Text-based Image Indexing and Retrieval using Formal Concept Analysis

Imran Shafiq Ahmad, ***Non-Member***
School of Computer Science, University of Windsor
Windsor, ON N9B 3P4 – Canada
[e-mail: imran@cs.uwindsor.ca]

Received April 14, 2008; revised May 6, 2008; accepted May 10, 2008; published June 25, 2008

Abstract

In recent years, main focus of research on image retrieval techniques is on content-based image retrieval. Text-based image retrieval schemes, on the other hand, provide semantic support and efficient retrieval of matching images. In this paper, based on Formal Concept Analysis (FCA), we propose a new image indexing and retrieval technique. The proposed scheme uses keywords and textual annotations and provides semantic support with fast retrieval of images. Retrieval efficiency in this scheme is independent of the number of images in the database and depends only on the number of attributes. This scheme provides dynamic support for addition of new images in the database and can be adopted to find images with any number of matching attributes.

Keywords: Image indexing, image retrieval, formal concept analysis, text-based retrieval, feature-based retrieval

Digital Object Identifier 10.3837/tiis.2008.03.002

1. Introduction

With advancements in digital photography and digitization process, a large number of digital images are created or produced on a daily basis. The size of digital image repositories is growing at a very fast pace. Therefore, there is a great demand for efficient techniques to effectively manage and organize these images and a mechanism to navigate through such image collections. Users in many fields are exploiting the advantages offered by such digital collections in new and exciting ways. Examples of such applications include but are not limited to geographical and medical information systems, digital photo albums, sports and training, news, advertisement and multimedia applications. A great majority of users of such applications experience difficulties in locating a desired image in these large and varied collections. The problems of image retrieval are widely recognized and search for solutions is an active research area.

There are two main approaches to address the issues of image retrieval from an image database. A widely discussed approach is the feature or content-based image retrieval (CBIR) technique and a more traditional approach, called the text-based retrieval technique. In CBIR, primary emphasis is on identification and automatic extraction of computable visual features such as color, texture and shape [1]. Examples of such systems can be found in [2][3][4][5][6][7][8]. Smeulders et al. [9] provide a comprehensive survey of content-based retrieval and automatic feature extraction schemes. The main emphasis of these techniques is on finding a match based on visual similarity and that may not necessarily correspond to semantic similarity [10].

In traditional text-based approach, retrieval depends on existence of searchable textual descriptions of image content as keywords or annotations. There are processes to enter these descriptions into the database along with the actual image. This may require a priori knowledge of the application domain and the queries. In some systems, a content-based scheme is combined with a text-based approach. Examples of such systems include Virage [3], RetrievalWare [4], QBIC [5], MARS [6], VisualSEEK / WebSEEK [7], Netra [8] and Photobook [11].

Despite all of the research and development efforts, efficient and precise image retrieval is still an open research problem. In past, many keyword-based text information retrieval systems achieved great success for indexing image collections, especially on web sites and are still a common practice [12]. Kodak Picture Exchange System (KPX) [13], PressLink [14] and Time pictures archive collection (Time) [15] are examples of such systems. These approaches represent both general and specific information about the image content and the constituent objects. The concept of textual descriptions and keywords to represent images is not new. It existed even before the availability of digitization process. Librarians, curators and archivists provided access to images in libraries and archives with the help of manually assigned classification codes, annotations and keywords. Unfortunately, most of the details about any of the text-based systems described here is only theoretical in nature with no technical details and specifics to obtain a meaningful comparison.

Despite the subjectivity of textual descriptions and keywords, text-based retrieval systems tend to be more useful and practical than content-based image retrieval systems [10]. It is mainly because keywords and annotations are inherently semantic and can satisfy requirements of many users by computing similarities between annotations and image content [16]. Zhou and Huang [17] report on attempts to integrate advantages of both text-based and automatic feature-based systems. Examples of advantages and approaches linking text and images can be found in [16][18][19]. In [20], Wang et al. provide a comparison between ontology and keyword based image search. To overcome problems associated with manual annotation, [16][18][21][22][23] provide different automatic and semiautomatic approaches to annotate visual data.

Proposed by Rudolf Wille [24], *Formal Concept Analysis* (FCA) is a technique to represent a hierarchy of relationships among objects sharing common properties. FCA is based on mathematical order theory and formalization of philosophical understanding of a concept. It provides graph-based visualization of tabular data and has successfully been applied to a number of different fields. Some of the examples of its applications include information retrieval schemes [25][26][27], image and video mining [28], software re-engineering (e.g., reuse, class hierarchy, construction) [29], web document retrieval [30] and text data mining [31][32]. By using machine learning data sets, Carpinieto and Roman [33] have successfully demonstrated the use of lattices for class discovery and class protection.

FCA provides a mechanism to objectively identify and group objects that have common attributes and facilitates creation of a concept lattice. Most of the algorithms proposed in literature are for the generation of FCA from binary relationships [24][34] which are computationally expensive. Many applications require only a partial reconstruction or incremental update of the lattice structure. A literature survey indicates only a few algorithms to accommodate incremental updates in a FCA lattice [27][33][35]. The algorithm proposed by Valtchev and Missaoui [36] is an improvement and a formalization of the algorithm presented in [35]. Relatively little is known about the performance of all of these algorithms. In many cases, the proposed update processes result in restructuring of most if not all of the lattice structure.

In this paper, we propose use of FCA to both catalog the descriptions or keywords associated with the image content and to search and retrieve images on the basis of matching keywords. Typically, retrieval efficiency is a function of the number of images in a database. However, retrieval time complexity in the proposed scheme is independent of the number of images and depends only on the number of attributes associated with real objects in the image collection. This paper introduces a method to rebuild only part of the lattice to accommodate updates. This reduces the time to re-build the entire lattice structure. We have compared performance of the proposed system with a similar system based on concept lattice and rough set theory [37].

The remainder of this paper is organized as follows. In Section 2, theory of formal concept analysis is presented. Section 3 describes our proposed image retrieval scheme and the concept lattice update process. Section 4 contains experimental results and Section 5 presents conclusion and future research directions.

Table 1. Example object and attribute sets

o_id	Object Set	a_id	Attribute Set
1	Leech	a	needs water to live
2	Bream	b	lives in water
3	Frog	c	lives on land
4	Dog	d	needs chlorophyll to produce food
5	Spike-weed	e	two seed leaves
6	Reed	f	one seed leaf
7	Bean	g	can move around
8	Maize	h	has limbs
		i	suckles its offspring

2. Formal Concept Analysis (FCA)

The central idea of formal concept analysis is the understanding that a fundamental unit of thought is a concept and a context. A concept consists of an *extent* and an *intent*, defined as:

1. *Extent* that contains all objects with common attributes, and
2. *Intent* that contains all attributes common to all objects.

Definition 2.1 A *formal context* is a triple (O, A, R) consisting of two sets O and A and a relation R where O is a set of objects, A is a set of attributes and $R \subseteq O \times A$ is a binary relation between O and A. A formal context is expressed as oRa or $(o, a) \in R$ such that $o \in O$, $a \in A$ and is read as "the object o has the attribute a".

Definition 2.2 A *formal concept* of the formal context (O, A, R) is a pair (E, I) with the following conditions:

$$E \subseteq O, I \subseteq A, E' = I \text{ and } I' = E$$

where:

- E is a set of objects (*extent*) and I is a set of attributes (*intent*).
- $E' \subseteq A$ satisfy oRa for all $o \in E$ and $I' \subseteq O$ satisfy oRa for all $a \in I$.

Definition 2.3 The set of all concepts of ordered (O, A, R) by the relation $'\leq'$ is called a *concept lattice* of (O, A, R) or $B(O, A, R)$. The relation $'\leq'$ is called the *hierarchical order* or simple order of the concepts and is defined as:

$$(E_1, I_1) \leq (E_2, I_2) \quad \text{iff } E_1 \subseteq E_2 \quad \text{or} \quad \text{iff } I_2 \subseteq I_1 .$$

Table 2. Formal context presented as cross table

Object	a	b	c	d	e	f	g	h	i
Leech	X	X					X		
Bream	X	X					X	X	
Frog	X	X	X				X	X	
Dog	X		X				X	X	X
Spike-weed	X	X		X		X			
Reed	X	X	X	X		X			
Bean	X		X	X	X				
Maize	X		X	X		X			

Definition 2.4 If (E_1, I_1) and (E_2, I_2) are concepts of a context, (E_1, I_1) is called a *subconcept* of (E_2, I_2) provided that $E_1 \subseteq E_2$ (which is equivalent to $I_2 \subseteq I_1$). In this case, (E_2, I_2) is a *superconcept* of (E_1, I_1) and we write $(E_1, I_1) \leq (E_2, I_2)$.

2.1 A FCA Example

To describe FCA and to demonstrate its use, we provide an example from Ganter and Wille [34]. Let the member of an object set be as listed in column labeled as *Object Set* in **Table 1**. A list of attributes of these objects is given in the column labeled *Attribute Set*. To simplify the representation of members of both the object and the attribute sets for further discussions, each member of the object set can be recognized by the numeric label (1 – 8) on its left in column labeled *o_id*; each member of the attribute set can be recognized by the letters (a – i) on its left in the column labeled *a_id*.

Table 3. Formal concepts for the object and the attribute sets of **Table 1**

$Concept_1$	: ({1,2,3,4,5,6,7,8},{a})
$Concept_2$	: ({1,2,3,5,6},{a,b})
$Concept_3$	: ({3,4,6,7,8},{a,c})
$Concept_4$	: ({1,2,3,4,},{a,g})
$Concept_5$	: ({5,6,7,8},{a,d})
$Concept_6$	: ({1,2,3},{a,b,g})
$Concept_7$	: ({2,3,4},{a,g,h})
$Concept_8$	: ({5,6,8},{a,d,f})
$Concept_9$	: ({6,7,8},{a,c,d})
$Concept_{10}$	: ({2,3},{a,b,g,h})
$Concept_{11}$	: ({3,4},{a,c,g,h})
$Concept_{12}$	: ({3,6},{a,b,c})
$Concept_{13}$	: ({5,6},{a,b,d,f})
$Concept_{14}$	: ({6,8},{a,c,d,f})
$Concept_{15}$	: ({3},{a,b,c,g,h})
$Concept_{16}$	: ({4},{a,c,g,h,i})
$Concept_{17}$	: ({6},{a,b,c,d,f})
$Concept_{18}$	: ({7},{a,c,d,e})
$Concept_{19}$	: ({},{a,b,c,d,e,f,g,h,i})

From the given object and attribute sets we can derive the formal context, generally represented by a cross table (**Table 2**). The first row and the first column in this table represent all elements of the attribute set and the object set respectively. An "X" in this cross table implies that the particular object has all of the marked attributes. For example, leech has attributes *a*, *b*, and *g*, i.e., a leech needs water to live, lives in water, and can move around.

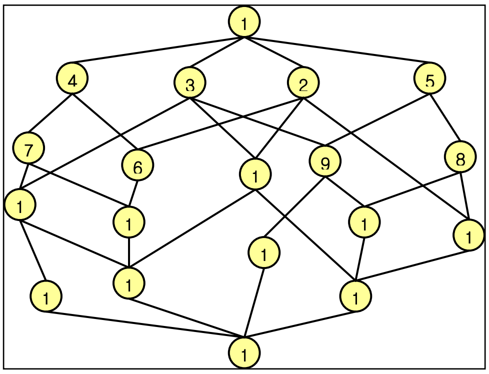


Fig. 1. Concept lattice

We can derive 19 formal concepts from the formal context (**Table 3**). The first set in each of these concepts consists of the members of the object set; the second set consists of the members of the attribute set. As an example, $Concept_{12}$ states that "*frog and reed need water to live, live in water and live on land*".

A concept lattice is a hierarchical structure in which all of the formal concepts are linked to each other by the definition of the "hierarchical order" relation. From the formal concepts given in **Table 3**, a concept lattice is obtained and is shown in **Fig. 1**. In this figure, a circle represents a concept and the number in the circle represents the concept number.

3. Image Retrieval Approach

To add images to the lattice, we propose use of the *bit set structure* and a new lattice update method in which this structure is used to represent an attribute set. As a result, on addition of a new image to the lattice, the proposed addition method rebuilds only a part of the lattice.

3.1 Attribute Set Structure

Let $O = \{o_1, o_2, ..., o_n\}$, $1 \le i \le n$ be a finite set of objects and $A_i = \{a_{i1}, a_{i2}, ..., a_{im}\}$, $1 \le j \le m$ represents the number of attributes of the i^{th} object. We propose use of the *bit set structure*, also known as the *bit vector* to represent attributes of an image in our scheme. A bit set structure is composed of a fixed number of bits, such that each bit represents only one member of the set. If a universal set U contains N items, an N-bit vector can represent any subset $S \in U$. Bit i is set to 1 if $i \in S$ and to 0 otherwise. The bit structure is initialized to 0 to indicate an empty set. This scheme uses only one bit per element and, is very space efficient. However, it requires prior knowledge of the application domain and size of the universe. Insertion or deletion of elements requires simply reversal of the appropriate bit and comparison can be made using bit-wise AND operation. Although explicit identification of individual elements in the bit set structure can take **O**(n) time, such identification may not be required in our approach. Formally, if a_q and a_i represent the query attribute set and the complete attribute set, then $a_i \cap a_q = a_q$ implies a match and, hence, retrieval of the concept.

Fig. 2. An example image and its attributes

As an example, suppose that the bit set structure is capable of representing 16 different attributes, marked a through p . Now suppose that we need to represent the attributes of the image in **Fig. 2** with attributes: building (b), car (g), hydrant (j), lawn (i) and tree (p). The bit set structure representation of this image is shown in **Fig. 3**. If the attributes we are looking for are $\{g, i, p\}$, a mere bit-wise AND of its bit set representation with that of the image and a comparison will establish retrieval validity.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	1

Fig. 3. Bit set structure representing attributes of image in **Fig. 2**

3.2 Lattice Structure

A survey of literature indicates that generally FCA has been applied to systems dealing with relatively small amount of data. The worst case time complexity for building a complete lattice structure is generally $\mathbf{O}(n^n)$ [34], prohibiting its use in applications with more data. Generally, to build a lattice, we gather all objects, extract all concepts and finally, link them together. All of the lattice must be rebuilt to add a new object. This requires expensive computational/mathematical operations and is a very time consuming process.

Lemma 3.1 *The total number of concepts in a lattice is at most $2^n - 2$ where n is the total number of attributes.*

Proof:

Let S_n be the number of concepts in a lattice (with the exception of the top and the bottom nodes). By induction:

If $n = 1$ then $S_1 = 0$

If $n = 2$ then $S_2 = {}_2C_1 = 2$

If $n = 3$ then $S_3 = {}_3C_1 + {}_3C_2 = 3 + 3 = 6$

For an arbitrary n, we need to find the number of concepts.

If $n = k$ then $S_k = {}_kC_{k-1} + {}_kC_{k-2} + ... + {}_kC_1 = \sum_{r=1}^{k-1} {}_kC_r$

From binomial expansion we know that $(a+b)^n = \sum_{r=0}^{n} {}_nC_r a^{n-r} b^r$. When $a = b = 1$ we get:

$$2^n = {}_nC_0 + {}_nC_1 + ... + {}_nC_{n-1} + {}_nC_n$$

where ${}_nC_1 + ... + {}_nC_{n-1} = \sum_{r=1}^{n-1} {}_nC_r \Rightarrow \sum_{r=1}^{n-1} {}_nC_r = 2^n - 2$.

Therefore, the total number of concepts $S_n = 2^n - 2$.

Corollary 3.2 *All of the Possible Concepts*[1] *are obtained from a maximum of $2^n - 2$ concepts.*

Possible concepts are obtained by comparing an attribute set with the attribute sets in the lattice. Since there are at most $2^n - 2$ concepts in the lattice, possible concepts are obtained by checking a total of $2^n - 2$ concepts.

3.3 Addition Method

In this paper, we introduce a new update method for rebuilding a concept lattice. This method is useful for an already established lattice and simply rebuilds only a part of it. To add a new concept (NC), we first find all possible concepts related to the new object followed by their super- and subconcepts. These possible concepts are linked to the super- and subconcepts by FCA's hierarchical order. If an attribute set of a possible concept is the same as that of a concept in the lattice, it is not added to the lattice. Therefore, the addition method may deal with only a subset of possible concepts. The addition method involves following steps:

Step 1: Find possible new concepts by comparing the attribute set of the new object with the attribute sets of the concepts in the lattice. The number of possible new concepts

[1] Possible Concept is a concept that, at times, could be one of the concepts in the lattice. If the lattice already has a concept with same attribute set as that of the possible concept, the possible concept is not added into the lattice and, hence, is called a possible concept.

obtained in this step determines the efficiency of the addition method, since the higher the number of possible new concepts, the higher the number of comparisons. During addition, two passes are made. In the first pass, all concepts in the lattice are examined and compared. In some cases, this result in redundant[2] or empty[3] attribute sets of possible new concepts. In the second pass, possible concepts with either redundant or empty attribute sets are removed. Formally, let a_{c_i} is the attribute set of the i^{th} concept in the lattice and a_j is the attribute set of a possible new concept, then:

- if $|a_j| = 0$ then a_j is an empty attribute set.
- if $|a_j| \neq 0$ then $a_j \equiv a_{c_i}$ and is a redundant attribute set where $1 \le j < i$ and $1 \le i \le 2^n - 2$.

A possible new concept whose attribute set is considered as redundant or empty is removed in this step.

Step 2: Starting from the root node, find superconcepts of those new possible concepts found in previous step using the depth-first and the top-down search methods. In order to be a superconcept for a possible concept, based on the order of the concept relation $(\le)$ as described in Definition 2.3, each concept in the lattice must satisfy following conditions:

a. Let C_1 be a concept in the lattice, a_1 be the attribute set of C_1, a_i be the attribute set of the subconcept of C_1 and a_n be the attribute set of the possible concept. If $a_1 \subseteq a_n$ and $\forall i$, $a_i \not\subset a_n$ then C_1 is a superconcept of the possible concept, as shown in **Fig. 4**.

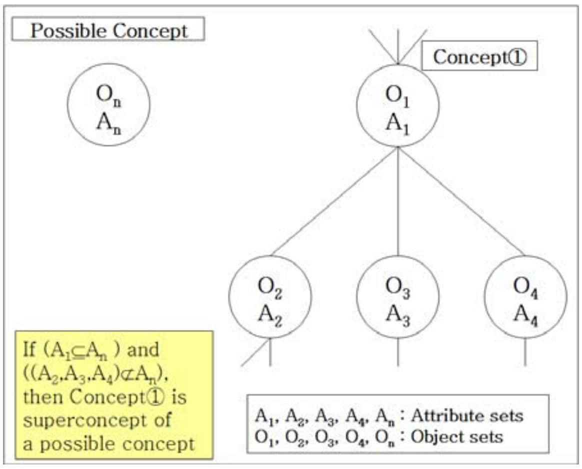


Fig. 4. Process of finding the superconcepts (Step 2)

b. If two attribute sets are identical and the number of elements of the object set in a possible concept is larger than the number of elements of the object set in a

2 Redundant Attribute Set is an attribute set whose elements are the same as that of a concept in the lattice.
3 Empty Attribute Set is a null attribute set and is obtained when the attribute set of the new object and those present in the lattice structure are the same.

concept in the lattice, simply replace the object set of the concept in the lattice with the object set of the possible concept.

Step 3: Find subconcepts of the possible concept based on depth-first and bottom-up search methods. We do not need to check the concepts to find subconcepts, since these have already been checked in previous step. This allows us to use a bottom-up search, instead of a top-down search with the condition:

- Let C_1 be a concept in the lattice, a_1 is the attribute set of C_1, a_i is the attribute set of the superconcept of C_1 and a_n is the attribute set of the possible concept. If $a_n \subseteq a_1$ and $\forall i,\ a_n \not\subseteq a_1$ then C_1 is a subconcept of the possible concept as shown in **Fig. 5**.

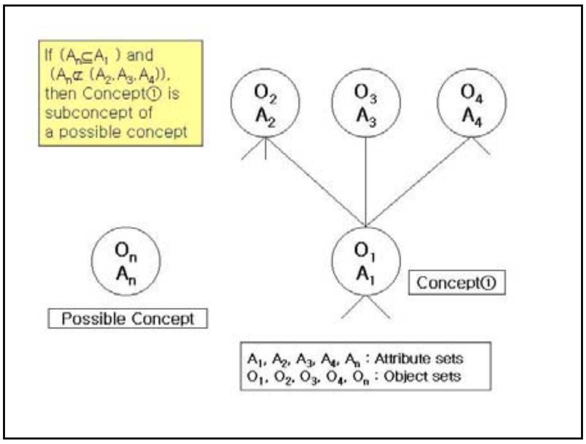


Fig. 5. Process of finding the subconcepts (Step 3)

Step 4: Link the possible concept to its super- and subconcepts found in previous steps. Find if the superconcept and the subconcept of a possible concept are already linked. If so, reconnect these as shown in **Fig. 6**. Repeat Steps 2 – 4 for all possible concepts created. Once all possible concepts have been added, we can add a new object as described in Step 5 and Step 6.

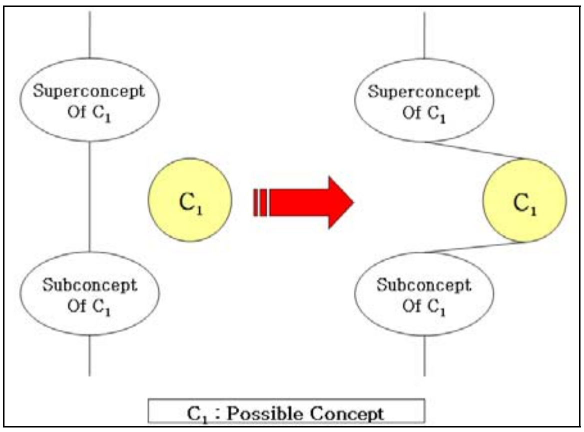


Fig. 6. Linking super- and subconcepts to a possible concept (Step 6)

Step 5: This step is similar to Step 2 and Step 3, but in this case we add a new object rather than a possible concept as shown in **Fig. 7**. We need to create a concept for the new object, before carrying out this step.

Step 6: In this step, we need to link the concept of the new object to the superconcepts and subconcepts found in Step 5. This is the same as Step 4.

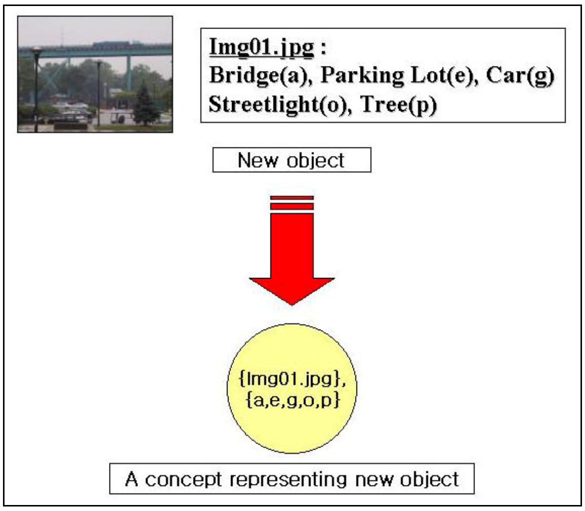


Fig. 7. Creation of a concept for a new object

Lemma 3.2 *The worst case time complexity of the proposed addition method is* $\mathbf{O}(2^n)$ *,where* n *is the total number of attributes.*

Proof:

From Lemma 3.1, the number of concepts in the lattice for the worst case is $2^n - 2$. According to Corollary 3.2, all possible concepts are obtained from these concepts. The addition method described in the previous section extracts these possible concepts. That is, the addition method checks $2^n - 2$ concepts, except for the top and the bottom nodes in the lattice structure, and extracts all possible concepts. Therefore, we need $2^n - 2$ comparisons i.e., the worst time complexity for the proposed addition method is $\mathrm{O}(2^n - 2) = \mathrm{O}(2^n)$ where n is the number of attributes.

3.4 Building the Concept Lattice - An Example

Initially, lattice is empty and contains only the top ([T]) and the bottom ([B]) nodes as shown in **Fig. 8-a**. Addition of first object to the lattice structure does not require Steps 1 – 4, as no concept is present in the lattice. During further processing, a concept for the new object as shown in **Fig. 7** is created which is then added to the lattice with [T] and [B] nodes as its super- and subconcepts respectively (**Fig. 8-b**).

Let us consider the case after addition of two more images (**Fig. 9-a**), resulting in a lattice with three concepts $C_1 - C_3$ (**Fig. 9-b**). Now assume that we need to add a new object (**Fig. 2**) to the lattice. From Step 1, the intersection of the attribute set of this object to those of the existing concepts already present in lattice results in three possible concepts $(PC_1 - PC_3)$ as shown in **Fig. 10**.

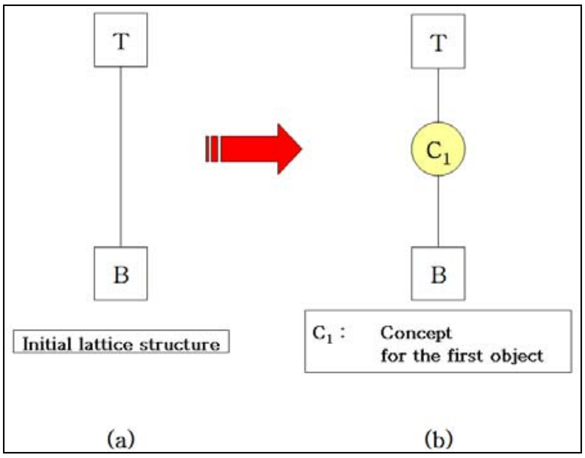


Fig. 8. (a) Initial lattice structure (b) after addition of the first object

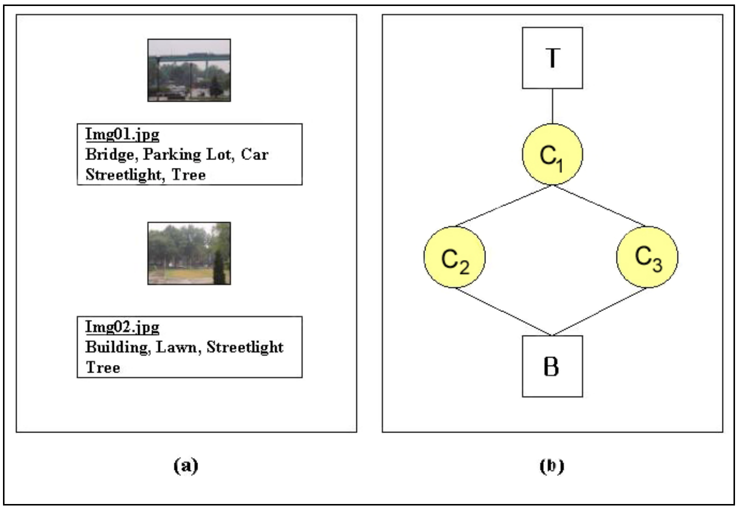


Fig. 9. (a) Two new images to be added and (b) resultant lattice structure

Now we need to find the super- and subconcepts of PC_1. In this example, [T] and C_1 are the super- and the subconcepts, respectively. They must be linked to PC_1 as shown in **Fig. 11-a**. For PC_2, PC_1 is a superconcept and C_2 is a subconcept as shown in **Fig. 11-b**. For PC_3, PC_1 is a superconcept and C_3 is a subconcept. The resulting lattice is shown in **Fig. 11-c**. So far, only the three possible concepts have been added to the lattice and we still need to add the object. This addition involves following steps:

- Find superconcepts. In this case, two superconcepts PC_2 and PC_3 are obtained.
- Find subconcepts. In this example, the bottom node [B] serves as the subconcept.
- Connect the nodes.

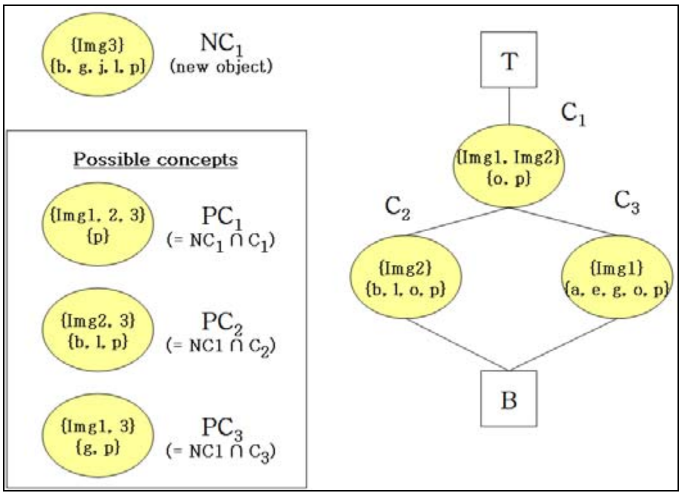


Fig. 10. Possible concepts after addition of a new object (**Fig. 2**) in an existing lattice

After addition of three objects, the lattice contains seven concepts as shown in **Fig. 12**.

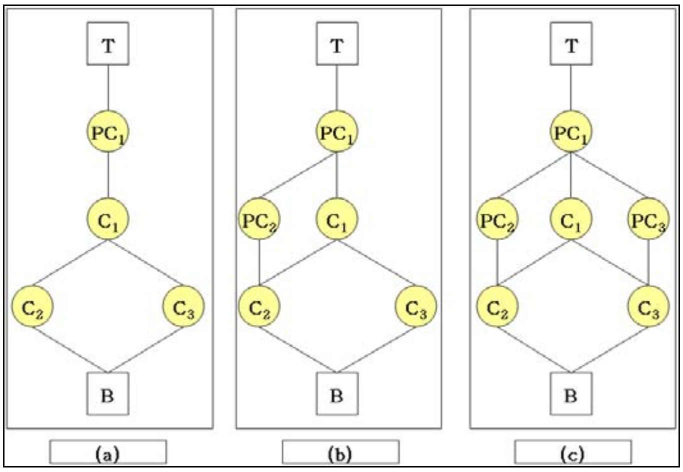


Fig. 11. Addition of three possible concepts

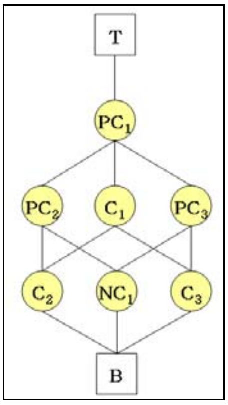


Fig. 12. Addition of a new object after addition of possible concepts

3.5 Image Retrieval

To retrieve similar images corresponding to a given query image, we simply need to find a concept whose attributes match with attributes a_q of the query image. This can provide us information about the object set and, in turn, about the corresponding set of images. The retrieval criterion is given as follows:

Let C_i be a concept with the attribute set a_i

C_{s_i} be a superconcept of C_i with the attribute set a_{s_i}

C_p be a subconcept of C_i

then

- if $a_i \cap a_q \neq a_q$ then move to C_p.
- if $a_i \cap a_q = a_q$ and $\exists i$ s.t. $a_{s_i} \cap a_q = a_q$ then move to C_{s_i} and keep searching.
- if $a_i \cap a_q = a_q$ and $\forall i$, $a_{s_i} \cap a_q \neq a_q$ then C_i is the concept satisfying the query.

Because of its hierarchical nature of FCA lattice, generally a top-down search is applied to find a concept since it allows extension of attributes while reducing the number of objects. Search for an image for retrieval in our case is a combination of depth-first and bottom-up search methods. Since all of the concepts and their attribute sets are not present (redundant and empty attribute sets have been removed), a simple a top-down search will fail to yield required information and. This may result in more comparisons than the bottom-up search since the hierarchical nature of the lattice implies that the superconcepts have less attributes then the subconcepts.

The time complexity of retrieval of images in this scheme is considered as the time complexity of finding a concept whose attribute set is equal to the attribute set of the user provided query image. The worst case time complexity for retrieval of images in this scheme is given in Lemma 3.3.

Lemma 3.3 [38] *The worst case time complexity for retrieving an image using FCA is* $\mathrm{O}((n-r)(r+1))$ *where* n *is the number of attributes of all of the images in the database and* r *is the number of attributes of the query image such that* $1 \leq r \leq n$.

Proof:

Assume that all of the concepts with same attribute set sizes are located at the same level. For an attribute set of size *n*, there will be *n* levels in the lattice. We can find that the number of concepts at level $(n-r) = {}_nC_r (= {}_nC_n - r)$ where $r = 1,2,...,n-1$. At level r, the size of attribute set of each concept is $(n-r)$. Therefore, in order to find a concept when the size of query attribute set is r, we need to consider concepts only from level 1 to level $(n-r)$.

Case 1: At level $l = 1$

At this level, the size of attribute set of each of the concept is $(n-1)$ and the number of concepts with r attributes is ${}_{n-r}C_{n-1-r} (= {}_{n-r}C_1 = n-r)$. These concepts are called the selectable concepts[4]. That is, the number of concepts that do not

[4] A concept is a selectable concept in the lattice if its attribute set includes all elements of the query attribute set. It implies that selectable concepts could be selected to check other concepts that are superconcepts of a selectable concept at the next level.

include r attributes is ${}_nC_1 - {}_{n-r}C_1 = r$. The worst-case time to find a concept with r attributes in its attribute set is ${}_nC_1 - {}_{n-r}C_1 + 1 = r+1$.

Case 2: At level $l = 2$

At this level, the size of attribute set of each of the concept is $(n-2)$ and the number of concepts including r attributes is ${}_{n-r}C_{n-2-r} = {}_{n-r}C_2$. All of the selectable concepts found at level $l = 2$ have $n-1-rC_1(= n-1-r)$ superconcepts and their attribute sets include the r attributes. That is, the number of concepts that do not include r attributes is $(n-1)-(n-1-r) = r$. Therefore, the worst case time of finding a concept, whose attribute set includes r attributes, is $r+1$.

Case 3: At level $l \geq 3$

By the same token, concepts with r attributes at level $n-r-1$ have $n-(n-r-1)-rC_{n-r-1} = {}_1C_{n-r-1} = 1$ superconcepts, such that their attribute sets include r attributes at level $n-r$. That is, the number of superconcepts whose attribute set does not include r attributes equals $(n-(n-r-1))-r = r$. Therefore, the worst case of finding a concept, whose attribute set includes the r attributes, is $r+1$. By induction, the number of worst case comparisons equals $(n-r)(r+1)$ except for the two comparisons for the intersection and the equality operations. Therefore, the worst case time complexity is $\mathbf{O}((n-r)(r+1))$.

3.5.1 Image Retrieval - An Example

Suppose $a_q = \{b, l, o, p\}$ represents the attribute set of a given query image and we need to find matching objects using the concept lattice. For clarity, **Fig. 13** shows only a small part of the complete lattice.

- While at [B], check its superconcepts:

  Concept C_7: $a_q \cap \{b, g, j, l, p\}_{C_7} \neq a_q$

  Go back to [B] and check its remaining superconcepts.

  Concept C_{16}: $a_q \cap \{a, b, e, g, i, p\}_{C_{16}} \neq a_q$

  Go back to [B] and check its remaining superconcepts.

  Concept C_{11}: $a_q \cap \{b, k, l, m, o, p\}_{C_{11}} = a_q$

  Go back to [B] and check its remaining superconcepts. In absence, move to the concept C_{11}.

- Check the superconcepts of concept C_{11}:

  Concept C_3: $a_q \cap \{b, l, o, p\}_{C_7} = a_q$

  Go back to C_{11} and check its remaining superconcepts. In absence, move to the concept C_3.

- At Concept C_3, check its superconcepts:

  Concept C_6: $a_q \cap \{b, l, p\}_{C_6} \neq a_q$

Go back to C_3 and check its remaining superconcepts.

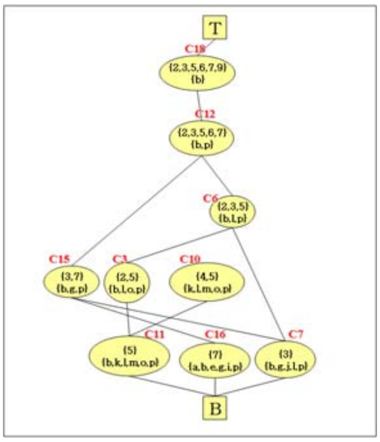


Fig. 13. A subsection of the complete lattice for an image retrieval example

- Since no superconcepts remain to be checked, and $a_q \subseteq C_3$, concept C_3 is the concept satisfying the criteria given in the query image. Therefore, by examining the object set of C_3, images *img02* and *img05* can be retrieved.

4. Experimental Results

We conducted a series of experiments on an image database to validate concepts presented in this paper. Our image database consists of more than 13,000 well annotated nature/architectural images. Results are collected through a Java-based application on a PC, equipped with a 2.4 GHz Intel Pentium processor and 2 GB of RAM in Windows XP environment. To simplify data collection process, the attribute set in these experiments are divided into five categories with a total of 45 distinct attributes as members of the attribute set.

Table 4 shows the number of concepts generated as a function of number of images present in the concept lattice for the four different orders of insertion. As one can observe from this table, different insertion orders result in small variations in the number of generated concepts. Though these variations for this particular data set do not appear to be drastic, they can affect the overall image insertion time that may become quite noticeable for a larger image data sets. During retrieval phase, the attribute set of the query image is compared against the concepts already in the concept lattice. Therefore, depth of concept lattice can contribute significantly effect retrieval efficiency. The greater the concept lattice depths, the more involved are the comparisons and the retrieval time is higher. We observed lattice depth at various points during the image insertion process as listed in **Table 4**. This depth is a function of the number of attributes, as well as a particular set of images, and may differ for a different set of attributes or domain.

Fig. 14 shows the average cumulative image insertion time as a function of number of images already present in the concept lattice. As one can observe from **Fig. 14**, the cumulative image insertion time exponentially increases with the number of images. This is due to generation of more and more redundant and/or possible concepts that need to be eliminated or absorbed in the already existing set of concepts, requiring additional comparisons. This is the time to build the entire lattice structure from the beginning. Since images are added to it only once, it can be an off-line process and will not affect the system retrieval performance.

Table 4. Number of concepts generated and the cumulative time as a function of number of images inserted in the concept lattice in four different orders

Number	Lattice	Number of Concepts for order			
		R_1	R_2	R_3	R_4
1000	4	1151	1450	1413	1532
2000	5	1824	2218	2582	2531
3000	5	3495	2809	3017	4107
4000	5	4195	4396	4397	5434
5000	6	4803	5688	5354	6890
6000	6	5234	6872	7743	7976
7000	6	7666	7283	9359	8469
8000	7	9607	8977	10670	8882
9000	7	11360	11100	11273	9394
10000	7	12964	12621	11705	12756
11000	8	14966	14969	12695	14533
12000	8	16651	16853	16386	16502
13000	9	18959	18959	18959	18959

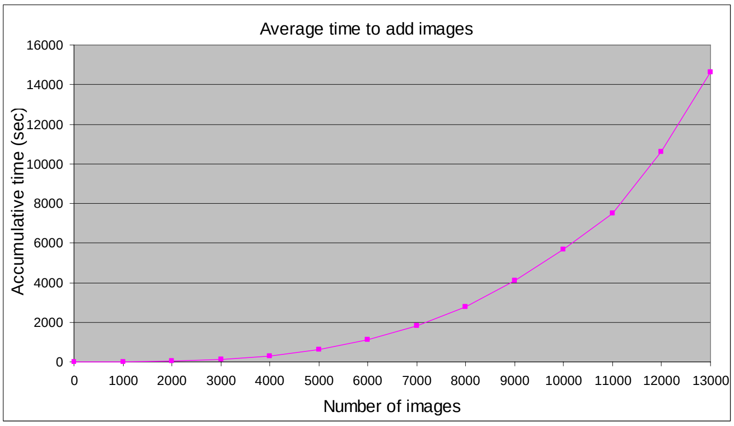


Fig. 14. Average time to add images to the concept lattice

Generally, with an increase in the number of concepts in lattice, the insertion time for addition of new images in the lattice structure also increases due to an increase in the number of possible and/or redundant concepts. However, at occasions, less time may be required to add new images, even though more concepts are present in the lattice. This reduction in time is primarily due to reduced number of possible concepts generated during the insertion process. Therefore, the two prime factors contributing to the addition of new images in the concept lattice are (a) the number of concepts in an existing lattice and (b) the number of newly generated possible concepts.

Fig. 15 shows the average number of concepts generated as a function of number of images. As one may expect, the number of concepts generated increases with the number of images. With our sample image database and 2^{45} worst case total number of concepts, the increase in number of concepts is nearly linear. However, a different order of insertion of images or a different domain may generate a totally different set of concepts.

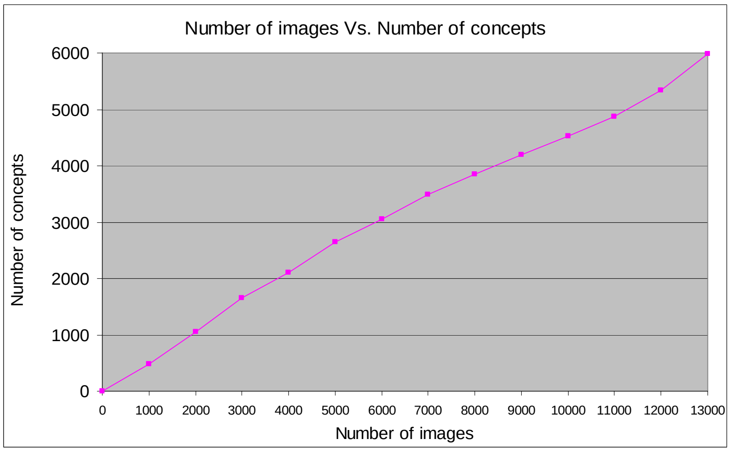


Fig. 15. Number of images vs. number of concepts

We also collected information about the time to build the lattice as a function of number of attributes. For this experiment, we arbitrarily selected 1000 images with different attributes for insertion into the lattice structure. The average results are shown in **Fig. 16**. The addition method and the attributes effect the total insertion time, but this may be an off-line process and retrieval performance is independent of such time.

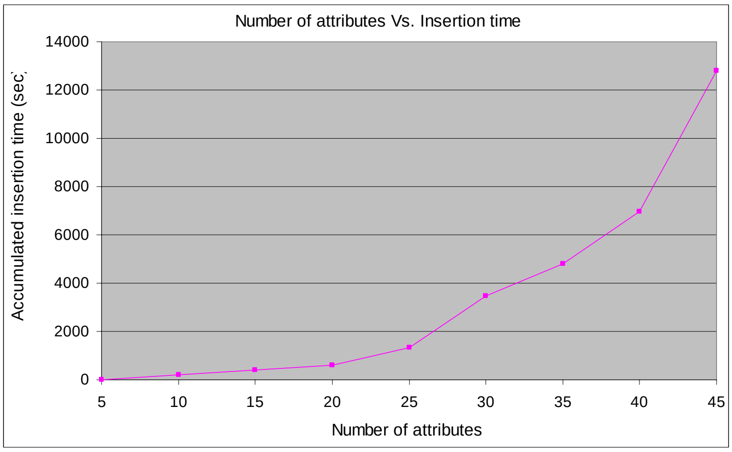


Fig. 16. Number of attributes vs. insertion time

The retrieval effectiveness of image retrieval systems is generally defined in terms of *precision* and *recall* [39]. Recall is defined as the ratio of the number of relevant images retrieved to that of the total number of relevant images in the system. Conversely, precision is the ratio of the number of relevant images retrieved to that of the number of all the images retrieved. We have compared the performance of our system with a similar system proposed by Li & Wang [37] for image retrieval. Their system is primarily based on concept lattice and rough set theory. Their system does not provide an update mechanism and the results presented in [37] are for only 500 images. Precision and recall graphs for the proposed system and that of the system presented in [37] for the same image data set are given in **Fig. 17**. As one can observe, precision of the proposed system is always higher that that of the system in

[37]. This is due to the fact that comparisons in our scheme involve matching keywords with that of the user specified query. However, it must be noted that the retrieval effectiveness is highly dependent on the accuracy of the image annotation process.

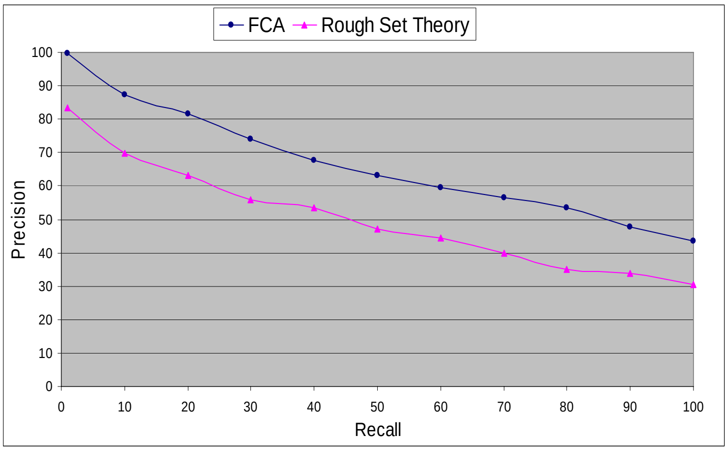


Fig. 17. Precision-Recall graph for images in the system

5. Conclusions and Future Directions

Generally, the time complexity of an image retrieval system is a function of number of images in the database. In the scheme presented here, it is independent of the number of images and depends only on the number of attributes. Further, the most common approach to accommodate any change in the number of objects for a given lattice is to rebuild the entire lattice. However, this is an expensive and time consuming process. To overcome this problem, this paper introduces a new approach for addition of new objects which requires only a part of the lattice to be rebuilt, thus resulting in a significantly reducing the lattice rebuilding cost but requires a prior knowledge of the application domain. This method can be improved further by eliminating its domain dependency and limiting the redundancy of possible concepts. Further extensions are possible by developing methods for addition and deletion of attributes as well as deletion of objects.

Acknowledgement

The Research was partially supported by the Natural Sciences and Engineering Council (NSERC) of Canada. I deeply appreciate the unknown reviewers' invaluable comments and very quick responses during the review time. And also I would like to express my great thanks to Prof. Chang-Gyoon Lim and Prof. Minho Jo who have invited me to this Transactions. Their comments and efforts definitely have enhanced quality of this paper.

References

[1] Yong Rui, Thomas S. Huang, and Shih-Fu Chang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues," *Journal of Visual Communication and Image Representation*, 10(1):39–62, Mar. 1999.

[2] Imran Ahmad and William I. Grosky, "Indexing and Retrieval of Images using Spatial

Constraints," *Journal of Visual Computing and Image Representation*, 14(3):291–320, 2003.

[3] Jeffrey R. Bach, Charles Fuller, Amarnath Gupta, Arun Hampapur, Bradley Horowitz, Rich Humphrey, Ramesh Jain, and Chiao-Fe Shu, "The Virage image search engine: An open framework for image management," In *IS&T SPIE Proceedings:Storage and Retrieval for Image and Video Databases IV*, volume 2670, pp. 76–87, San Jose, California, Feb. 1996.

[4] Roberto Brunelli and Ornella Mich, "Image Retrieval by Example," *IEEE Transactions on Multimedia*, 3(2):164–171, 2000.

[5] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System," *IEEE Computer*, 28(9):23–32, Sep. 1995.

[6] Michael Ortega, Yong Rui, Kaushik Chakrabarti, Sharad Mehrotra, and Thomas Huang, "Supporting Similarity Queries in MARS," In *Proceedings of 5th ACM International Conference on Multimedia*, pp. 403–413, Nov. 1997.

[7] John R. Smith and Shih-Fu Chang, "VisualSEEk: A Fully Automated Content-Based Image Query System," In *Proceedings of 4th ACM International Conference on Multimedia*, pp. 211–218, 1996.

[8] B.S. Manjunath Wei-Ying Ma, "NeTra: A Toolbox for Navigating Large Image Database," *Multimedia Systems*, 7(3):184–198, 1999.

[9] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, and Ramesh Jain Amarnath Gupta, "Content-Based Image Retrieval at the End of the Early Years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[10] Ye Lu, Chunhui Hu, Xingquan Zhu, HongJiang Zhang, and Qiang Yang, "A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval Systems," In *Proceedings of 9th ACM International Conference on Multimedia*, pp. 31–38, 2001.

[11] Alex Pentland, Rosalind W. Picard, and Stanley Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," In *SPIE Storage and Retrieval for Image and Video Databases II*, volume 2185, pp. 34–47, 1994.

[12] Abby Goodrum, "Image Information Retrieval: An Overview of Current Research," *Information Science*, 3(2):63–67, 2000.

[13] J. Larish, "Kodak's Still Picture Exchange for Print and Film Use," *Advanced Imaging*, 10(4):38–39, 1995.

[14] M. Martucci, "Digital Still Marketing at PressLink," *Advanced Imaging*, 10(4):34–36, 1995.

[15] L. Bielski, "20 Million Photos will be digitized at Time Warner: The Image Database of the Future begins," *Advanced Imaging*, 4(10):26–28 and 91, 1995.

[16] Florent Monay and Daniel Gatica-Perez, "On Image Auto-Annotation with Latent Space Models," In *Proceedings of 11th ACM Multimedia*, pp. 275–282, Berekley, California, Nov. 2003.

[17] Xiang Sean Zhou and Thomas S. Huang, "Unifying Keywords and Visual Contents in Image Retrieval," *IEEE Multimedia*, 9(2):23–33, Apr. 2002.

[18] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan, "Matching Words and Pictures," *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[19] Marjo Markkula and Eero Sormunen. End-User Searching Challenges Indexing Practices in the Digital Newspaper Photo Achive," *Information Retrieval*, 1(4):259–285, 2000.

[20] Huang Wang, Song Liu, and Liang-Tien Chia, "Does Ontology and Multi-Modality Ontology Approaches," In *14th ACM International Conference on Multimedia (MM'06)*, pp. 109-112, Santa Barbara, California, Oct. 23-27, 2006.

[21] David M. Blei and Michael Jordan, "Modeling Annotated Data," In *Proceedings of 26th ACM SIGIR Conference*, pp. 127–134, Toronto, Jul. 2003.

[22] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic Image Annotation and Retrieval using Cross-Media Relevance Models," In *Proceedings of 26th ACM SIGIR Conference*, pp. 119–126, Toronto, Jul. 2003.

[23] Jia Li and James Z. Wang, "Automatic Linguistic Indexing of Pictures by a Statistical Modelling Approach," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25(9):1075–1088, 2003.

[24] Rudolf Wille, "Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts," In Ivan Rival, editor, *Ordered Sets*, pp. 445–470, 1982.

[25] Richard J. Cole and Peter W. Eklund, "Application of Formal Concept Analysis to Information Retrieval Using a Hierarchically Structured Thesauris," In *Proceedings of the 5th International Conference on Conceptual Structures (ICCS'96)*, pp. 1–12, 1996.

[26] Jon Ducrou and Peter Eklund, "Browsing and searching MPEG-7 images using formal concept analysis," In *AIA'06: Proceedings of the 24th IASTED International conference on Artificial Intelligence and Applications*, pp. 317–322, Anaheim, CA, USA, 2006.

[27] Robert Godin, Rokia Missaoui, and Alain April, "Experimental Comparison of Navigation in a Galois Lattice with Conventional Information Retrieval Methods," *International Journal of Man-Machines Studies*, 38(5):747–767, 1993.

[28] Rokia Missaoui and Roman M. Palenichka, "Effective Image and Video Mining: An Overview of Model-based Approaches," In *MDM'05: Proceedings of the 6th International Workshop on Multimedia Data Mining*, pp. 43–52, Chicago, Illinois, 2005.

[29] Robert Godin G. Mineau, Rokia Missaoui, M. St-Germain, and N. Faraj, "Applying Concept Formation Methods to Software Reuse," *International Journal of Software Engineering*, 5(1):119–142, 1994.

[30] W. C. Cho and D. Richards, "Ontology Construction and Concept Reuse with Formal Concept Analysis for Improved Web Document Retrieval," *Journal of Web Intelligence and Agent Systems*, 5(1):109–126, 2007.

[31] Young Park, "Software Retrieval by Samples Using Concept Analysis," *Journal of Systems and Software*, 3(54):179–183, 2000.

[32] Michael Siff and Thomas Reps, "Identifying Modules Via Concept Analysis," In *Proceedings of theInternational Conference on Software Maintenance*, pp. 170–179, Bari, Italy, Oct. 1997.

[33] Claudio Carpineto and Giovanni Romano, "An Order-Theoretic Approach to Conceptual Clustering," In *Proceedings of the 10th Machine Learning Conference*, pp. 33–40, Amherst, MA, 1993.

[34] Bernhard Ganter and Rudolf Wille, "Formal Concept Analysis: Mathematical Foundations," *Springer-Verlag*, 1999.

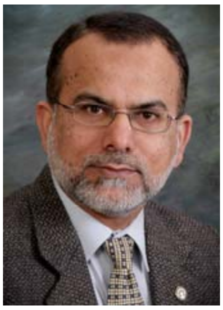
[35] Robert Godin, Rokia Missaoui, and Hassan Alaoui, "Incremental Concept Formation Algorithms based on Galois (Concept) ," *Computational Intelligence*, 11(2):246–267, 1995.

[36] Petko Valtchev and Rokia Missaoui, "Building Galois (Concept) Lattices From Parts: Generalizing the Incremental Approach," In *Proceedings of the 9th International Conference on Conceptual Structures (ICCS'2001)*, pp. 290–303, Stanford, CA, 2001.

[37] Ming Li and Tong Wong, "An Approach to Image Retrieval Based on Concept Lattices and Rough Set Theory," In *6th IEEE International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2005)*, pp. 845–849, Dec. 05-08, 2005.

[38] Imran Ahmad and Taek-Sueng Jang, "Old Fashion Text-based Image Retrieval using FCA," In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2003)*, volume III, pp. 33–36, Spain, Sep. 2003.

[39] Cornelis J. van Rijsbergen, "Retrieval Effectiveness," In Karen Sparck Jones, editor, *Information Retrieval Experiments*, pp. 32–43. Butterworths, 1981.



Imran Shafiq Ahmad is Associate Professor of Computer Science at the University of Windsor – Canada. He earned M.Sc. in Applied Physics (Electronics) from the University of Karachi – Pakistan, M.S. and Ph.D. in Computer Science from Central Michigan University – USA and Wayne State University – USA respectively. Before joining University of Windsor, he was Assistant Professor at the Western Connecticut State University. His research interests lie in image and video retrieval and multimedia systems. He has authored numerous scientific articles, has served on technical committees for a number of international conferences and has reviewed articles for many journals, conferences and workshops.