

Galois' Lattices as a Classification Technique for Image Retrieval

ERWAN LOISANT,^{†,††} JOSÉ MARTINEZ,[†] HIROSHI ISHIKAWA^{††}
and KAORU KATAYAMA^{††}

Going one step ahead feedback querying in integrating users into a search process, navigation is the more recent approach to finding images in a large image collection by using content-based information. Rather than using queries or going into a feedback querying process that would be both heavy in terms of human-computer interaction and computer processing time, navigation on a pre-computed data structure is easier and smoother for the user. In particular, we found Galois' lattices to be convenient structures for that purpose. However, while properties extracted from images are usually real-valued data, most of the time a navigation structure has to deal with binary links from an image (or a group of images) to another. A trivial solution to get a binary relationship from real-valued data is to apply a threshold, but this solution not only leads to a loss of information but also tends to create sparse areas in the lattice. In this paper, we propose a technique to incrementally build a Galois' lattice from real-valued properties by taking into account the existing structure, thus limiting the size of the lattice by avoiding the creation of sparse nodes. Experiments showed that this technique produces a navigation structure of better quality, making search process faster and more efficient, thus improving user's experience.

1. Introduction

There exists a range of ways to querying content-based image retrieval (CBIR) systems, which have been implemented – partly – in several prototypes. They can be classified into three complementary approaches:

- formal querying where the emphasis is put on specifying the image to find, in other words building a query (even if the “querying” language may be very simple, even reduced to a linear function like in QBIC⁴⁾ or Virage²⁾), or graphical like in QBIC or Ref. 4), NETRA¹⁰⁾;
- feedback querying where the user is involved in an iterative retrieval process where he or she indicates whether images of the result set are relevant (Surfimage¹²⁾);
- image search based on navigation that allows the user to move into a complex, organised space through mere clicks¹⁵⁾.

Our research work on image search based on navigation has been implemented into a prototype called $Click_{AGE}^{Im}$, part of the larger project $Find_{AGE}^{Im}$ that also includes the aforementioned approaches for CBIR.

Figure 1 shows the user's interface, which currently consists of XHTML pages accessible

from any web browser with simple processing based on ECMAScript. We can see, in the central part of this screen shot, the current images that the user is browsing. By selecting images in the lower part, the user can specialise his or her query. On the contrary, he or she can select images in the upper part to make his or her query more generic. A “query” is a conjunction of descriptions; a click on the upper part removes descriptions from the list while a click on the lower part adds descriptions.

This project was extended by the present proposal, a technique called *Fuzzy insertion by key matching*, that makes use of the existing structure when inserting a new element.

In this paper, we present a navigation technique based on Galois' (concept) lattices. A Galois' lattice is a graph structure build from a binary relationship, in our case the image/metadata relationship. More specifically, we propose a technique to construct this navigation structure that avoids sparse elements (elements that are far away from each others), for sparse elements will lead to a fast growth of the number of nodes, hence something that we assimilate to *noise*.

An image is usually represented by descriptors represented by real numbers. More specifically, in this work we use fuzzy logic values called *typicity* values, i.e., belonging to the $[0, 1]$ interval. For example, an image can feature the description “*blue top part*” associated with

[†] Atlas-GRIM Team, INRIA & LINA (FRE CNRS 2729), France

^{††} Graduate School of Engineering, Tokyo Metropolitan University

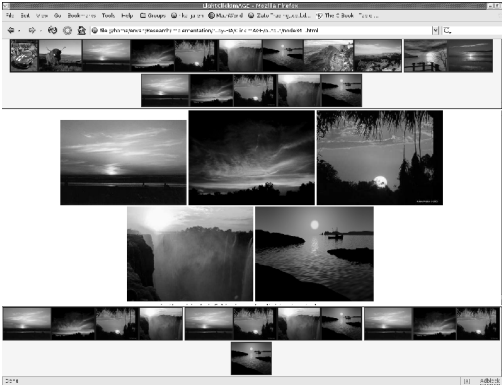


Fig. 1 *ClickIm_{AGE}* user's interface.

a typicality of 0.5 if the blue colour covers, say, half of the upper part, while another image may feature the same description associated with a typicality of 0.7 if the blue colour covers a bigger area of the upper part of that image. However, the navigation structure should be binary. The trivial solution, which is applying a constant threshold to the typicality of each description, leads to the creation of sparse areas in the structure as well as isolated elements.

The technique presented here addresses this issue by building a better navigation structure, i.e., a structure with fewer and more compact groups of images. In effect, it produced good results as it can reduce the complexity of a navigation structure based on Galois' lattice without suppressing information on the images. Moreover, there was no significant performance loss compared to a simple construction based on a constant threshold.

Compared to other so-called "navigation" techniques that actually send a query to the database at each iteration to build the new answer, our proposal, based on Galois' lattices, builds a navigation structure before-hand; thus, navigation is done on a static structure and cannot be more efficient. Fortunately, Galois' lattice is a graph structure complex enough to avoid a bias in the navigation possibilities. The technique described in this paper takes into account the existing structure in order to improve its quality by reducing the number of small nodes without sacrificing the intrinsic advantages of Galois' lattices.

Our research work has been implemented into a working prototype, *ClickIm_{AGE}*, calculating metadata for a collection of images and producing a navigation structure based on Galois'

lattices as a set of XHTML pages.

The outline of the paper is as follows: In Section 3, we present a simple, illustrative metadata model to describe image content. This model is based on geometric informations (orientation and elongation) and colour information.

In Section 4, we introduce the definition and the properties of a Galois' lattice, why and how we use it for image retrieval.

In Section 5 we present our technique to convert a fuzzy model into a binary model while operating the noise reduction. This is done by taking the existing structure into account while adding a new element into the lattice.

Finally, in Section 6 we validate our hypothesis by applying our proposal to a collection of 3,000 images and comparing it to the trivial technique (applying a constant threshold). It appeared to considerably reduce the number of sparse nodes, thus reducing the number of steps to find the images that the user is looking for.

2. Background and Comparison

Let us introduce the rationale for relying on Galois' lattices¹¹, then compare their use to more classical querying approaches as well as to other usages.

2.1 Choice of Galois' Lattices

An image lattice is just another instantiation of a Galois' (or concept) lattice. (This mathematical structure has been largely exploited in the field of knowledge discovery⁶.) As a navigation structure, the advantages of Galois' lattices are really numerous:

- It is very fast to navigate through a graph structure that has been computed off-line. If we neglect the time required to load sample images, navigating from one node to another is optimal, i.e., in $O(1)$.
- A Galois' lattice is intrinsically a multi-dimensional classification technique. Indeed, no dimension is privileged. Hence, it can be seen as a structure that dichotomises the hyper-cube associated to the property subsets along any hyper-plane.
- The distance from the inf or sup nodes of the graph to any other node is at most logarithmic in the number of used properties (details on Galois' lattices will appear in Section 4).
- Since there is no distance computation, this tool is insensitive to correlations. If images

with a given property (almost) always exhibit another property, then the images will simply be located within the same node.

- Being based on navigation, this tool helps to correct users' mistakes very easily. Whenever a user selects a direct descendant node, he or she adds implicitly a new constraint. If he or she figures out, much later, when seeing more specific sample images, that this browsing direction is slightly bad, he or she has only to move to a different *direct* ancestor node. This operation removes a constraint and undoes the erroneous move without having to go back to the point where the "error" actually occurred.
- The Galois' lattice structure easily hides unwanted features. This is a problem that cannot always be taken into account by similarity measures. (A counter-example is Surfimage¹²), but the measures are limited to mean and variance of supposed Gaussian distributions.)

However, Galois' lattices used as a navigation structure also have some drawbacks.

- Constructing a Galois' lattice is not an easy task. The time complexity is quadratic ($O(n^2)$, where n is the number of nodes). Theoretical improvements on this bound are still unknown, to our knowledge, and algorithmic variants do not achieve actual improvements in the implementations⁶).
- The description space associated to a Galois' lattice is exponential in the number of properties. (We could easily use several hundreds!) Of course, this should not occur, unless we index such a large number of images. However, if several images share common properties but have unique properties too, then a (localised) exponential explosion appears.

We addressed the first drawback of Galois' lattices (scalability) in Ref.9) by building the lattice on a collection of *clusters of images* rather than on a collection of *images*.

The focus of this paper is on the second drawback, i.e., avoiding local explosions due to "noise" by inserting new images in an existing node, when possible, rather than creating a new one.

2.2 Querying vs. Browsing

The aim of our work is to provide a fully working prototype for content-based image retrieval, from metadata extraction to image

search itself. Our work is focused on a navigation-based process that integrates the user into the search process and thus ensures a permanent feedback between the user and the system.

A common problem in CBIR systems is that the user is usually unable to understand the underlying model used by the system. Cases where a user looks for an image knowing which colours or shapes he or she likes, and how to describe it, are very rare. In a navigation process, the user does not need to describe his or her need. He or she just browses the collection, constructing a path from an entry point to the images he or she mostly likes.

When looking at most proposals of navigation through a multimedia collection, we noticed that they are usually built as a layer over similarity search: they recreate a new state from user input, and display it to the user for another interaction.

Kaester, et al.'s work⁸) proposes to combine several input methods to search for images, including touch screen (to select parts of images or perform gestures) and speech recognition. By using these non-classical input methods Kaester could produce a graphical interface that makes the user feel like he or she is navigating the image database, however this system is still based on similarity search: the user will actually select images or parts of the images for the system to find images similar to these samples. The collection is stored by using multi-dimensional indexing techniques.

In their project El Niño, Santini et al. worked on integrating browsing and querying¹⁵). Their proposal is a set of search engines connected by a mediator that dispatches the queries to the search engines, collects the results and displays them to the user. Images are arranged on a two-dimensional plane, and the user interacts with the system mainly by two ways:

- By clicking on an image, the user asks the system to move this image to the centre. From the user's point of view, he or she is moving inside the collection; from the system's point of view, the user is launching the query "show the images similar this one."
- By drag-and-dropping images, the user teaches the system similarities that were not present. The user can then tell the system that from his or her point of view, two images are similar. In other words, this is

a user-personalisation process.

Our work is very different from Santini's or Kaester's ones in the sense that it is based on a navigation structure that is built before-hand. Consequently, there is no calculation during the search process; this leads to a very fast and responsive system. Compared to a system where queries are written at each iteration (which is usually a very costly process), user's experience is greatly improved.

Compared to other systems performing navigation through image collections^{3),14)}, our system has the particularity to build a structure that is both an index structure and a structure used for navigation, while the others separate the indexing process from the navigation one.

While we have to face a more costly process to index our images, once the collection is indexed, the search process is very fast and easy to publish. The navigation structure can even be produced in the form of a set of static XHTML (see Fig. 1), and for example burned on a CD-ROM.

2.3 Galois' Lattices in CBIR

Despite of its several advantages, there are still few proposals using Galois' lattices in image search.

- Zenou, et al.¹⁸⁾ use Galois' lattices to characterise images and classify them into known sets. In that work, Galois' lattices are used mainly to identify descriptions that are specific to a given set, using examples.
- Ahamad, et al.¹⁾ use Galois' lattices as an index structure to perform classical queries on an image collection.

Our approach is different from Zenou's and Ahamad's approaches in the sense that we use Galois' lattices not only for indexing or classification but also for the search process itself. Thus, if we neglect the time of loading images to be displayed, our prototype is optimal, i.e., the navigation process is $O(1)$.

3. Establishing a Fuzzy Model for an Image Collection

This section presents the model that we used for our $Click_{AGE}^{Im}$ prototype. The interested reader may read⁹⁾ for more detailed explanations.

In this study, we restrict our attention to "standard" images: two-dimensional, rectangular, without transparency channel and not bounded to any particular area (such as satel-

lite or medical images). Specific kinds of images require adequate descriptions that can be far different from the ones introduced hereafter.

An exhaustive description of the content of an image requires a lot of properties. However, considering too much properties at once generally suffers from some drawbacks. Firstly, query performances degrade rapidly, that is called the "high dimensionality curse problem." This is not a problem here since building the hyper-text of images is done off-line. Hence, navigation offers optimal performances both from the processing and storage points of view, because the links are "hard-coded." Secondly, common weighted queries are very sensitive to correlated features¹⁶⁾. The technique that we use is insensitive to this problem due to the absence of weights, and even of distances.

The metrics we chose are:

- Geometric informations: orientation, elongation, and size. Some semantic is present in these metrics since the photographer chooses the orientation of his or her picture according to the content; the same remark applies to elongation since the choice of, for example, a panoramic view is also linked to semantics.
- Dominant colours classified within a division of the image into five parts (top, bottom, left, right, and centre). Most of the semantic that can be automatically extracted is in this colour information (in the sense that a large number of "real-world objects" have a known and uniform colour assigned to them).

3.1 Colour Model

Colour information, in particular colour histograms, has been largely exploited in CBIR; it is recognised as a good measure for image search¹⁷⁾.

To represent it, we chose the Hue/Saturation/Value (HSV) colour model, illustrated on **Fig. 2**: it combines a good perceptual fidelity with a low processing cost.

There is one problem in the use of HSV for CBIR: the hue is not perceptually linear. Some colours, like red or blue, are over-represented while others, like yellow, are under-represented. We address this problem by providing a perceptual partitioning of the HSV space, consequently eliminating the bias in the results.

Definition 1 (Description Space)

The description space \mathcal{D} consists of the union of several sub-descriptions:

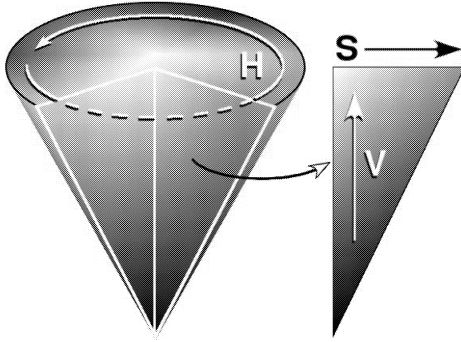


Fig. 2 The Hue/Saturation/Value (HSV) colour model.

$$\begin{aligned} \mathcal{D} = & \mathcal{D}_{area} \cup \mathcal{D}_{orientation} \cup \mathcal{D}_{elongation} \cup \\ & (region \times \mathcal{D}_{hue}) \cup (region \times \mathcal{D}_{saturation}) \\ & \cup (region \times \mathcal{D}_{intensity}) \quad (1) \end{aligned}$$

3.2 Fuzzy Linguistic Labels for Colour

Fuzzy linguistic Labels (FL for short), like *pink*, need to be described over the colour domain, namely, the HSV space. Each FL c is associated to a membership function μ_c with values within $[0, 1]$, reflecting how well the FL c describe the colour of a particular image or region of an image. For instance, the *pink* label could be represented over the HSV colour space by using a membership function $\mu_{pink}(h, s, v)$ with high membership values associated to colours with a hue close to red, a low saturation, and a very high value.

Definition 2 (Fuzzy Subset)

A fuzzy subset A of a set X is defined by a membership function (or characteristic function) μ_A where $\forall x \in X, \mu_A(x) \in [0, 1]$. $\mu_A(x)$ represents the membership degree of x in the fuzzy subset A .

This general approach¹³⁾ allows to represent any FL with a fuzzy set directly defined over the colour domain. This leads to great flexibility and precision, but raises some problems:

- it is difficult for users to set all the linguistic label values in this three dimensional space;
- It is difficult to achieve the complete coverage of the colour space.

Coverage of the colour space is required to ensure that any image will have at least one representation in terms of linguistic descriptors. Allowing FL to be defined separately from each other would allow greater flexibility but will need a check to ensure that the designer has not left some “holes” in the description of the colour space. To overcome this problem, the fol-

lowing method has been preferred. Any colour linguistic label c is formed from the concatenation of elementary labels $c.h$, $c.s$ and $c.v$ defined on each of the H, S and V dimension. The membership function associated with c is computed as a conjunction of the elementary label membership functions:

$$\mu_c(h, s, v) = \mu_{c.h}(h) \otimes \mu_{c.s}(s) \otimes \mu_{c.v}(v)$$

where \otimes represents a T-norm (max for instance) used to compute the conjunction of the individual channels. In this test, we used the following partition:

- Hue : *red, orange, yellow, green, cyan, blue, purple*;
- Saturation : *vivid* and *dull*;
- Value : *dark* and *bright*.

These partitions (the hue partition in particular) have been defined in order (1) to match the human eye perception and (2) to cover the whole colour space.

Colour labels are formed by combining those terms (e.g., *vivid bright red*). Under a certain value level, a colour is perceived as black. In the same way, colour is perceived as grey or white, under a certain saturation level. Hence, to the above collection of colour terms, we add *black, dark grey, bright grey*, and *white* resulting in a total of 32 individual terms. This number is in the range of the QBIC system⁴⁾ which defines a set of 25 colours.

This method is much simpler from the user’s point of view, and the coverage of the HSV space is guaranteed as long as each dimension is well covered, which is trivial to achieve.

3.2.1 Zone Colour Characterisation

Colour perception results from the juxtaposition of individual pixels. The perceived colour of an arrangement of pixels ranges from uniform pure colour to complex colour arrangement without dominating colour.

Considering our linguistic representation of colours, each pixel colour is expressed in terms of colour labels with different weights. For a pixel, the weight is the membership degree of its colour to the fuzzy set associated with a colour label. For instance, in our paradigm of representation, a pink pixel could be defined as two colour labels:

- *vivid bright red* with a membership degree of 0.1;
- *dull bright red* with a membership degree of 0.9.

Considering a region S as a collection of adjacent pixels, the relative importance $\tau_S(d)$ of a

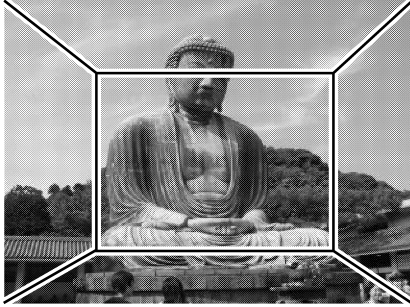


Fig. 3 Syntactical division: Big Buddha.

colour label d inside S , is computed as the sum of membership degrees $\mu_d(p)$ of each pixel:

$$\tau_S(d) = \frac{\sum_{p \in S} \mu_d(p)}{\sum_{d' \in \mathcal{D}} \sum_{p \in S} \mu_{d'}(p)}, \quad (2)$$

with \mathcal{D} the set of all colour labels.

3.3 Syntactical Division

An image “segmentation” is used to allow for a more accurate description of image colours. However, due to (1) the high algorithmic cost of the segmentation algorithms and (2) the failure of these algorithms to recognise real-world objects made of several parts of different colour and texture, we relied on a syntactical partitioning of the images rather than a true segmentation.

Considering general photographic pictures, the main subject often stands in the centre and the surrounding areas represent the image background. In a landscape picture, for instance, the sky is likely to have blue or grey hues, while the ground will probably be green. Among different divisions, we tried a division into five trapezoids seems a good model to separate semantics in the general case.

These five trapezoids are respectively the centre, the left, the right, the top, and the bottom of the an image. **Figure 3** shows an image example featuring Kamakura’s Big Buddha divided into trapezoids. The central zone covers 40% of the total surface and the four surrounding zones are trapezoids, the wideness of which is 15% of the image wideness.

3.4 General Geometrical Measures

Along with colour, we consider geometrical measures to represent *area*, *orientation*, and *elongation*.

The use of orientation and elongation for im-

age search is quite obvious: an image wider than high is usually called a “landscape” image precisely because most landscape images have these proportions. Even when it is not an actual landscape, a landscape-oriented image usually represents several real-world objects, a very large image being usually a panoramic view. On the contrary, an image higher than wide is called a “portrait” because portrait usually have these proportions. When it is not a portrait, a portrait-like image usually represents a close-up of a unique real-world object. Thus, there is an implicit correlation between orientation and elongation, and parts of the semantics of an image.

Area is also an important measure since user may be interested by retrieving images of a certain size according to his or her needs.

3.4.1 Numerical Features

Firstly, \mathcal{D}_{area} , $\mathcal{D}_{orientation}$, and $\mathcal{D}_{elongation}$ are format information (MPEG-7 level 1), related to the whole image i , based on the following *scalar* functions:

$$area(i) = width(i) \times height(i) \quad (3)$$

$$orientation(i) = \frac{1 - \cos(2\alpha(i))}{2} \quad (4)$$

$$elongation(i) = \left| \frac{4\alpha(i)}{\pi} - 1 \right| \quad (5)$$

where $\alpha(i)$ is the angle of the diagonal defined as $\text{atan} \frac{width(i)}{height(i)}$.

These formulæ have been chosen among various alternatives in order to remove (or minimise) the correlations. Orientation of the image (i.e., portrait or landscape) is independent of the area of the image.

Also, using an absolute value removes the correlation between orientation and elongation measures: the covariance is null on $[0, \pi/2]$. (However, correlation is complete per halves, i.e., on $[0, \pi/4]$ and $[\pi/4, \pi/2]$ independently, due to the functional dependency between orientation and elongation.)

3.5 Linguistic Variables

The introduced numerical features are used in turn to provide metadata on images as subsets of the following descriptions:

$$\mathcal{D}_{area} = \{tiny, small, medium, large, huge\} \quad (6)$$

$$\mathcal{D}_{orientation} = \{portrait, square, landscape\} \quad (7)$$

$$\mathcal{D}_{\text{elongation}} = \left\{ \begin{array}{l} \text{none, standard,} \\ \text{panoramic, elongated} \end{array} \right\} \quad (8)$$

$$\mathcal{D}_{\text{hue}} = \{\text{red, orange, } \dots, \text{blue, purple}\} \quad (9)$$

$$\mathcal{D}_{\text{saturation}} = \{\text{vivid, light, pale}\} \quad (10)$$

$$\mathcal{D}_{\text{intensity}} = \{\text{black, dark, light, white}\} \quad (11)$$

These discrete subsets are obtained from the corresponding fuzzy linguistic variables and fuzzy subsets, and subsequent thresholding.

The three following definitions inform the reader about the exact computations conducted in order to produce the fuzzy metadata. However, they are not central to this paper and can be skipped.

Definition 3 (Scalar Fuzzy Value)

The fuzzy value of a scalar property with respect to a fuzzy subset A is directly given by its membership function:

$$FV_i : \begin{array}{l} \mathbb{R} \times \mathcal{D}_i \rightarrow [0, 1] \\ (x, A) \mapsto \mu_A(x) \end{array} \quad (12)$$

where $i \in \{\text{area, orientation, elongation}\}$, \mathbb{R} being the set of the real numbers.

Definition 4 (Vectorial Fuzzy Value)

The fuzzy value of an histogram h with respect to a subset A is computed as follows:

$$FV_j : \begin{array}{l} ([0, 1] \rightarrow [0, 1]) \times \mathcal{D}_j \rightarrow [0, 1] \\ (h, A) \mapsto \int_0^1 h(x) \times \mu_A(x) dx \end{array} \quad (13)$$

where $j \in \{\text{hue, saturation, intensity}\}$.

Definition 5 (Fuzzy Description)

A fuzzy description is a set of fuzzy subset names and corresponding non null fuzzy values:

$$FD_i : \begin{array}{l} \mathbb{R} \cup ([0, 1] \rightarrow \mathbb{N}) \rightarrow 2^{\mathcal{D}_i \times [0, 1]} \\ p \mapsto \{(A, FV_i(p, A)) \\ | A \in \mathcal{D}_i \wedge FV_i(p, A) > 0\} \end{array} \quad (14)$$

\mathbb{N} being the set of the natural numbers.

4. Constructing a Galois' Lattice from a Binary Relationship

Browsing techniques are numerous, ranging from mere guided and indexed tours^{5),7)} to advanced tours. Whatever technique is used, it requires prior structuring of the data. For instance, a guided tour corresponds roughly to a linked list constructed from a set by possibly selecting some items and specifying a given order between them.

Images are described by various properties, their so-called metadata. If we envisage only simple tours on simple properties, like the list

of all the grey-level images, the user will be unsatisfied with lengthy lists. Combining several properties is unavoidable, hence the use of a Galois' lattice for our $Click_{AGE}^{Im}$ system. Through mere clicks one can either focus on more and more constrained images (for example mainly blue, panoramic images, with a strong texture), or on the contrary have a fast access to general classes of images (just the previous grey-level images).

4.1 Galois' Lattices

A Galois' lattice can be constructed starting from any binary relation on discrete domains.

Definition 6 (metadata)

A simple *metadata* structure is defined in the form of a binary relation:

$$R : \mathcal{I} \times \mathcal{D} \quad (15)$$

where $\mathcal{I} = \mathbb{N} \times \mathbb{N} \rightarrow [0, 1]^3$ is the set of images, and \mathcal{D} is a set of admissible descriptions.

In order to use Galois' lattices, we restrict \mathcal{D} to discrete values. Basically, it allows to create a special kind of directed acyclic graph, the nodes of which group a set of instances, i.e., an *extension*, and a set of descriptions, i.e., an *intension*.

We derive a *Galois' lattice* from this relation. A lattice is an directed graph, without any loop, and including an *inferior node* (no vertex ends to this node) and a *superior node* (no vertex starts at this node). In a *Galois' lattice*, nodes are pairs (X, X') where $X \subset \mathcal{I}$ and $X' \subset \mathcal{D}$. We note $\mathcal{C} = \mathcal{I} \times \mathcal{D}$, the set of pairs (possible nodes). These pairs must be *complete pairs*, defined as follows⁶⁾:

Definition 7 (Complete Pair)

A pair (X, X') is complete with respect to \mathbb{R} if and only if the two following properties are satisfied:

- $X' = \{d \in \mathcal{D} | \forall i \in X, (i, d) \in R\}$
- $X = \{i \in \mathcal{I} | \forall d \in X', (i, d) \in R\}$

Only maximally extended pairs (for which there is no other pair (X_1, X'_1) such as $X \in X_1$ and $X' \in X'_1$) are kept.

Basically, that means that an image $i \in X$ features *at least* each property $d \in X'$ and a property $d \in X'$ is featured by *at least* each image $i \in X$. The X' set is called the *intension* and the X set *extension*.

The following property may easily be demonstrated:

Property 1 Given two complete pairs $((X_1, X'_1), (X_2, X'_2)) \in \mathcal{C}^2$:

$$X_1 \subset X_2 \iff X'_2 \subset X'_1 \quad (16)$$

We can now define a *partial order* between

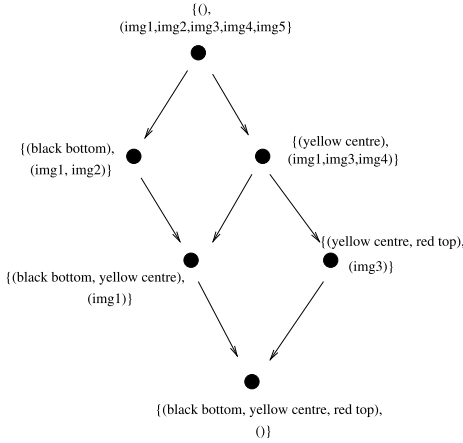


Fig. 4 An example of an image lattice.

pairs:

$$\begin{aligned} \forall (C_1 = (X_1, X'_1), C_2 = (X_2, X'_2)) \in \mathcal{C}^2, \\ C_1 < C_2 \iff X_1 \subset X_2 \iff X'_2 \subset X'_1 \end{aligned} \quad (17)$$

This partial order is used to generate the lattice graph as follows: there is an edge (C_1, C_2) if $C_1 < C_2$ and there is no other element C_3 in the lattice such as $C_1 < C_3 < C_2$.

Example 1 (Fig. 4)

From $R = \{(img_1, blackbottom), (img_1, yellowcentre), (img_2, blackbottom), (img_3, yellowcentre), (img_3, redtop), (img_4, yellowcentre)\}$

We derive: $r(img_2) = \{blackbottom\}$,
 $r(img_1) = \{blackbottom, yellowcentre\}$,
 $r(img_3) = \{yellowcentre, redtop\}$ and
 $r(img_4) = \{yellowcentre\}$;

and conversely $r'(redtop) = \{img_3\}$,
 $r'(blackbottom) = \{img_1, img_2\}$ and
 $r'(yellowcentre) = \{img_1, img_3, img_4\}$.
(See Fig. 4)

In addition, in a Galois' lattice, all the possible intersections for which at least a descendant node is non empty are added.

Intuitively, we are interested in the set of images that share *exactly* the same description as well as those sharing *at least* the same description.

4.2 Using a Galois Lattice as a Navigation Structure

Our approach does not use Galois' lattices only for indexing but also for browsing itself, from the user's point of view. A set of XHTML pages is constructed from the Galois' lattice, that is stored either in central memory or in

a database management system. These pages being generated only after a new image is inserted into the structure, the user will browse a set of static pages; consequently, the navigation process is optimal.

For one node of the Galois lattice, corresponding to an *intention* (a set of descriptions) and an *extension* (a set of images featuring at least these descriptions), one XHTML page is generated.

The intention is not displayed to the user. We consider that the user should make up his or her choice by visualising sample images, not by abstracting his or her needs. Consequently, a node is represented by a set of images.

Rather than using the complete extension of the node to represent it, we use the *reduced content* of the node.

Definition 8 (Reduced content)

The reduced content Y of a node (X, X') is defined as:

$$Y = \{i \in \mathcal{I} \mid \forall d \in X', (i, d) \in R\}$$

It means that the reduced content of a node is the set of images that features all the descriptions of its intention, and only these descriptions. Note that since the extension X is the set of images featuring *at least* the descriptions in the intention X' , Y is subset of X . The calculation of the reduced content has been integrated into Godin's incremental lattice building algorithm⁶, and does not change its complexity.

A node is represented as follows:

- When the reduced content of a node is not reduced to the empty set, these images are used to represent this node.
- When the reduced content is reduced to the empty set, the node's representation is built recursively from the reduced content of its children nodes.
- In the top of the page, links to father nodes using these nodes' representations are provided.
- Similarly, in the bottom of the page, links to children nodes using these nodes' representations are provided.

Figure 1 shows the graphical user interface of our prototype $Click_{AGE}^{Im}$. It shows a part of the lattice representing images with a lot of black and orange, among which most are quite naturally sunsets.

5. From a Fuzzy Model to Crisp descriptions

We have seen that a fuzzy model is adapted

to describing images (based on numerical features), but to build a Galois' lattice we need a binary relationship: a given property is either present, or not in a given image. This is true not only for a Galois' lattice but also for any navigation structure that does not weight links between (groups of) images. Even if the links are weighted, one should set a threshold to decide whether a neighbour should appear or not.

The most trivial answer, the one used in the first prototype of *Click_{AGE}^m*¹¹⁾, is to rely on a constant threshold, determined empirically. Static or dynamic, the threshold has to be chosen in order to satisfy the following two conditions:

- Adequacy to human perception: a human observer should mostly agree with the system to see a property as “present” or not. This can be obtained by a form of training or just statistical analysis.
- Efficiency of structure: the threshold should produce a number of properties per instance that is neither too small (it would lack precision), nor too big (this would hurt performances as well as prevent the system to discover similarities since each image would be too specific). Objective criteria such as the performance measure exist; indeed, subjective criteria such as the satisfaction of the user, or the ease of browsing are important but require exhaustive and costly experiments.

Our first attempt to reduce noise in the dataset relied on an estimation of the resulting structure by taking the whole fuzzy relationship into account. A small (i.e., permissive) threshold was applied to properties present in few images, whereas a high (i.e., severe) threshold was applied to properties present in a lot of images. Thus, supposing that an ideally discriminant property is a property that divides the dataset into two equal subsets (subset of images featuring the given property and subset of images not featuring the given property), we expected all the properties to gain in discriminative power. Unfortunately, experiments showed a side-effect that we did not expect beforehand: most images featured a similar number of properties, and consequently the relevant subpart of the lattice is even more reduced. Regarding the bad results of this approach, we do not present it here.

5.1 Key Matching Fuzzy Insertion

In this section, we propose a technique to

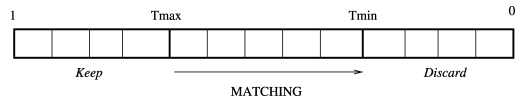


Fig. 5 Key matching.

build a Galois' lattice from a fuzzy relationship. Here, we do not apply a constant threshold to convert the fuzzy descriptors into crisp ones but rather build incrementally the lattice and select which descriptions to use *according to the existing lattice*. The goal of this procedure is to avoid noise and build a lattice with compact nodes in order to get an efficient navigation structure.

Hereafter, we call “key” the set of descriptions associated to a given image. The following algorithm is used in order to determine which descriptions should be used when inserting a given image into an existing lattice:

- (1) Descriptions are sorted in descending order of membership degree (see Fig. 5);
- (2) Descriptions under a T_{\min} threshold are discarded and descriptions over a T_{\max} threshold are always kept;
- (3) Starting from the set of descriptions over T_{\max} , we add successively the descriptions in the interval $[T_{\max}, T_{\min}]$ and try to match the longest resulting key with existing keys in the lattice;
- (4) When a matching key is found, this key is used to insert the image;
- (5) If no matching key is found, the image is inserted with the set of descriptions over the T_{\min} threshold.

Since the number of properties for a given image is barely constant, this algorithm has a linear complexity, $O(n)$, where n is the number of nodes of the existing lattice. It can easily be optimised by keeping an index of the nodes on the key.

The benefit of this algorithm is as follows: in order to reduce the number of nodes, this algorithm ensures that a given image is inserted, whenever possible, into an existing node. By reducing the number of nodes, we make the navigation process more efficient. Additionally, since the Galois' lattice is used to search for existing keys, there is no additional algorithmic cost.

6. Evaluation and Experiments

The proposal being mostly heuristic, we have to check its efficiency and effectiveness on real data.

6.1 Experimental Protocol

Based on the same code that was written to test Galois' lattices for the basic navigation process, we conducted experiments (1) to determine if our extension actually gives the expected results, and (2) to ensure that there is no increase of the time complexity.

We prepared a set of about 3,000 images, extracted from the Internet database Flickr.com. Flickr.com is a web site where anyone can upload their photographs to share them with their relatives, or anonymous visitors, and add key word annotations, so-called *tags*, such as "Tokyo", "wedding", or "dog." We randomly selected these 3,000 images under nine tags among the most popular ones (Flickr proposes such a list): *art, city, flower, party, sunsets, birthday, dog, snow, and nature*.

By using as a source a web site where anyone can post his or her own photographs, we are sure to use a *real world* data set. All the images are photographs that people actually take, and have been chosen because they represent a tag that is popular. Consequently, we consider our data set as being quite representative of the photographs that need to be indexed at home.

On this set, as well as on subsets, a Galois' lattice is constructed for different values of the parameters. Several metrics are calculated on these lattices in order to evaluate the quality of the lattice in term of usability.

6.1.1 Metrics

We define the following metrics on a lattice:

- Cardinal: the number of nodes of the lattice; since information is hard to find in a too large structure, we prefer a lattice with a limited number of nodes.
- Average size of nodes: the size of a node is defined as the number of images in its *reduced content*. The reduced content of a node is defined in Section 4.2; we use it rather than extension in order to match the representation of the nodes from the user's point of view, also described in the same section.
- Ratio of non-empty nodes: we call empty node a node, the *reduced content* of which is reduced to the empty set. This metric shows the proportions of nodes actually containing data in the lattice, the empty nodes being only useful for navigating.

6.2 Results

We made measures on the key matching fuzzy insertion; we then compared it to the triv-

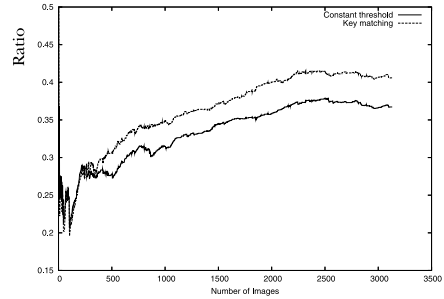


Fig. 6 Ratio of Non-Empty Nodes in the Lattice With $\text{Ratio} = \frac{\text{Number of non empty nodes}}{\text{Total number of nodes}}$.

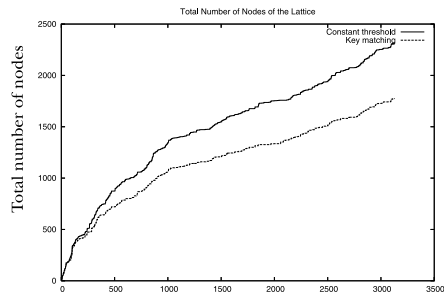


Fig. 7 Total Number of Nodes.

ial technique, based on a constant threshold. For constant insertion, we chose a threshold of $T = 0.3$. For key matching insertion, we chose $T_{\min} = 0.2$ and $T_{\max} = 0.4$. These values were determined empirically, trying to match human perception, and we chose on purpose the threshold of constant insertion as the average of T_{\min} and T_{\max} . In this way, we ensure that the difference observed is not due to a difference in the threshold choice.

Key matching fuzzy insertion demonstrates a clear difference with respect to constant threshold. **Figure 6** shows the percentage of non-empty nodes, i.e., nodes that actually contain information, while other nodes are only present for a navigational purpose. (Of course, with only one image, the lattice is reduced to a unique node that is not empty. Consequently, when the number of images is 1, the ratio is 1; however, in order to have a better representation of other parts of the graph, we chose a scale that stops at $y = 0.5$.)

Figures 7 and **8** respectively show the total number of nodes and the average size of a node. The total number of nodes is clearly lower for key matching insertion; in this sense the structure will be easier to browse than the big structure built by the trivial algorithm. Having less nodes, the lattice built by key matching obvi-

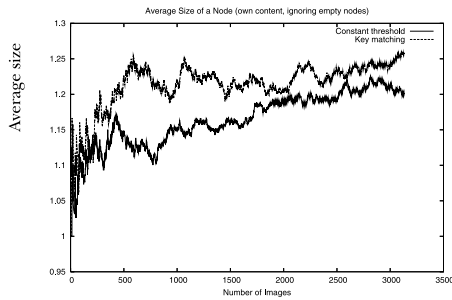


Fig. 8 Average size of a node (excluding empty nodes).

ously has bigger nodes. However, the size of each node is still small enough for a comfortable browsing.

When looking at Fig. 8, we notice that there are several local extrema on both curves. Moreover, local extrema on the lower curve, representing the constant threshold approach, do not correspond to local extrema for the key matching upper curve. For example, around 700, there is a local maximum for the key matching approach while the curve for constant threshold is still decreasing. In fact, after the 693rd image, the insertion of a new category starts: images tagged with *flower* (after *art* and *city*). Photographs related to *flower* are very specific: the border is green representing leaves or grass, and the centre is colourful representing the flower itself. Consequently, new descriptions and new combinations appear, forcing the key matching algorithm to create new nodes reduced to one element: the average size of a node decreases. Shortly after, when enough nodes have been created, the average size of a node increases again. Before the 693rd image, when images were quite similar, the size of each node was increasing for the key matching algorithm whilst the constant threshold algorithm failed to see similarities in the descriptions.

Unfortunately, user's experience is not as easily quantifiable. Currently, our personal experience as well as the comments of a few volunteers confirm the statistical results: retrieving images from a lattice built with key matching is easier than with constant threshold. The reason is mainly that the number of nodes has been reduced, in particular there are fewer empty nodes. Thus, browsing from one part of the lattice to another is both faster and more informative.

Figure 9 shows the time to build the Galois' lattice. First of all, it confirms Godin's

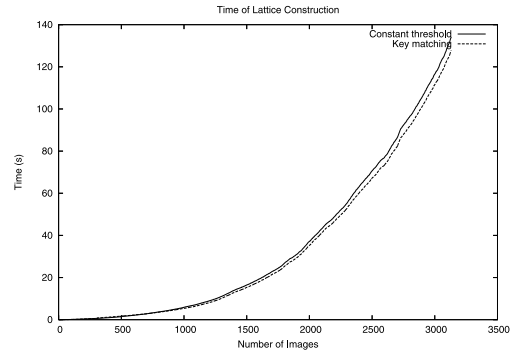


Fig. 9 Construction time.

assertion that the empiric complexity of his algorithm is linear, $O(n)$, for adding one element, thus quadratic, $O(n^2)$, for building a whole lattice. Secondly, it confirms that using *key matching* does not change the complexity of the algorithm.

6.3 Galois' Lattices as a Navigation Structure and Relevance Feedback

Since most content-based image retrieval systems (CBIR) are based on separate models for the indexing and retrieval processes, they cannot be easily compared to our proposal based on a single graph structure both for indexing and retrieving.

However, from the user's point of view, our prototype can be similar to relevance feedback based proposals: at each step, a set of images are shown to the user and he or she is asked to interact with the system for it to propose a new set of images.

On the one hand, a feedback-querying system will show the user a set of images as the result of a query. For the first iteration, a random set of images (or a selected summary) is chosen. At each iteration, the user will select examples and/or counter-examples from the result set, with more or less precision according to the system (it may be simple binary flags or relevance mark from 0 to 1). From these pieces of information, the system will formulate a new query and present the result to the user.

On the other hand, our system, based on a unique indexing and browsing structure, will propose to the user a node of its structure. The user is then free to browse the structure either by asking for specific images (by navigating towards the *max* node, that features all descriptions) or for general images (by navigating towards the *min* node, that features no description).

From the performance point of view, our proposal is more costly for image insertion, but optimally efficient during the browsing process: a feedback querying approach needs to write a query at each iteration and execute it (a very costly process) while in our proposal all the calculations are done off-line, making the browsing process fast and responsive.

From the user's point of view, the relevance feedback approach requires the user to be somewhat precise about the kind of images that he or she is looking for, thus having to choose carefully which images are actually close to his or her visual requirements. Our proposal offers an efficient way to visualise the whole collection. Consequently, relevance feedback is more precise and may be faster for a user looking for very specific images that actually exist in the collection. In contrast, when the user has few knowledge of the collection and is unsure about which kind of images can best suit his or her needs, an approach based on navigation gives a better overview of the collection.

7. Conclusion

In this paper, we investigate the problem of building an efficient navigation structure for an image collection. We relied on a Galois' lattice computed off-line. Building such a structure rather than computing a query at each step of the search process is much more efficient. However, optimising it both for efficiency and effectiveness is challenging.

This paper presented a way to improve the "quality" of a Galois' lattice by reducing noise while constructing it. Being based on a static structure, the retrieval process cannot be more efficient. (Moreover, we addressed the scalability problem elsewhere.) Thanks to the introduced noise reduction technique, the average time of user's activity to find a given image is also reduced since the number of – small – nodes has been reduced without sacrificing the intrinsic qualities of a Galois' lattice, nor augmenting the time complexity to build it.

Indeed, this proposal does not suppress more information than do the trivial method, and ensures that (1) a property is discarded if its membership degree does not exceed a certain threshold T_{\min} and (2) a property is not be discarded if its membership degree exceeds another threshold T_{\max} . In addition, this technique avoids most of the local exponential that could occur in a Galois' lattice without it.

Experimentally, it produced good results and validated our hypothesis made prior to the implementation. Not only were we able to reduce the complexity of the navigation structure without losing more information than existing methods, but this technique even reduced the empirical complexity of the Galois' lattice construction, which is particularly high.

References

- 1) Ahamd, I. and Jang, T.-S.: Old fashion text-based image retrieval using fca, *Proc. 2003 International Conference on Image Processing* (2003).
- 2) Bach, J.R., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R.C. and Shu, C.: Virage image search engine: An open framework for image management, *Proc. IS&T/SPIE International Symposium on Electronic Imaging: Science and Technology, Storage & Retrieval for Image and Video Databases IV (EI'96)*, pp.76–87 (1996).
- 3) Jin, J.S. and Kurniawati, R.: A scheme for intelligent image retrieval in multimedia databases, *Journal of Visual Communication and Image Representation*, Vol.7, No.4, pp.369–377 (Dec. 1996).
- 4) Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D. and Yanker, P.: Query by image and video content: The QBIC system, *IEEE Comput.*, Vol.28, No.9, pp.23–32 (Sep. 1995).
- 5) Garzotto, F., Paolini, P. and Schwabe, D.: HDM — A model-based approach to hypertext application design, *ACM Trans. Inf. Syst.*, Vol.11, No.1, pp.1–26 (Jan. 1993).
- 6) Godin, R., Missaoui, R. and Alaoui, H.: Incremental concept formation algorithms based on Galois (concept) lattices, *Computational Intelligence*, Vol.11, No.2, pp.246–267 (1995).
- 7) Isakowitz, T., Stohr, E. and Balasubramanian, P.: RMM: A methodology for structured hypermedia design, *Comm. ACM*, Vol.38, No.8, pp.34–44 (Aug. 1995).
- 8) Käster, T., Pfeiffer, M., Bauchhage, C. and Sagerer, G.: Combining Speech and Haptics for Intuitive and Efficient Navigation through Image Databases, *Proc. International Conference on Multimodal Interfaces (ICMI'03)*, pp.180–187, ACM (2003).
- 9) Loisant, E., Saint-Paul, R., Martinez, J., Raschia, G. and Mouaddib, N.: Browsing clusters of similar images, *Actes des 19e Journées Bases de Données Avancées (BDA'2003)*, pp.109–128, Lyon, France (Oct. 2003).

- 10) Ma, W.Y. and Manjunath, B.S.: NETRA: A toolbox for navigating large image databases, *Proc. IEEE International Conference on Image Processing (ICIP'97)* (1997).
- 11) Martinez, J. and Loisant, E.: Browsing image databases with Galois' lattices, *Proc. ACM International Symposium on Applied Computing (SAC'02)*, pp.971–975, Madrid, Spain, March 11–14 2002, ACM Computer Press (2002).
- 12) Nastar, C., Mitschke, M., Meilhac, C. and Boujemaa, N.: SurfImage: A flexible content-based image retrieval system, *Proc. 6th ACM International Multimedia Conference (ACM-MM'98)*, Bristol, UK, September 14–16 1998, ACM, ACM Press (1998).
- 13) Saint-Paul, R., Raschia, G. and Mouaddib, N.: Prototyping and browsing image databases using linguistic summaries, *Proc. IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'2002)*, Honolulu (Hawaii), USA (May 2002).
- 14) Santini, S., Gupta, A. and Jain, R.: Emergent semantics through interaction in image databases, *Knowledge and Data Engineering*, Vol.13, No.3, pp.337–351 (2001).
- 15) Santini, S. and Jain, R.: Integrated browsing and querying for image databases, *IEEE MultiMedia*, Vol.7, No.3, pp.26–39 (2000).
- 16) Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A. and Jain, R.: Content-based image retrieval: The end of the early years, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* (2000).
- 17) Smith, J.R. and Chang, S.-F.: Tools and techniques for color image retrieval, *Storage and Retrieval for Image and Video Databases IV*, San Jose, CA, USA (1996).
- 18) Zenou, E. and Samuelides, M.: Characterization of image sets : The galois lattice approach, *Reconnaissance des formes et d'Intelligence artificielle (RFIA2004)* (2004).

(Received June 19, 2005)

(Accepted August 12, 2005)

(Editor in Charge: *Tatsuo Tsuji*)



Erwan Loisant is a Ph.D. student under a co-direction from the Polytechnic School of the University of Nantes and Tokyo Metropolitan University Graduate School of Engineering. He has been working on browsing through multimedia collections.



José Martinez is a full professor at the Polytechnic School of the University of Nantes, France. He received his Ph.D. from the University of Montpellier II, France, in 1992. He obtained his D.Sc. in 2002 from the University of Nantes. His current interests are multimedia content indexing, intelligent retrieval and browsing, and efficient DBMS implementations.



Hiroshi Ishikawa received the B.S. and Ph.D degrees in Computer Science from the University of Tokyo. After working for Fujitsu Laboratories, he is a professor of Tokyo Metropolitan University. His research interests include database, Web, and e-commerce. He has published actively in international, refereed journals and conferences, such as ACM TODS, IEEE TKDE, VLDB, IEEE ICDE. He authored some books, which include books entitled “Object-Oriented Database System” (Springer-Verlag) and “Next-Generation of Databases and Data Mining” (CQ Publishing). He received the Sakai Memorial Distinguished Award from Information Processing Society of Japan (IPSJ) and the Director General Award from Science and Technology Agency of Japan. He was an invited professor at the Polytechnic School of the University of Nantes, France in 2003 and 2005. He is a trustee board member of the Database Society of Japan and was an editorial board member of VLDB Journal. He is the chairman of the SIG on Database Systems of IPSJ and an editor-in-chief of IPSJ Trans. on Databases.



Kaoru Katayama received the Ph.D. degree in Informatics from Kyoto University in 2000. He is a research associate of Tokyo Metropolitan University. His research interests include data mining and query optimization. He is a member of IPSJ.