User-Adaptive Navigation Structures for Image Retrieval

Erwan LOISANT♣ Hiroshi ISHIKAWA♦
José MARTINEZ♣ Manabu OHTA♦
Kaoru KATAYAMA♦

Galois' lattices have been shown to be a convenient navigation structure for visualization. However, experiments showed that after a few hundreds of images the number of children and parents for a given node is likely to increase and to lead to user confusion. In order to deal with larger sets of images, we introduce a linear complexity algorithm to define a sub-lattice that will still respect the lattice axioms and thus will remain an acceptable navigation structure. The result is a user-adaptive, usage-adaptive and cheap-to-build structure showing only most relevant images and links, thus increasing usability.

1. Introduction

Unlike classical textual data that can easily be stored to and retrieved from classical database management systems, multimedia data can not be normalized [11]. These kind of data contain a variable density of information, moreover information depends on the observer and the context. Image is one of most studied multimedia data; human annotation being subjective and costly, it is not enough for retrieval through an image collection. Thus, content-based image retrieval (CBIR) systems appeared using information extracted from content. First CBIR used concepts from classical DBMS to query a database of content-based information [5], then similarity based querying and relevance-feedback querying [10] appeared. Finally, navigation-based systems aimed at producing a user-friendly and fast system, yet loosing precision over query-based retrieval [12]. Now, work is still done in these three approaches that are complementary.

Navigation through a before-hand calculated structure has shown good results: it provides a fast and intuitive way to retrieve information from an image collection. However, each user has his own goal while looking for an image, and different goals should have different structures to optimize retrieval.

♣Student Member Graduate School of Engineering, Tokyo Metropolitan University - Polytechnic School, Nantes University
erwan@loisant.org

♦Regular Member Graduate School of Engineering, Tokyo Metropolitan University
{ishikawa, ohta, katayama}@eei.metro-u.ac.jp

♣Polytechnic School, Nantes University
jose.martinez@polytech.univ-nantes.fr

In this paper, we propose to apply masks on navigation structures, i.e. hide parts of the graphs or connections to display a subgraph closer to user's expectation. This is done by keeping an underlying structure common to all users and all retrieval processes, keeping the advantages of a before-hand calculated structure: the most costly processes are done before-hand, and retrieval itself is fast and reactive. This results in improving our existing prototype by adding user customization without denying the performances advantages. Our proposal is consequently more efficient than systems based on feed-back querying or similarity search, and more relevant than systems based solely on a pre-calculated structure.

First, in section 2, we present the intrinsic information we extract from images to describe them. In section 3 we give a quick introduction on Galois' Lattices, then in section 4 we propose a general scheme for applying masks on a particular navigation structure based on Galois lattice. Finally in section 5 we propose a masking technique based on this scheme.

2. Meta-data representation

There are several data to take care of in order to organize adequately an image database [1]. The standardization effort of MPEG-7 [8], [9] separates meta-data into several levels, from physical information to perceptual one, as well as manual annotations and transcriptions. Our study focuses on structural information and perceptual information: a general segmentation of image and dominant colors on these parts.

2.1. Color models

Color is known to be a tri-dimensional parameter, however several models exist. The HSV color model, used in this work, is recognized to be one of the most perceptually evident for users [4]. HSV stands for *Hue*, *Saturation* and *Value*. In this model *pink* is seen as a red Hue with some white in it to decrease its saturation.

2.2. Zone color characterization

Color perception results from the juxtaposition of individual pixels. The perceived color of an arrangement of pixels ranges from uniform pure color to complex color arrangement without dominating color. Considering our representation of colors, each pixel color is expressed in terms of color labels with different weights. For a pixel, the weight is the membership degree of its color to the fuzzy set associated to a color label. For instance, in our paradigm of representation, a pink pixel could be defined as two color labels: *vivid bright red* with a membership degree of 0.1 and *dull bright red* with a membership degree of 0.9.

For the need of Galois' lattices (further described in section 3), the properties have to be keywords. Thus, we make this relationship binary by considering a color *present* is its relative is above an experimentally fixed threshold .

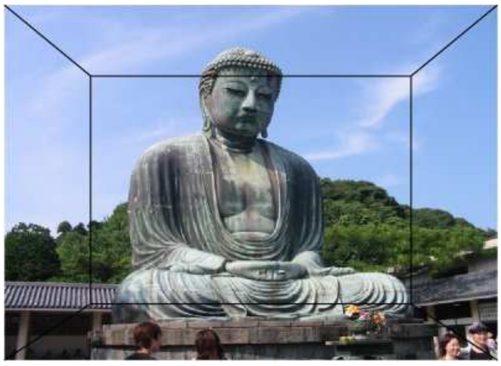

Figure 1: Five parts segmentation

2.3. Segmentation

An image segmentation is used to allow a more accurate description of image colors. Considering general photographic pictures, the main subject often stands in the center and the surrounding areas represent the image background.

In addition, color homogeneity is expected to be enhanced if smaller zones are considered. In a landscape picture for instance, the sky is likely to have blue or gray hues, while the ground will probably be green. In our tests, we used a five zone segmentation. The center zone covers 49% of the total surface and the four surrounding zones are trapezoids whose wideness is 15% of the image wideness.

Figure 1 shows an example image of Kamakura's Big Buddha. Using a discrete description of the colors (a color being simply present or not), this image would be represented using the following properties: (1) light vivid blue top, (2) light unsaturated yellow center, dark unsaturated red center, (3) light vivid blue left, dark saturated green left, black left, (4) light vivid blue right, dark saturated green right, black left, (5) and black bottom.

3. Galois' Lattices

This part gives a quick introduction to Galois' lattices, mainly to precise axioms that we will have to respect while applying filter on it and to introduce notation that will be used later. Interested reader may refer to [8] where Galois' lattices applied to image retrieval are detailed.

A Galois' (or concept) lattice is a mathematical structure that has been largely exploited in the field of knowledge discovery [5]. It can be defined whenever there is a binary relation, in our case between *images* and their associated *meta-data*: $R : I \times D$ where I is the set of images, and D is a set of descriptions. Note carefully that a Galois' lattice can be defined only over discrete domains. Also, meta-data descriptions vary from application to application. They can be related to the intrinsic content of the images, e.g., color, or they can add some semantics to them, e.g., through mere keywords.

A lattice being a directed acyclic graph featuring a minimal node (*inf*) and a maximal node (*sup*), a Galois' lattice is a special kind of lattice derived from a binary relationship.

The problem of updating a Galois' lattice is not trivial, since it is necessary to generate not only the new pairs and its connections but usually several other pairs needed to respect the Galois' lattice definition. [4] proposes an incremental algorithm that has an exponential complexity in the worst case. However, in most case we experience a linear complexity for adding one instance.

A Galois' lattice will be noted (N, E), where N is a set of nodes and E a set of oriented edges.

4. Masking lattices

The time complexity of the Galois' lattice construction algorithm being experimentally $o(n^2)$ [4], it allows to reach a size of 10,000 instances [7]. In this case, a node explosion can happen and the path to the wanted image may be long. Moreover, if descriptions are randomly distributed on the image set, the number of edges can be very important and lead to confusion when user is to choose between too many children nodes. In order to reduce the number of node by hiding only non-relevant one, and by limiting processing time, we propose to take the original Galois' lattice as a base to apply a mask.

A mask is a filter applied to a given Galois' lattice to hide elements, that can be nodes or links. It should be noted that while the resulting graph may not be a Galois' lattice since it will not represent a binary relation between two sets, it has to be a lattice. Since user will browse a direct representation of the resulting graph, every lattice axiom is mandatory to ensure that this browsing will allow user to reach every non-masked image in a natural navigation path.

Different kinds of masking serve different goals. For example, one may want to reduce the cardinal of the images set or the cardinal of the description set. However, any kind of masking is represented in the same way.

Definition: Given a lattice (N, E), a *lattice mask* M is defined as $M = (N_M, E_M, E_A, N_{Me})$ where $N_M \subset N$, $E_M \subset E$, $E_A \subset N^2$ and $N_{Me} \subset N_M^{\ 2}$. Also, N_{Me} is such as $\forall (N_1, N_2) \in F_M$, N_1 is a father node of N_2.

N_M represents the set of nodes to be masked, E_M the set of edges to be masked, E_A the set of edges to be added and N_{Me} the set of pair of nodes to be merged.

5. Masking techniques

In this section, we present two kind of filtering, both with different goals: *node masking* and *edge masking*. Node masking consists in masking some set of images if the system already have informations about what kind of images are relevant to current retrieval and which images are not. Applying such a filtering will result in hiding complete nodes to user if most of its members are irrelevant to current search. On the contrary, edge masking consists in masking links if the relation represents a description irrelevant for this search.

Both masking techniques results in masking both *nodes* and *edges*. However, we call "node masking" a masking where we want to mask nodes (to mask edge being a consequence) and "edge masking" a masking where we want to mask edge (to mask node being a consequence).

5.1. Node masking

The system operates a *node masking* when it has gathered informations about what kind of images user is looking for, enough to reduce the number of images to propose but not enough to give user a final result. Node identified as irrelevant to current retrieval should be masked.

A node masking operation is defined by a node filtering function f on nodes: $f : N \rightarrow \{0, 1\}$. The selection of nodes to mask is done by asking user for examples of images to be masked, and inferring an approaching query. To ensure good performance, a low-complexity algorithm is chosen over better but high-complexity algorithms used in systems mainly based on relevance feedback.

Algorithm

Considering a node filtering function f and a Galois lattice $G = (N, E)$, we note $N_F = \{n \in N \mid f(N)=0\}$ the set of nodes to mask. The following gives a algorithm to determine a mask $M = (N_M, E_M, F_M)$ that applied to G results in a lattice according to section 3.

```
Nm <- Nf \ {min(G), max(G)};
FORALL n in Nm:
  FORALL e connecting n:
    add e to Em;
  FORALL p, parent node of Nf:
    CASE cardinal(non_masked_children(p)):
     0: FORALL c, child of Nf:
          add (p, c) to Ea;
     1: IF c, unique children of p
                   has no other parent:
          add (p, c) to Fm; add (p, c) to Em;
     else: nothing
  FORALL c, children node of Nf:
    CASE cardinal(non_masked_parent(c)):
     0: FORALL p, parent of Nf:
          add (p, c) to Ea;
     1: IF p, unique parent of c
                   has no other child:
          add (p, c) to Fm; add (p, c)to Em;
     else: nothing
```

Actually, this algorithm performs the following operations: (1) The set of nodes to mask will be equal to the set of nodes defined by the filtering function, except that the minimum and maximum nodes can not be masked, (2) any edge connected to a masked node will be masked, (3) if a node other than *min(G)* ends with no parent, it should be connected to all parent of its last former parent, (4) if a node other than *max(G)* ends with no child, it should be connected to all child of its last former child, (5) if a node ends with a unique child and this child has a unique parent, these nodes should be merged, and (6) if a node ends with a unique parent and this parent has a unique child, these nodes should be merged.

The complexity of this algorithm depends on the number of nodes to mask, and the average number of parents and children a node can have. Experimentally, we noticed that this number does not exceed a certain maximum. Indeed, since the low-level properties are correlated regarding their semantic meaning, we noticed that the number of children for a given node doesn't reach the number of properties but is at worst 25% it. Thus, we conclude that this algorithm has an empiric linear complexity according the number of nodes to mask, i.e. o(n). This complexity is acceptable regarding the number of nodes to consider.

If a node had more than one parent, and all of them are masked in the process, then the result will depend on the last node masked by the algorithm. Since the order to process nodes is arbitrary chosen, this algorithm is not deterministic. However, parent nodes sharing all the same role, we do not see that point as a issue. There is a symmetric problem when masking children.

5.2. Edge masking

While node masking aims at changing the content to show to user, the goal of edge masking is to change the link between elements to match user's needs. Several ways can be considered to gather information about and decide which properties may be less relevant to user: analyze user's way of navigating through the structure, explicitly ask for properties to be ignored, etc. For example, a user navigating through the system may be not interested in the color of the *upper part* of the image. Thus, links between nodes related to this information will be considered as noise and masking them would improve relevance of the navigation results.

Our research about edge masking is still in progress, and to complete node masking in a middle term.

6. Implementation and Evaluation

Our prototype can be divided into two parts: (1) the before-hand structure calculation, which results in a set of XHTML pages directly readable by a standard-compliant web browser and (2) the customization system, implemented as a browser extension operating client-side processing. The second part is still in development. Figure 2 shows a screenshot of browsing a navigation structure using a web browser. The center part is a view of current node, while the higher and lower part are respectively the fathers nodes and the children nodes. By clicking in the lower or higher part in the set of images she likes, the user can respectively specialize or generalize her query.

After achieving implementation work, an evaluation protocol is to be applied. Several sets of images will be prepared, and metrics will be calculated.

The characteristics of the different sets of image will vary: (1) randomly distributed sets and sets containing several homogeneous subsets (such as sunset, nature images, urban images, different views of the same object), and (2) small sets (a few hundreds of images) and large sets (a few thousands of images). Having homogeneous subsets should give better results, but randomly distributed sets can appear in a real world so should not be excluded.

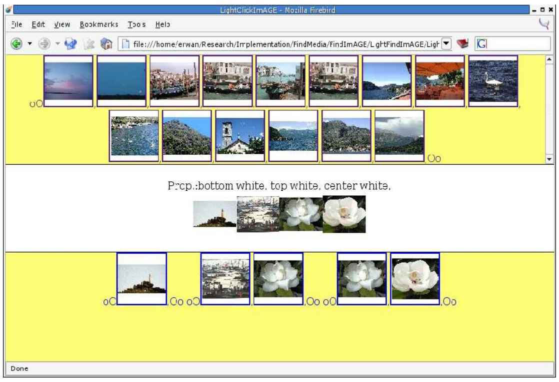

Figure 2: Graphical navigation interface

Working on these sets, we will compare the navigation structure obtained without customization (corresponding to previous work) to a structure customized from one to three times. Metrics will include: (1) number of images per nodes, (2) average number of children of a node and (3) shortest path from the inferior node to the superior node. For all of these metrics, a small value is considered as better since it reduce user's disorientation.

We expect that each customization iteration greatly improves the system, and that a lattice featuring several thousands of images difficult to browse without customization can become usable with customization.

7. Conclusion

In this paper we presented a technique to ease navigation through a large Galois' lattice. Using a before-hand calculated structure and applying to it a linear complexity algorithm, we ensure to keep better performances than relevance feedback or similarity query. Experimentation is still to be done, however we expect that this techniques greatly improves user's experience by reducing (1) the number of images simultaneously displayed on the screen (2) the links he has to choose to specialize or generalize the query. Further work will include developing new masking methods on the same framework, that would not focus on which images should be masked but which concepts, or links between images.

References

[1] D. H. Ballard and C. M. Brown: "Computer Vision". Prentice-Hall, 1982. 523p. (1982).

[2] J. M. Corridoni, A. Del Bimbo, and E. Vicario: Image retrieval by color semantics with incomplete knowledge. Journal of ASIS, 49(3):267--282 (1998).

[3] T. Gevers and A. Smeulders: A comparative study of several color models for color image invariant retrieval (1996).

[4] R. Godin, R. Missaoui, and H. Alaoui: Incremental concept formation algorithms based on Galois (concept) lattices. Computational Intelligence, 11(2): 246--267 (1995).

[5] P. M. Kelly, T. M. Cannon, and D. R. Hush: Query by image example: The CANDID approach. In SPIE SRIVD III, volume 2420, pages 238--248 (1995).

[6] J. Martinez and S. Guillaume: Colour image retrieval fitted to classical querying. NISJ, 1(2-3):251--278 (1998).

[7] J. Martinez and E. Loisant: Browsing image databases with Galois' lattices. ACM SAC'02, pages 971--975 (2002).

[8] F. Nack and A. Lindsay: Everything you wanted to know about MPEG-7: Part 1. IEEE Multimedia, 6(3): 65--77, July/September (1999).

[9] F. Nack and A. Lindsay: Everything you wanted to know about MPEG-7: Part 2. IEEE Multimedia, 6(4): 64--73, October/November (1999).

[10] C. Nastar, M. Mitschke, and C. Meilhac. Efficient query refinement for image retrieval. (CVPR'98), Santa Barbara, California, June 23-25 (1998).

[11] S. Santini and R. Jain: Beyond query by example. ACM-MM'98, Bristol, UK, September 14-16 (1998).

[12] Simone Santini and Ramesh Jain: Integrated browsing and querying for image databases. IEEE MultiMedia, 7(3):26--39, -- (2000).

Erwan LOISANT

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

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 is a trustee board member of DBSJ and 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.

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 interest being on multimedia content indexing, intelligent retrieval and browsing, and efficient DBMS implementations.

Manabu OHTA

Received B.E., M.E. and Dr. Eng. in Electrical Engineering from the University of Tokyo in 1994, 1996 and 1999, respectively. He is a research associate of Tokyo Metropolitan University. His research interests are information retrieval and its application to Web systems. He is a member of IPSJ and IEICE.

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.