ユーザカスタマイズ可能な画像検索のためのナビゲーション構造

ルワゾン・エルワン†,††　石川博†　マルチネーズ・ジョゼ††　太田学†　片山薫†

† 東京都立大学大学院工学研究科
†† ナント大学エコール・ポリテクニック（仏）
E-mail: †erwan@hiver.eei.metro-u.ac.jp, ††{ishikawa,ohta,katayama}@eei.metro-u.ac.jp,
†††jose.martinez@polytech.univ-nantes.fr

User-Adaptative Navigation Structures for Image Retrieval

Dynamic Galois' Sub-lattices

Erwan LOISANT†,††, Hiroshi ISHIKAWA†, José MARTINEZ††, Manabu OHTA†, and Kaoru KATAYAMA†

† Tokyo Metropolitan University, Graduate School of Engineering
†† Nantes University, Polytechnic School (France)
E-mail: †erwan@hiver.eei.metro-u.ac.jp, ††{ishikawa,ohta,katayama}@eei.metro-u.ac.jp,
†††jose.martinez@polytech.univ-nantes.fr

Abstract Navigation is along with similarity search and feedback querying one of the main approaches studied to query image databases. Galois' (concept) lattices have shown to be a convenient navigation structure for visualization while operating a simple classification. However, if the n^2 complexity of the lattice construction algorithm allows to reach about 10,000 images, 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 make it easy to browse a large lattice, we propose a technique to hide some sets of images and some links between them according to information users give on which kind of images may be relevant. 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-adaptative, usage-adaptative and cheap-to-build structure showing only most relevant images and links, thus increasing usability.

Key words Multimedia, information retrieval, personalization, user interfaces, Galois lattices

1. Introduction

Unlike classical textual data that can easily be stored to and retrieved from a relational database management system (RDBMS) or object-oriented database management system (OODMBS), multimedia data (like video or sound) can not be normalized [14]. 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; if first image retrieval systems were based solely on external annotation, rely on human annotation is limited: not only manual annotation is very costly, but it remains a subjective annotation. Two annotators will produce a different annotation on the same image, even the same annotator would produce different annotations if he is asked to annotate the same image at different times.

Consequently content-based image retrieval (CBIR) systems appeared using information automatically extracted from content. First CBIR used concepts from RDBMS and OODMBS to query a database containing content-based information [7], [8]. Describing an image using content-based information being particularly difficult for non-expert user, similarity based querying and relevance-feedback querying [10], [13] appeared. Instead of asking user for a query, the system infers it from examples and counter-examples. It was a improvement over querying but inferring algorithms were usually costly non-deterministic algorithms. Finally, navigation-based retrieval aimed at producing a very user-friendly and fast retrieval system, yet loosing precision over query-based retrieval [15], [9]. Now, research is still done in these three approaches that are complementary.

Navigation threw 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.

In this paper, we propose to apply masks on navigation structures; that means hide parts of the graphs or connections to display a subgraph closer to user's expectation, taking user's needs and specificities into account. This is done by keeping an underlying common structure to all users and all retrieval processes, consequently 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 called ${Click_{Im}}^{AGE}$ by adding user customization without denying the performances advantages. Our proposal is consequently more efficient than system based on feed-back querying or similarity search, and more relevant than system 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 techniques 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 [11], [12] separates:

(1) format information (stereo for audio, infra-red for image...),

(2) physical information (sound energy, main colors...),

(3) perceptual information (male voice, hot colors...),

(4) structural information (splitting a video into planes, an image into regions...),

(5) intrinsic meta-data (keywords...), and

(6) miscellaneous annotations.

Our study focuses on structural information and physical information: a general segmentation of image and dominant colors on these parts. Some studies also work on *shape* information, but we consider algorithmic complexity of segmentation algorithms too costly. Moreover, color gives a good semantic information [3].

2.1 Color models

Color is know to be a tri-dimensional parameter, however several models exist.

For computer image manipulation, technical color models like RGB or CMYK are preferred. Those models reflect the way pixel's colors are produced by the rendering device. Those technical models however are not suitable for human intuitive color representation. The *pink* color for instance is not easy to describe in term of red, green and blue combination. More accurate models, also said *perceptual* models are then used. The first of those models was proposed by A.H. Munsell in 1915.

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. All those components are immediately understandable as they reflect the way artists compose their color: they first choose the Hue of the color from different tubes, next they set the saturation by adding white and finally set the value by adding some black. In this model *pink* is seen as a red Hue with some white in it to decrease its saturation. In the HSV space, this description is represented by the vector *pink* $= \langle 0.0, 0.3, 1.0\rangle$, with:

- $pink.Hue = 0.0$: hue is defined, on the chromatic circle, as an angle in $[0, 2\pi]$ where 0 means *red*;
- $pink.Saturation = 0.3$: the saturation scale ranges from 0 to 1;
- $pink.Value = 1.0$: the value is defined on $[0, 1]$.

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 linguistic 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;
- *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 color label d inside S, is computed as the sum of membership degrees $\mu_d(p)$ of each pixel:

$$\tau_S(d) = \frac{\sum_{p\in S} \mu_d(p)}{\sum_{d'\in\mathcal{D}} \sum_{p\in S} \mu'_d(p)}, \tag{1}$$

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

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

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 ar-

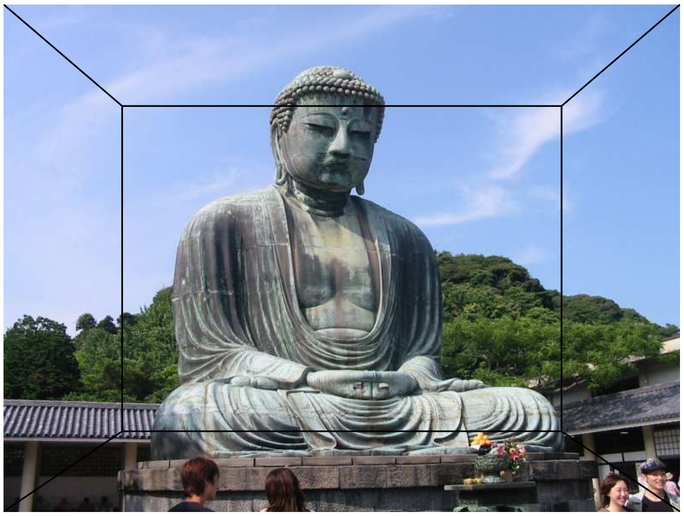

図 1　Five parts segmentation

	$blackbottom$	$yellowcenter$	$redtop$
$img1$	1	1	0
$img2$	1	0	0
$img3$	0	1	1
$img4$	0	1	0
$img5$	0	0	0

図 2　A binary relationship

eas represent the image background. 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:

- light vivid blue top
- light unsaturated yellow center, dark unsaturated red center
- light vivid blue left, dark saturated green left, black left
- light vivid blue right, dark saturated green right, black left
- black bottom

The top part shows trivial results: since the upper part is mainly made of sky, there is no surprise it is blue and will consequently be classified with other pictures featuring a clear blue sky. The left and right parts also give expected results, as well as the bottom part which is quite dark.

The result of the center part may be more surprising: it does not look red and yellow. However, the Big Buddha being made of copper, it is actually its color. It does not look red to a human observer because the saturation is low, but the *light unsaturated yellow* represents the less oxidized parts while the *dark unsaturated red* represents the more oxidized parts. It will not make sense for users, but these colors being representative of copper, other copper constructions will be classified close to the Big Buddha photograph.

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 [9] 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*:

$$R : \mathcal{I} \times \mathcal{D} \tag{2}$$

where $\mathcal{I}$ is the set of images, and $\mathcal{D}$ 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., colour, 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 relation.

Each node of this graph groups a set of instances, i.e., an *extension*, and a set of descriptions, i.e., an *intention*. From R, one derives the *Galois' connection* between $\mathcal{I}$ and $\mathcal{D}$, which consists in two dual functions, or point of views on R:

$$\begin{array}{rccl} r: & \mathcal{I} & \rightarrow & 2^{\mathcal{D}} \\ & i & \mapsto & \{d \in \mathcal{D} | (i,d) \in R\} \end{array} \tag{3}$$

$$\begin{array}{rccl} r': & \mathcal{D} & \rightarrow & 2^{\mathcal{I}} \\ & d & \mapsto & \{i \in \mathcal{I} | (i,d) \in R\} \end{array} \tag{4}$$

Intuitively, r gives the description of each image, i.e., its associated meta-data. In contrast, r' gives images featuring a given property.

The resulting graph is oriented according the following partial order:

$$\begin{array}{lccc} S: & (2^{\mathcal{D}} \times 2^{\mathcal{I}})^2 & \rightarrow & \{0,1\} \\ & ((X_1, X_1'), (X_2, X_2')) & \mapsto & (X_1 \subset X_2) \bigwedge (X_2' \subset X_1') \end{array} \tag{5}$$

The inf node and the sup node are also defined according this partial order: the inf node will be the smallest property set associated to the largest image set, and the sup node will be the largest set of property set associated to the smallest image set.

is insufficient to provide a fine description of various classes of images. Hence, a *class extension* is defined as:

$$\begin{array}{rccl} c: & \mathcal{I} & \rightarrow & 2^{\mathcal{I}} \\ & i & \mapsto & \{i' \in I | r(i') = r(i)\} \end{array} \tag{6}$$

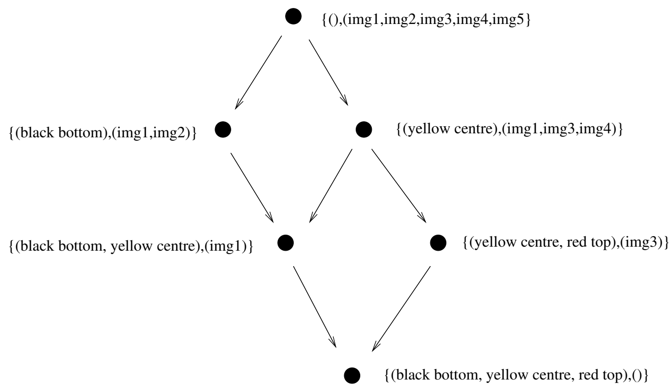

図 3　A simple example of an image lattice

Intuitively, we are interested in the set of images that share *exactly* the same description, and moreover *at least* the same description.

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. [5] 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 $(\mathcal{N}, \mathcal{E})$, where N is a set of nodes and E a set of oriented edges.

Figure 3 shows a simple example of the Galois' lattice derived from the binary relationship from figure 2

4. Masking lattices

The time complexity of the Galois' lattice construction algorithm being experimentally $o(n^2)$ [5], it allows to reach a size of 10,000 instances [9]. 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.

4.1 Formalization

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 1］　Given a lattice $(\mathcal{N}, \mathcal{E})$, a *lattice mask* M is defined as $M = (N_M, E_M, E_A, N_{Me})$ where $N_M \subset \mathcal{N}$, $E_M \subset \mathcal{E}$, $E_A \subset \mathcal{N}^2$ 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 technics 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:

$$f : \mathcal{N} \rightarrow \{0, 1\}$$

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 $G = (\mathcal{N}, \mathcal{E})$, we note $N_F = N \in \mathcal{N} | f(N) = 0$ 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 will result 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:

- 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,
- any edge connected to a masked node will be masked,
- if a node other than $min(G)$ ends with no parent, it should be connected to all parent of its last former parent
- if a node other than $max(G)$ ends with no child, it should be connected to all child of its last former child
- if a node ends with a unique child and this child has a unique parent, these nodes should be merged,
- 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: analyse user's way of navigating throw the structure, explicitly ask for properties to be ignored, etc.

For example, a user navigating throw 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.

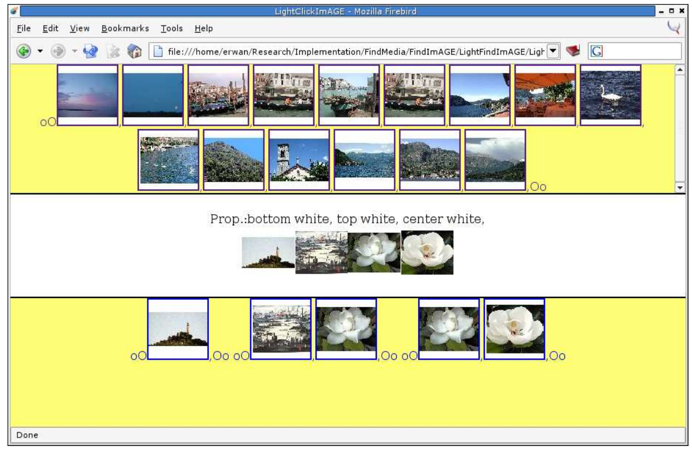

図 4 Current browsing graphical interface

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 [2] pages directly readable by a standard-compliant web browser and (2) the customization system, implemented as a browser extension operating client-side processing using ECMAScript [6]. The second part is still in development. Figure 4 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.

Evaluation

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.

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

- number of image per nodes,
- average number of children of a node
- and 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.

文　献

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

[2] The World Wide Web Consortium. Xhtml recommandation. http://www.w3.org/TR/xhtml1/.

[3] J. M. Corridoni, A. Del Bimbo, and E. Vicario. Image retrieval by color semantics with incomplete knowledge. *Journal of the American Society for Information Science*, 49(3):267–282, 1998.

[4] T. Gevers and A. Smeulders. A comparative study of several color models for color image invariant retreival, 1996.

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

[6] ECMA International. Standard ecma-262 (ecmascript language specification). http://www.ecma-international.org/... ...publications/standards/Ecma-262.htm.

[7] P. M. Kelly, T. M. Cannon, and D. R. Hush. Query by image example: The CANDID approach. In *SPIE Storage and Retrieval for Image and Video Databases III*, volume 2420, pages 238–248, 1995.

[8] J. Martinez and S. Guillaume. Colour image retrieval fitted to classical querying. *Networking and Information Systems Journal (NISJ)*, 1(2-3):251–278, 1998.

[9] J. Martinez and E. Loisant. Browsing image databases with Galois' lattices. In *Proceedings of the ACM International Symposium on Applied Computing (SAC'02)*, pages 971–975, Madrid, Spain, March 11-14 2002. ACM Computer Press.

[10] J. Martinez and S. Marchand. Towards intelligent retrieval in image databases. In B. Berra, editor, *Proceedings of the 5^{th} IEEE International Workshop on Multi-Media Data Base Management Systems (MMDBMS'98)*, pages 38–45, Dayton, Ohio, August 5-7 1998. IEEE Computer Press.

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

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

[13] C. Nastar, M. Mitschke, and C. Meilhac. Efficient query refinement for image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98)*, Santa Barbara, California, June 23-25 1998.

[14] S. Santini and R. Jain. Beyond query by example. In *Proceedings of the 6^{th} ACM International Multimedia Conference (ACM-MM'98)*, Bristol, UK, September 14-16 1998.

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