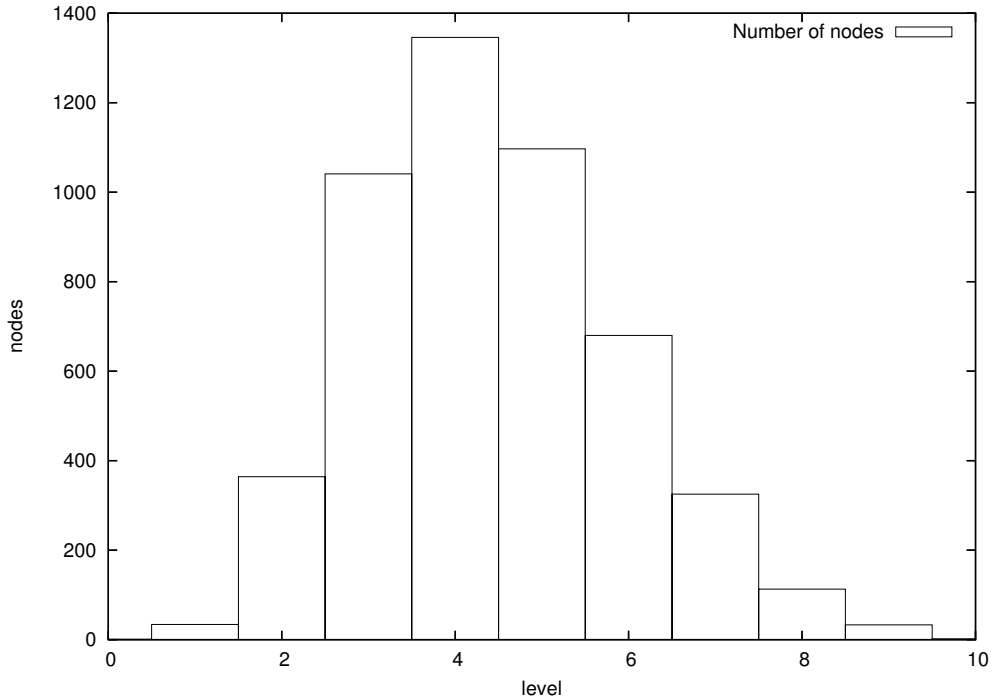
SNOW

FLOWER

DOG

SUNSET

Statistics on the Resulting Lattice



Number of nodes by level (level n = n properties).

Conclusion

Several advantages:

- **Easy to use**, since there is no query formalisation phase.
- **Very fast to navigate** through a graph structure that has been computed off-line (no calculation, optimal).
- It is insensitive to correlations.
- It helps to correct the users' mistakes very easily.

Conclusion

Several advantages:

- **Easy to use**, since there is no query formalisation phase.
- **Very fast to navigate** through a graph structure that has been computed off-line (no calculation, optimal).
- It is insensitive to correlations.
- It helps to correct the users' mistakes very easily.

But drawbacks:

- The time complexity of the construction algorithm is $O(n^2)$ $\longrightarrow$ scalability problem.
- A local explosion of the number of nodes may appear.

Outline

Problem Definition

- Regarding the complexity ($O(n^2)$) of the construction algorithm, to build a lattice with more than about 10,000 images would be very long.
- Additionnally, such a lattice would be too big to be browsed comfortably by the user.
- However, there are databases much bigger than this (several millions for Corbis)

Problem Definition

- Regarding the complexity ($O(n^2)$) of the construction algorithm, to build a lattice with more than about 10,000 images would be very long.
- Additionnally, such a lattice would be too big to be browsed comfortably by the user.
- However, there are databases much bigger than this (several millions for Corbis)

Problem

Considering the scalability limits of Galois' lattices, can these graphs be useful for such large collections?

A Galois' Lattice of Clusters

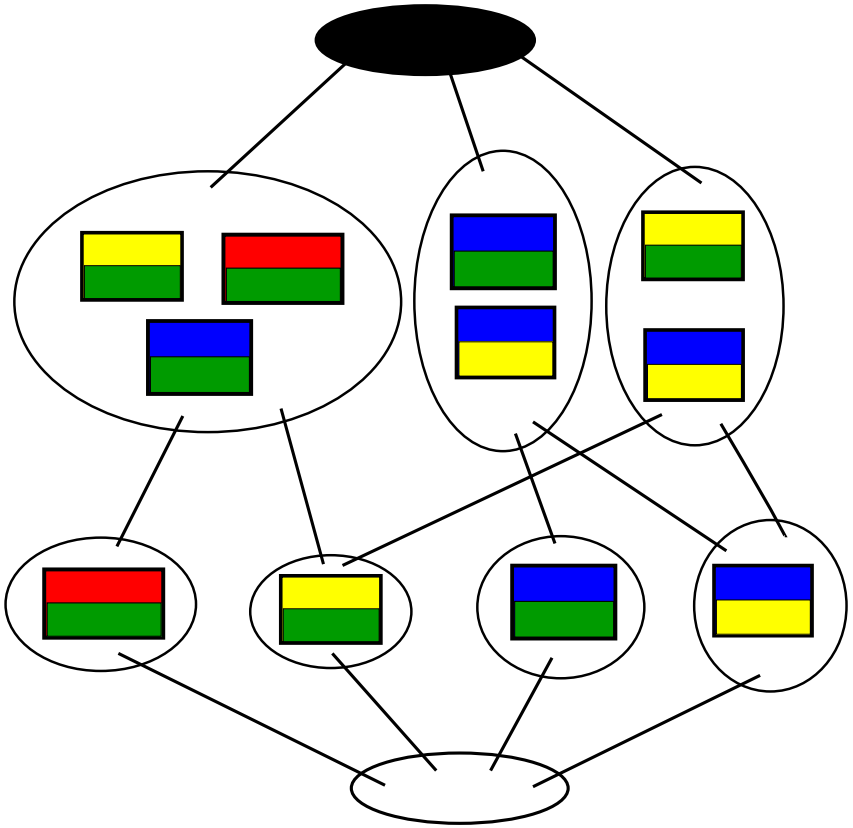
Associate a clustering technique to our Galois' lattice proposal.

1. The meta-data of images are calculated.
2. Images are organised into clusters of similar images (using a linear time complexity algorithm)
3. A Galois' lattice is built on the clusters
4. Navigation becomes on *two-levels*
   - inter-cluster navigation
   - intra-cluster navigation

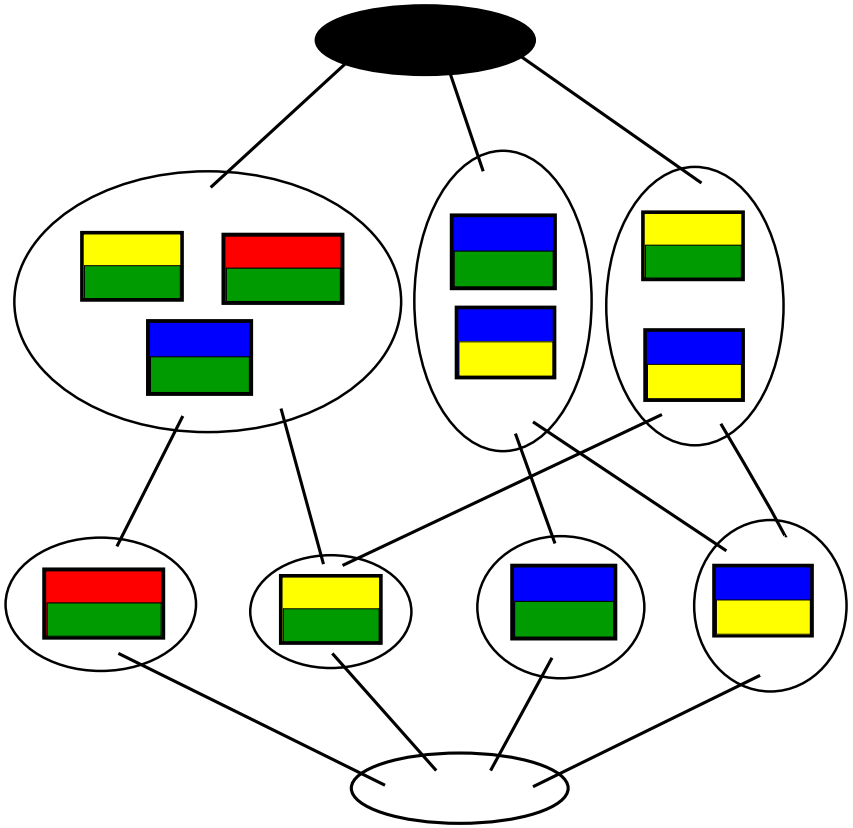
Benefits

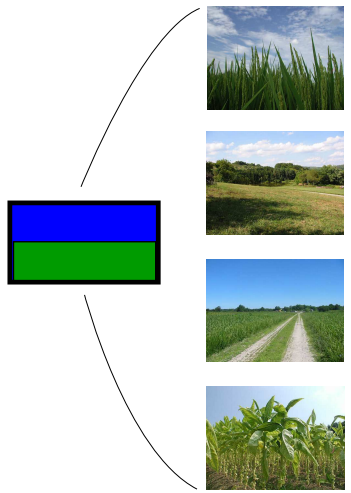
- Time complexity: square $O(n^2)$ $\rightarrow$ **linear** $O(n + \left(\frac{n}{C}\right)^2)$ (n for the clustering and $\left(\frac{n}{C}\right)^2$) for the lattice construction - with C number of clusters, $\left(\frac{n}{C}\right)^2 \ll n$.
- Images grouped by clusters $\rightarrow$ easier for the user to browse the collection.

A Navigation on Two Levels



A Navigation on Two Levels





The Clustering Technique

1. Based on the techniques of SaintEtiQ
2. Using the fuzzy descriptions detailed in previous part
   - *e.g.* $1.0/$*vivid dark green* $+ 0.8/$*vivid light blue*
3. A cluster hierarchy is built incrementally
   - from the most general (the *root*)
   - to the more specific (the *leaves*)
4. Each cluster is labelled using a notation similar to images
5. Linear time complexity ($O(n)$, results from SaintEtiQ)

The SaintEtiQ projet

- SEQ is a clustering project from Nantes University (France)
- It uses fuzzy logic to generate summaries of data (in the general case)
- Main authors:
  - Noureddine Mouaddib
  - Guillaume Raschia
  - Régis Saint-Paul
- G. Raschia, N. Mouaddib, A fuzzy set-based approach to database summarization, *Int. Journal of Fuzzy Sets and Systems*, 129(2):137-162, July 2002
- http://www.simulation.fr/seq/ *(English)*

What is a Clustering?

Given a sequential presentation of tuples (images) and their associated descriptions, the main goals of concept formation are:

1. identifying clusters that group the tuples into categories;
2. defining an intentional description (i.e., a summary) that corresponds to each category;
3. organising these summaries into a hierarchy.

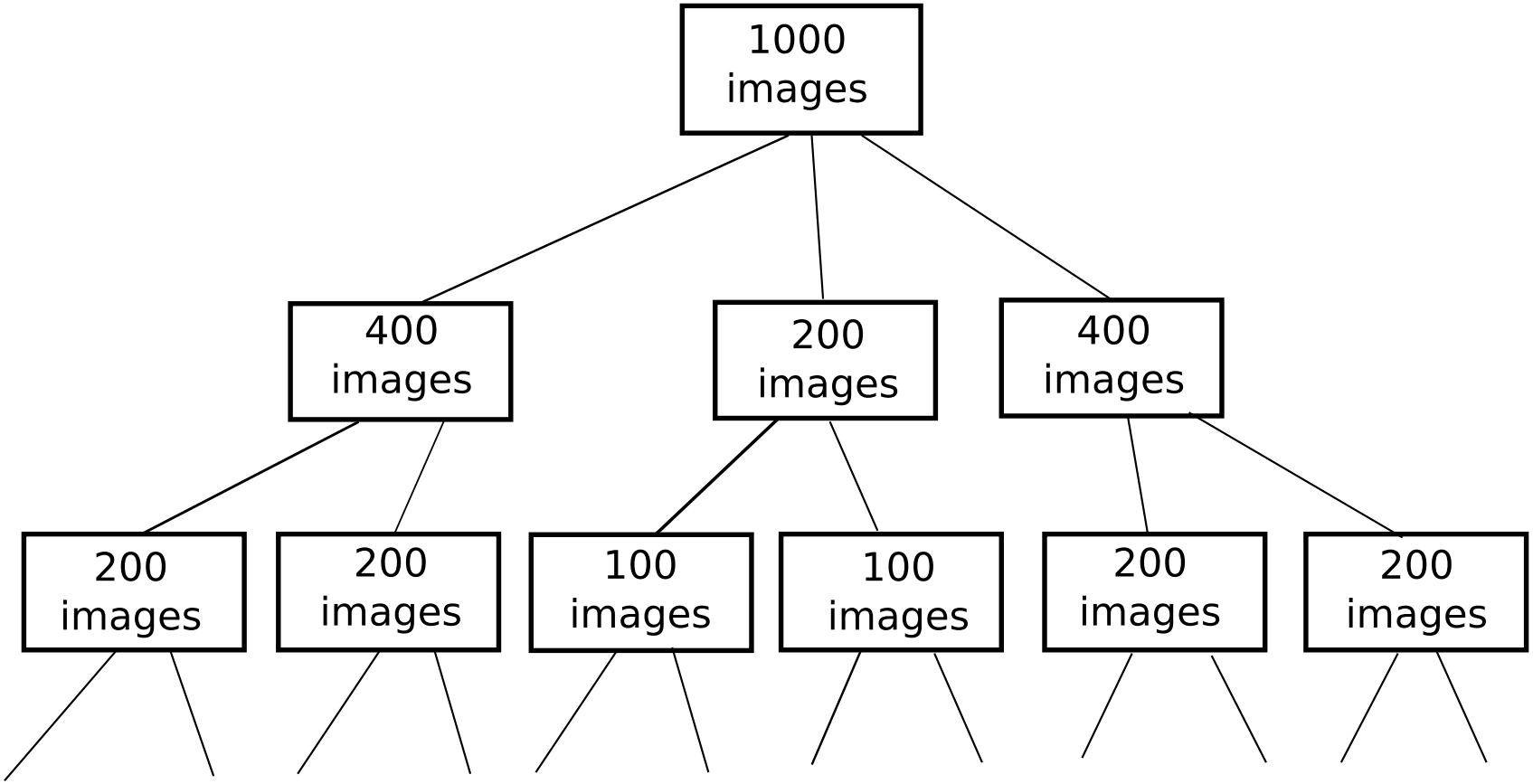
Additionally, in our case the learning is incremental.

Incremental Algorithm

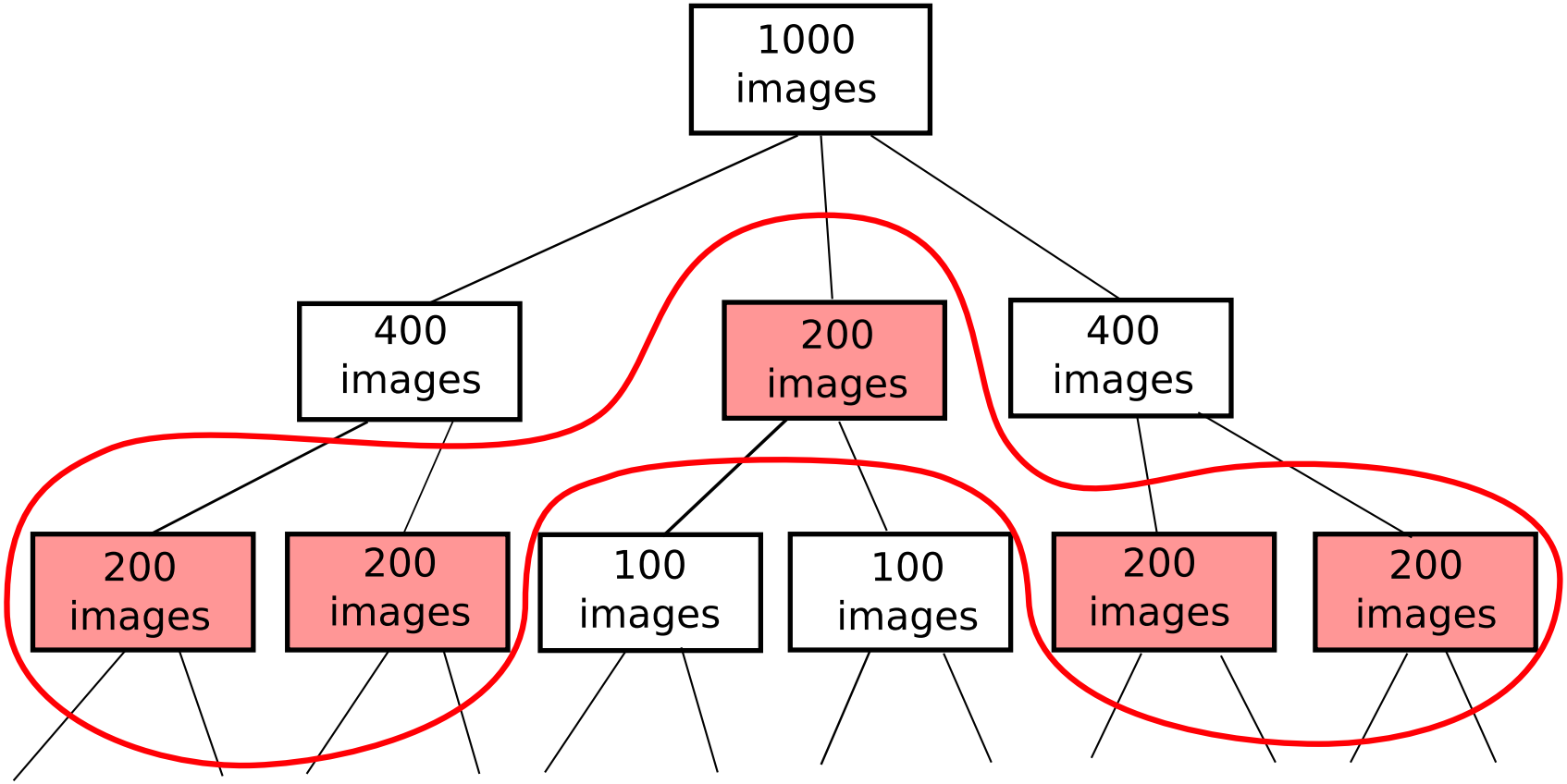
Inserting a New Tuple in the Cluster Hierarchy:

```
Insert(Node n, Image i)
  if n.children == [] // Leaf node
    add(n, i);
    return;
  else
    result = Match(n.children); // Ordered list
    if StrongMatch(i, result[0]);
      Insert(result[0], i)
    else
      CreateNewNode(n, i);
    EvaluateSplitting(result[0]);
    EvaluateMerging(result[0], result[1]);
end of Insert;
```

Selecting Clusters for the Lattice Generation



Selecting Clusters for the Lattice Generation



Hypermedia Representation

From the user point of view, the navigation becomes *two-levels*.

- *Inter-clusters* navigation: similar to navigating a lattice of images.
- *Intra-clusters* navigation
  - The average cluster size is about 200 elements

We assume that a user can view about **50 thumbnails** on the same screen. Thus, for about **200 images** a navigation structure is not required.

Hypermedia Representation

From the user point of view, the navigation becomes *two-levels*.

- *Inter-clusters* navigation: similar to navigating a lattice of images.
- *Intra-clusters* navigation
  - The average cluster size is about 200 elements

We assume that a user can view about **50 thumbnails** on the same screen. Thus, for about **200 images** a navigation structure is not required.

Problem

How to select the first 50 elements to present?

Algorithm

Selecting the best n representatives of the cluster z (recursive algorithm) for intra-cluster search:

```
Choose(n integer, z cluster) returns a set of images
  if (z content <= n) then
     return all z content
  else if (z is a leaf) then
     return n random samples
  else return for (each z_child) do
     Choose((n / number_of_children(z)), z_child )
end of Choose;
```

Conclusion

We associate a scalable clustering technique to our Galois' lattice approach.

- Time complexity for construction:
  square $O(n^2) \rightarrow$ **linear** $O(n + \left(\frac{n}{C}\right)^2)$ (n number of images).
- The two-levels approach makes navigation easier for the user.

We can provide a navigation structure for very large databases

Example

5,000 clusters of 200 images $\rightarrow$ 1,000,000 images

Outline

Problem

- The user should be able to identify parts of the structure as irrelevant, in order to work on a reduced structure
- A Galois lattice being a multidimensional structure, naive approaches to modify it may be exponential.

Problem

- The user should be able to identify parts of the structure as irrelevant, in order to work on a reduced structure
- A Galois lattice being a multidimensional structure, naive approaches to modify it may be exponential.

Goal

Provide the user an efficient way to personalise the structure to his needs.

(Feedback querying provides an answer, but the problem of query generation is NP-complete.)

Masking Lattices

We propose to provide **user-personalisation** by applying user-defined *masks*. The resulting lattice is called a **sub-lattice** of the Galois' lattice.

- During the retrieval process, the user selects either *elements* to mask or *properties* to mask.
- At any time, the user can ask for calculation of a *lattice mask*.

Benefits

- *Efficient way to provide user-personalisation.*

Processing is done before-hand, on a sub-lattice each step's time complexity is $O(1)$. $\neq$ Feedback querying: NP-complete problem.

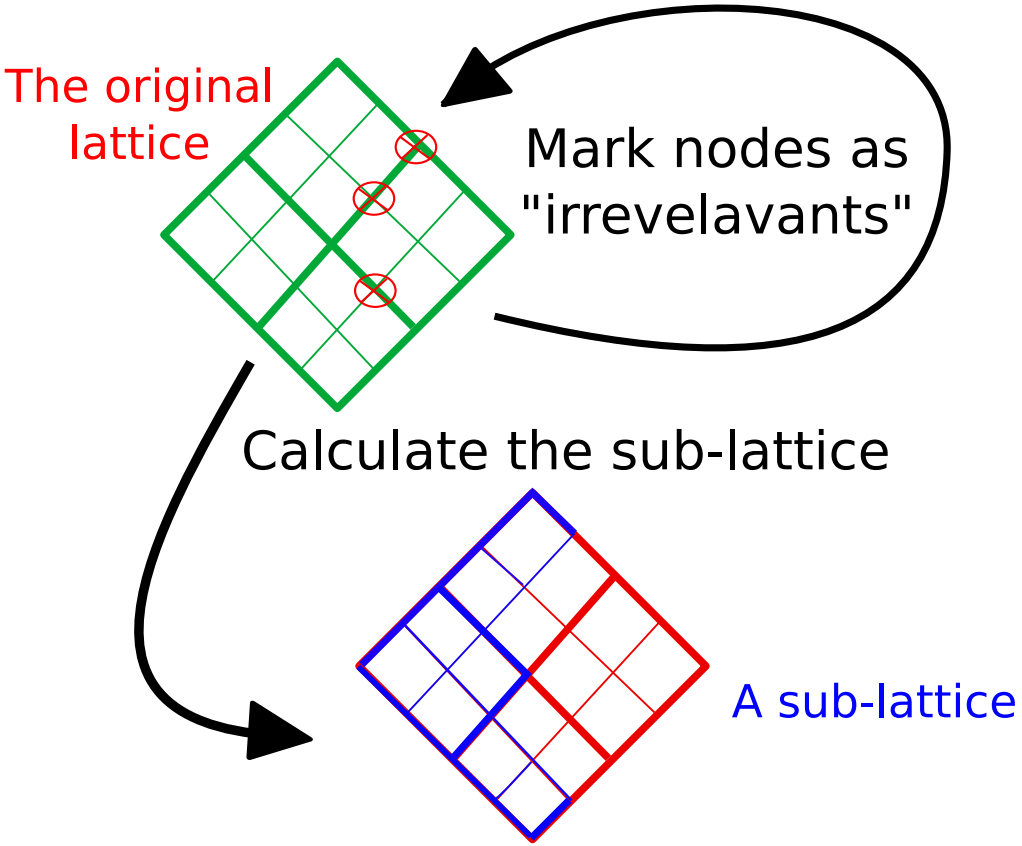
Algorithm

The user can mark nodes as *irrelevant* for his search. To determine the nodes and edges to mask:

- Minimum and maximum nodes cannot be masked,
- Any edge connecting a masked node will be masked,
- Edges are added to ensure unicity of the *min* and the *max*
- Nodes are merged to avoid nodes with a single child or single parent

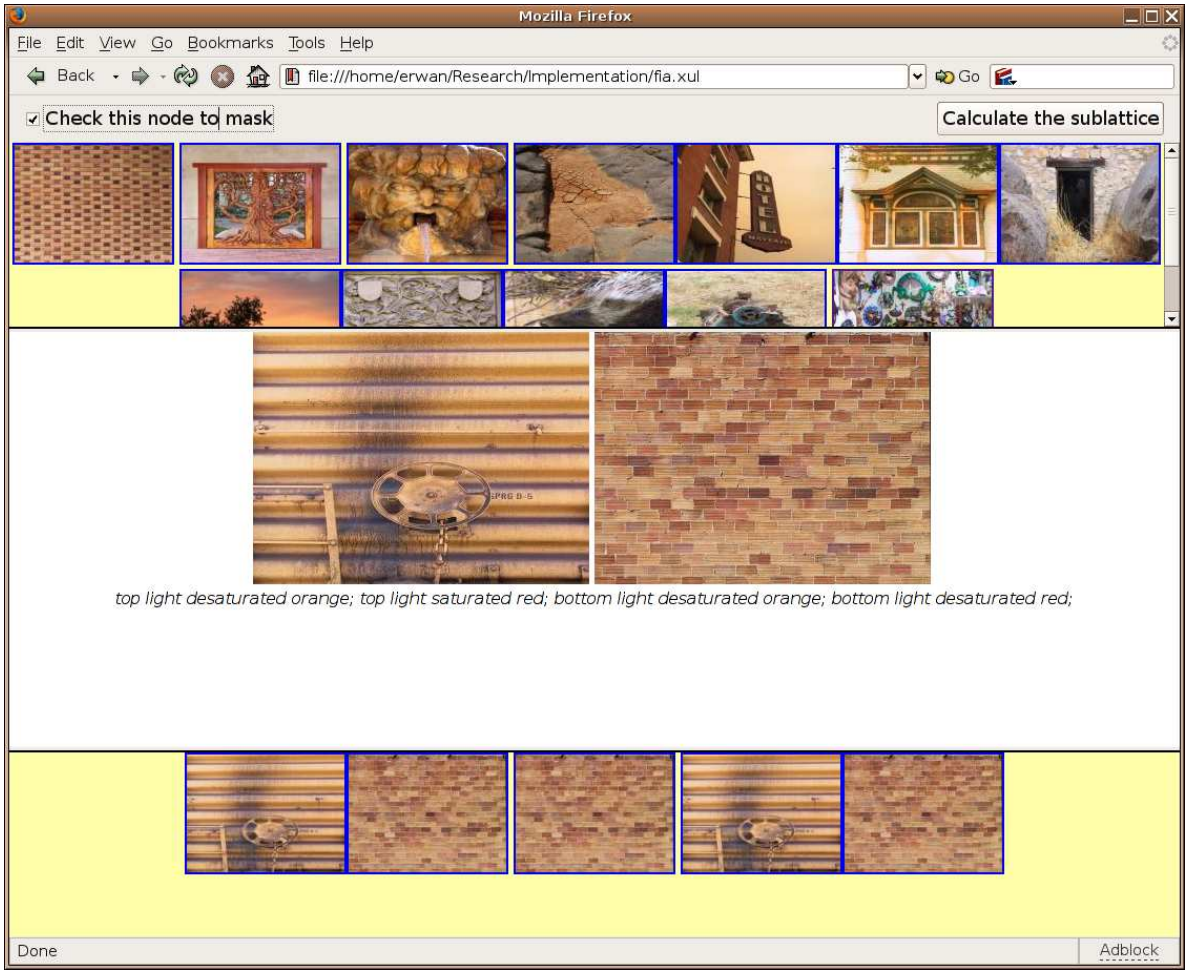
Algorithmic complexity: $O(n)$ (n being the number of nodes to mask).

Process



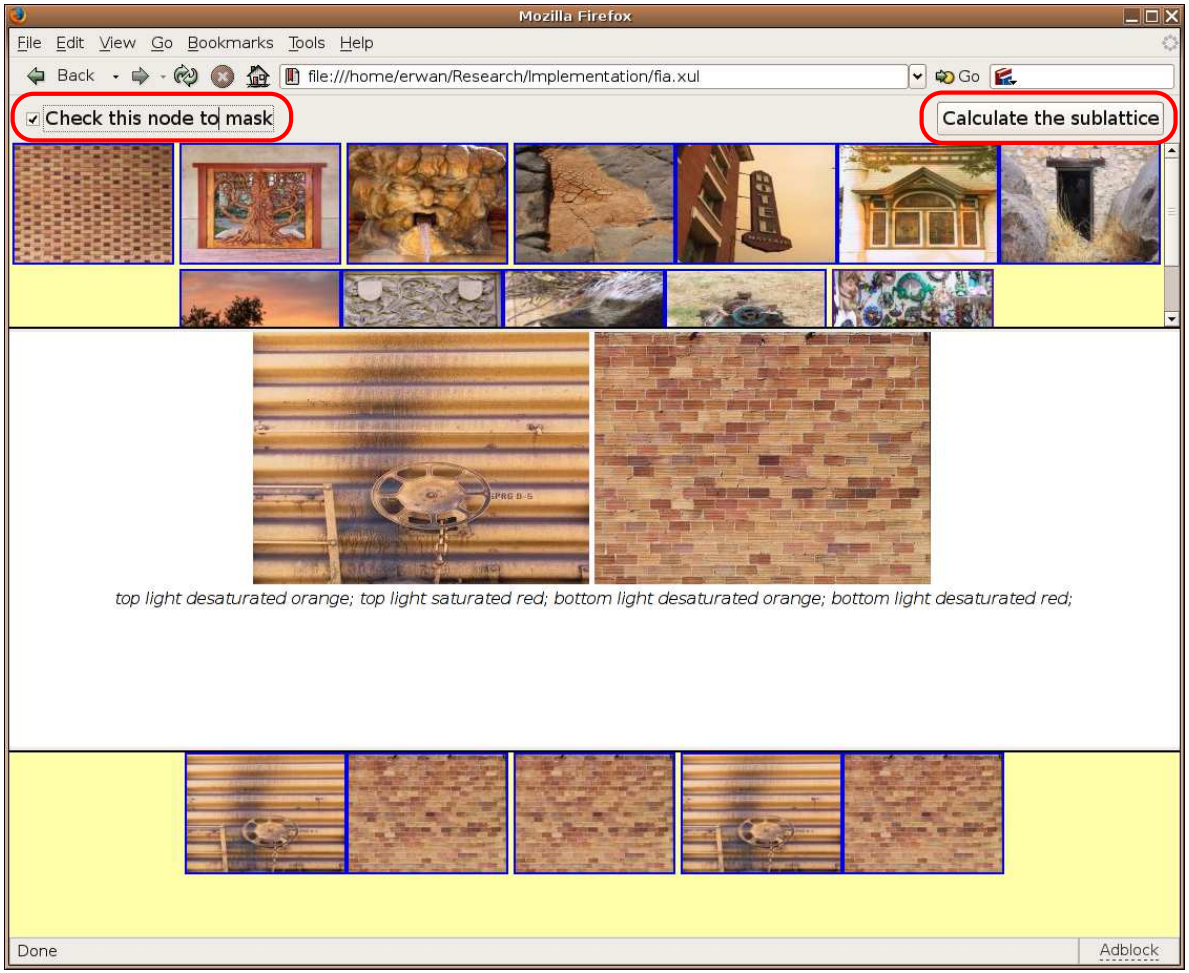
Masking a Lattice (From the User Point of View)

Example



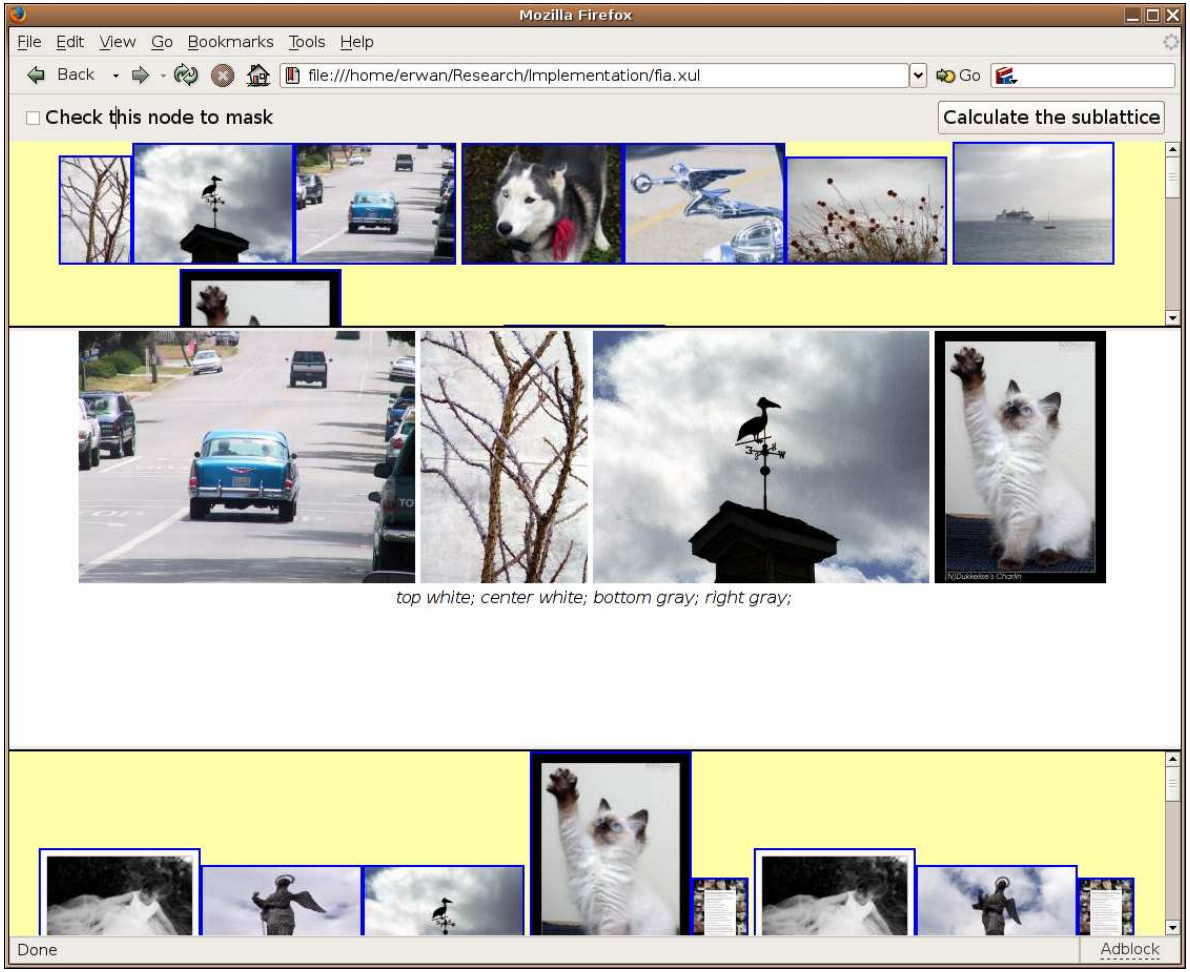
An irrelevant node

Example



An irrelevant node

Example



A more relevant structure

Conclusions

Our proposal:

- Provides user-personalisation, allowing users to define sub-lattices.
- Does not deny the performances advantages.

Compared to other systems, it is:

- More relevant than a system based solely on a pre-calculated structure.
- More efficient than a system based on feed-back querying ($O(n)$ for the sub-lattice calculation, $O(1)$ for navigation).

Outline

Problem

Problem

A non-adaptative algorithm leads to the creation of too many *virtual nodes*

Virtual Node: a node with no image matching its exact set of properties. Virtual nodes are required for the navigation, but too many virtual nodes make the navigation more difficult.

Problem

Problem

A non-adaptative algorithm leads to the creation of too many *virtual nodes*

Virtual Node: a node with no image matching its exact set of properties. Virtual nodes are required for the navigation, but too many virtual nodes make the navigation more difficult.
Example: Using a threshold of 0.3:

- $\{blue/0.5, red/0.31\}$, $\{blue/0.5, yellow/0.31\}$ and $\{blue/0.5, green/0.31\}$ will lead to the creation of **3 nodes**
- $\{blue/0.5, red/0.29\}$, $\{blue/0.5, yellow/0.29\}$ and $\{blue/0.5, green/0.29\}$ will be integrated into the **same node**.

Our Approach: Insertion by Node Matching

Idea

When inserting a new image (with fuzzy value associated), try to match an existing node for the binarisation process.

- The building process remains incremental;
- New images are inserted into existing nodes rather than in a new node.

Our Approach: Insertion by Node Matching

Idea

When inserting a new image (with fuzzy value associated), try to match an existing node for the binarisation process.

- The building process remains incremental;
- New images are inserted into existing nodes rather than in a new node.

Benefits

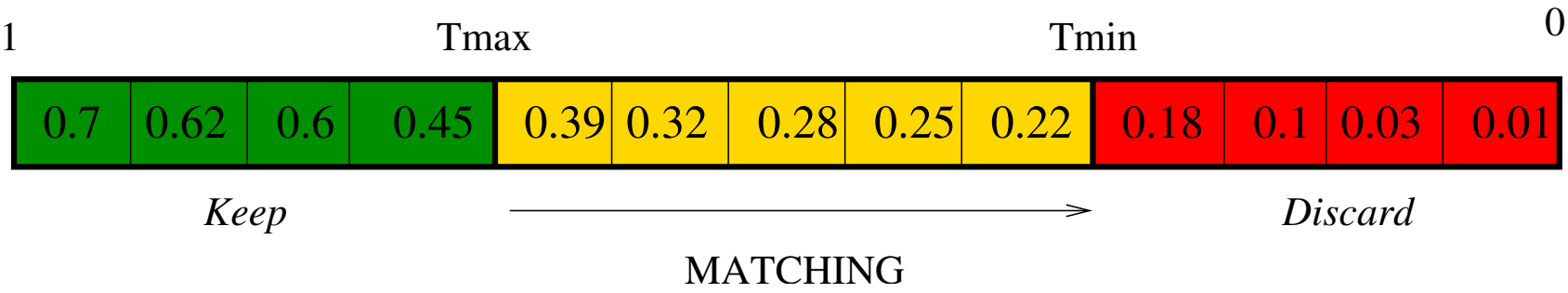
- The number of nodes is reduced without information loss.
- Thus, the average path to a given image is shortened.
- The number of element in each node is increased.

To search images in such a structure is easier.

Algorithm

For a given image:

- Descriptions are ordered by tipicity
- Descriptions under a T_{min} threshold are discarded, descriptions over a T_{max} threshold are kept;
- Starting from the set of descriptions under T_{min}, we remove descriptions successively and try to match the resulting key with existing keys in the lattice;
- If no match can be found, an average threshold is used.



Experiments

Done on the collection of Flickr.com images described previously (5,000 images);

- Navigation structures built for:
  - The naive solution (constant threshold);
  - Our proposal (node matching).
- Metrics: the structure quality:
  - Cardinal;
  - Average node size;
  - Ratio of *real* nodes (opposed to *virtual* nodes).
- The resulting structures have been tested on a few users.

Results: Real nodes

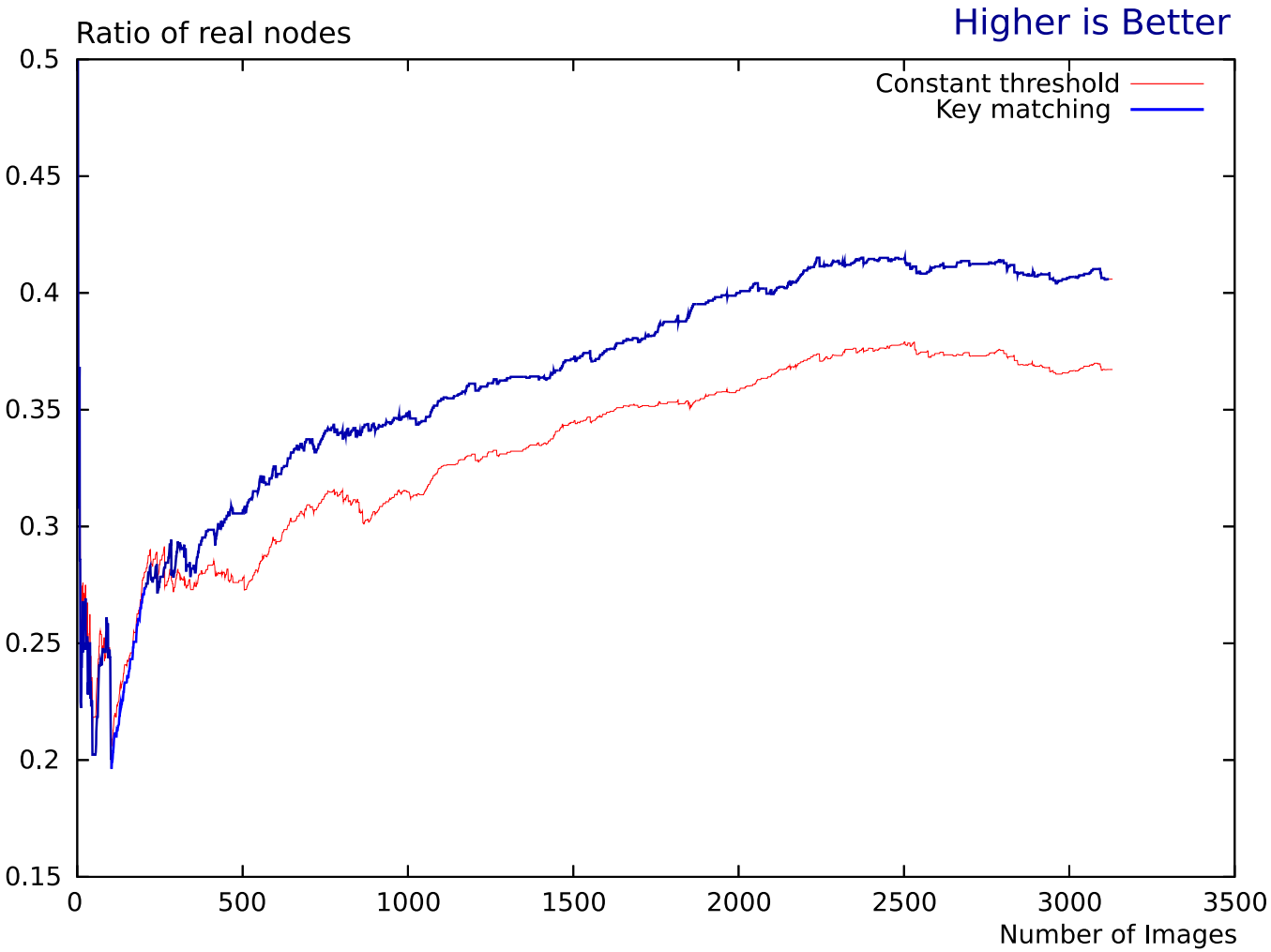


Figure: Ratio of Real Nodes in the Lattice

Results: Total Number of Nodes

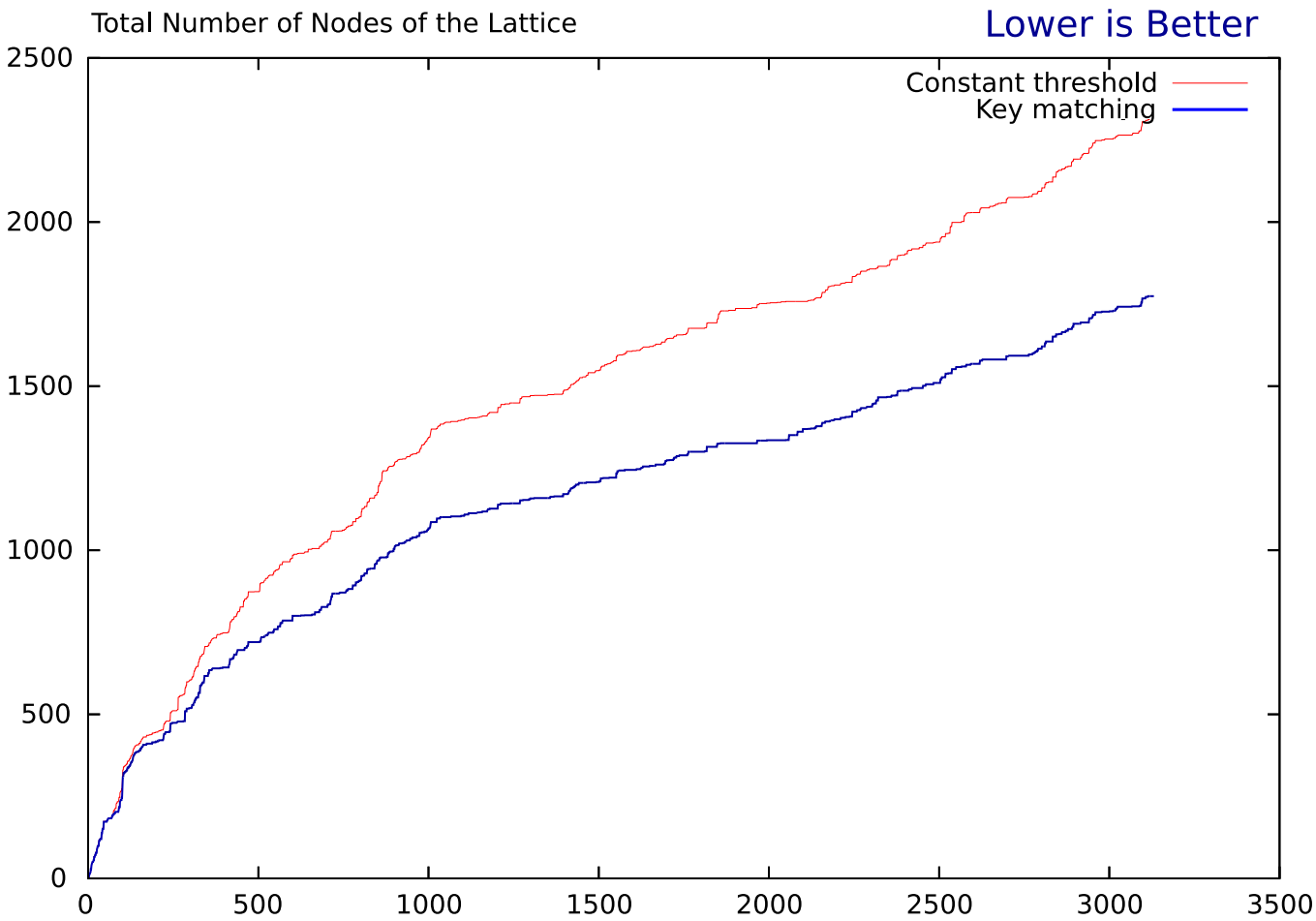


Figure: Total Number of Nodes

Results: Average Size of a Node

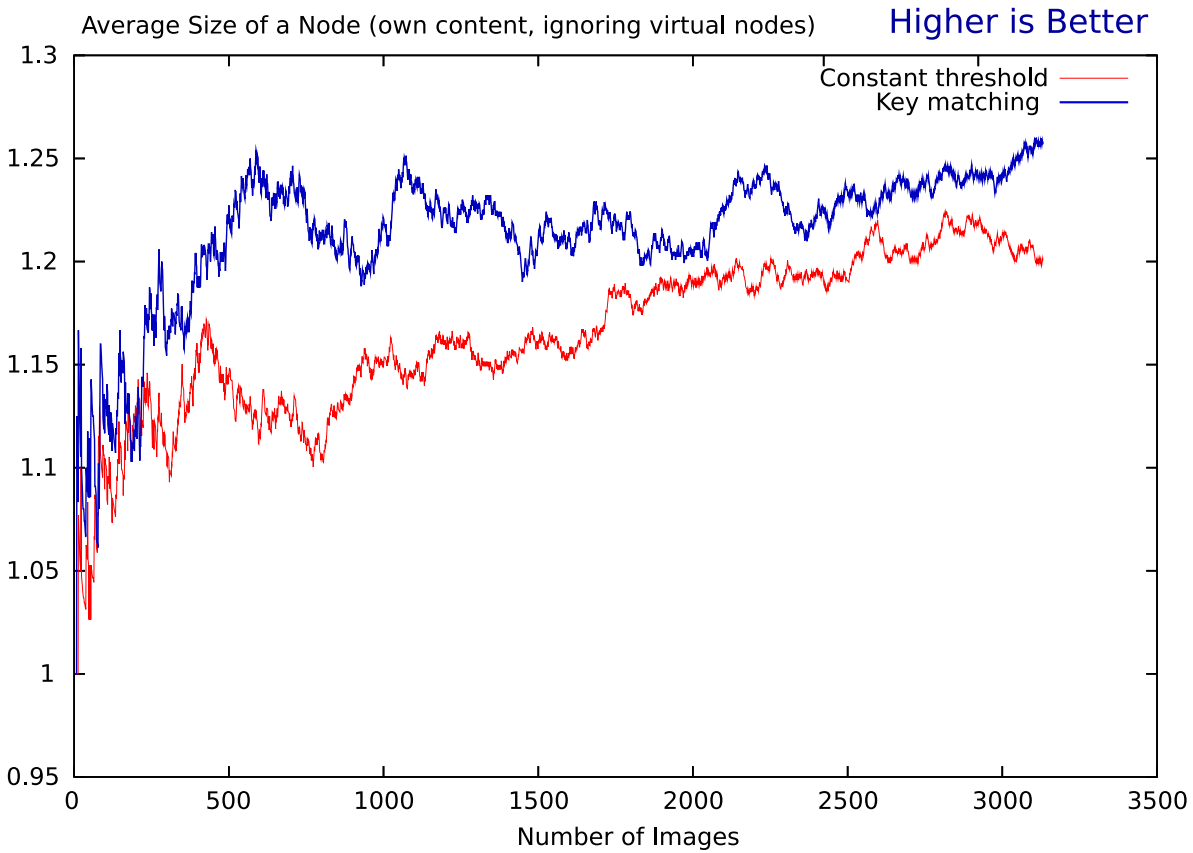


Figure: Average Size of a Node (excluding virtual nodes)

Results: Average Size of a Node

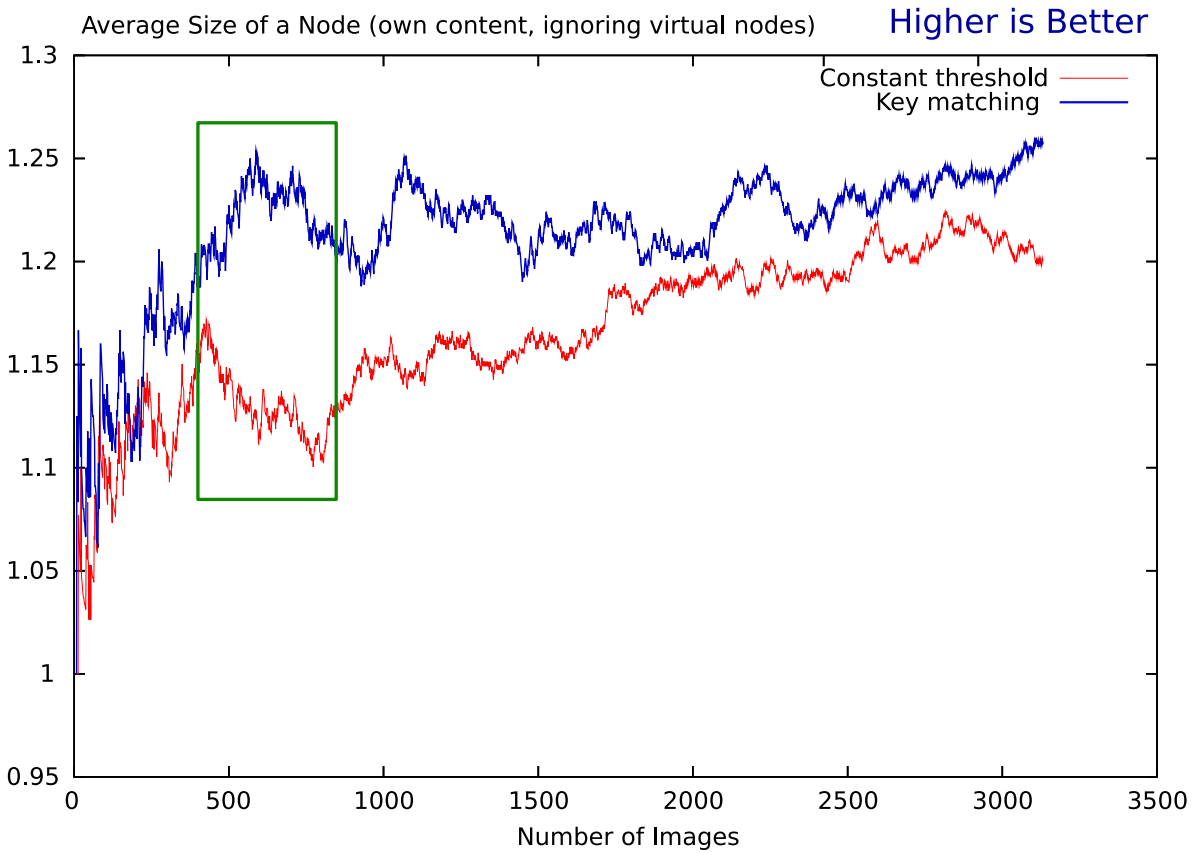


Figure: Average Size of a Node (excluding virtual nodes)

Conclusions

Node matching insertion produced a structure of better quality than other approaches:

- The number of nodes is reduced without information loss.
- Thus, the average path to a given image is shortened.
- The number of element in each node is increased.

Consequently, the user experience is better due to a more compact structure.

Part III

Conclusion

Outline

Outline

Achievements

In this study, we looked for a way to replace query by navigation in image databases.

- We presented a technique to search for images by browsing an hypermedia representation of a Galois lattice
- We presented several extensions to:
  - Address the scalability problem by using a clustering technique
  - Allow the user to customise the structure for his needs
  - Make a better use of the fuzzy description of the images to build a better structure

Benefits and Limits

Benefits:

- **Easy to use:** the user does not have to describe formally his needs, he just selects the images he "likes". He can still give feedback to personalise the search.
- **Fast and responsive:** browsing is optimal since it is done on a static structure.
- **Scalable:** Associated with a clustering technique, it can reach very large collections.

Limits:

- Not applicable for open collections such as the World Wide Web.

Outline

Further Work

- Application to other media types, such as video. Requires:
  - A suitable metamodel
  - A compact way to represent the medium
- Mobile computing
  - Insertion requires no interaction
  - A mobile phone can provide additional metadata
  - The user interaction is very simple