

An Intelligent Web Image Retrieval System

Sungyong Hong, Chungwoo Lee, and Yunmook Nah

Department of Computer Engineering, Dankook University

Seoul 140-714, Korea

{syhong, cman, ymnah}@dankook.ac.kr

ABSTRACT

Recently, the web sites such as e-business sites and shopping mall sites deal with lots of image information. To find a specific image from these image sources, we usually use web search engines or image database engines which rely on keyword only retrievals or color based retrievals with limited search capabilities. This paper presents an intelligent web image retrieval system. We propose the system architecture, the texture and color based image classification and indexing techniques, and representation schemes of user usage patterns. The query can be given by providing keywords, by selecting one or more sample texture patterns, by assigning color values within positional color blocks, or by combining some or all of these factors. The system keeps track of user's preferences by generating user query logs and automatically add more search information to subsequent user queries. To show the usefulness of the proposed system, some experimental results showing recall and precision are also explained.

Keywords: image databases, texture, non-texture, color, bit vector index, content-based retrieval, data mining

1. INTRODUCTION

With the rapid development of internet technology, the number of internet users and the amount of multimedia information on the internet is ever increasing. Recently, the web sites such as e-business sites and shopping mall sites deal with lots of image information. To find a specific image from these image sources, we usually use image database engines or web search engines. But, the feature based retrieval capabilities of these systems are quite limited, especially for the web images.

Many image retrieval systems have been developed, such as QBIC, Safe, VisualSEEK, Photobook, WBIIS, SIMPLIcity, Blobworld[2,3,4,5,6]. Some systems rely on keyword only retrievals and others support image content based retrievals. In the latter approach, they support image retrievals based on the image feature information, such as average colors, color histograms[8], texture patterns[2,7], and shape objects[2]. But, most of them are developed for image database applications.

This paper is an effort to develop an intelligent web image retrieval system. The scope of this paper is a little bit different. We are currently developing the agent-based image search engine which supports the content-based retrieval on web images. We are trying to support various access paths on web images with customized feedback according to internet user's preferences. To support this, we have applied data mining[9,10,11,12] and web mining techniques[13].

This system supports hybrid image retrievals based on query keywords, colors, and textures. For a given web image, our system classifies it whether it is textured or non-textured and, for the textured image, assigns the appropriate texture pattern(s) to the image. The proposed system also generates positional color information. The query can be given by providing keywords, by selecting one or more sample texture patterns, by assigning colors within positional color blocks, or by combining some or all of these factors.

The proposed system is intelligent since it remembers user's preferences and adds more feature information to the given query. For example, if the user provides search keyword 'shirts', the system automatically adds texture patterns(e.g., check pattern) and color information according to user's previous preferences. For this purpose, the system keeps track of user's query logs and applies data mining techniques to determine user usage patterns on colors and textures.

In the next section, the system architecture is introduced. In section 3, the indexing techniques on textures and colors are proposed. In section 4, we explain various image retrieval methods. Section 5 describes how user logs can be used to determine user's preferences. Section 6 briefly shows the experimental results in terms of recall and precision. A concluding remarks can be found in section 7.

2. SYSTEM ARCHITECTURE

The architecture of the agent-based image search engine, IWIR(Intelligent Web Image Retrieval), which supports the content-based retrieval on web images is shown in Figure 1. Web images are collected by the Intelligent Agents. The Feature Extractor preprocesses web images to extract metadata, color and texture information. Here, image metadata represents image size, image type, etc.

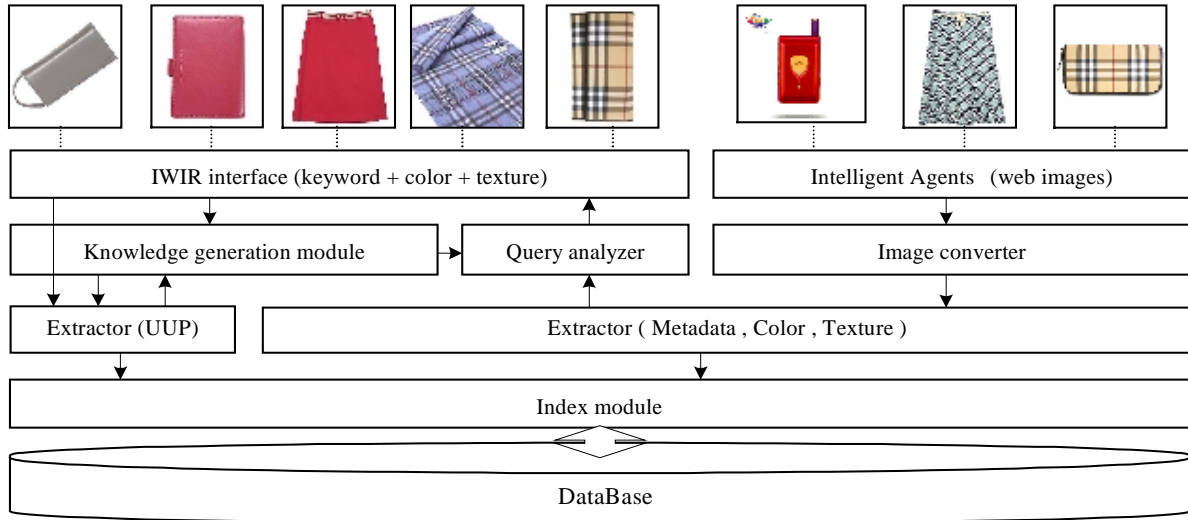


Figure 1. System architecture of the intelligent web image retrieval system IWIR

The Index Module generates texture bitmap indexes and color region indexes for web images. The IWIR Interface provides various search forms for content based retrievals. The image search logs are collected and used by the Knowledge Generation Module to build user usage pattern(UUP)s. The Query Analyzer utilizes user usage patterns as well as image feature information to return more accurate search results, thus increasing user's satisfaction.

3. TEXTURE AND COLOR BASED INDEXING

3.1 Indexing on image textures

Users of e-business shopping mall sites are usually interested in the texture, color, shape, and size of products. These web catalog images can be classified as textured(Figure 2a) and non-textured(Figure 2b).



Figure 2. Image types

We represent set of stored images $I = \{I_1, I_2, \dots, I_n\}$ and set of texture patterns as $T = \{T_1, T_2, \dots, T_t\}$. An image can have zero or more texture patterns, so that we represent the set of textures of the i -th image as $\text{texture}(I_i) = \{T_1, T_2, \dots, T_k\}$ or $\text{texture}(I_i) = \{\}$, where k is the number of textures included in the i -th image. The empty set means that the given image is non-textured. We represent the texture information of an image as t -bit bitmap vector. The bit 1 of the bitmap vector for an image represents the presence of the texture within it. Figure 3 shows an example of bit vector index for n images. In Figure 3, the image I_4 , having all zero bits, represents a non-textured image. The first and last row represent $\text{texture}(I_1) = \{T_1, T_t\}$ and $\text{texture}(I_n) = \{T_2, T_t\}$. This means that I_1 and I_n have one common texture T_t . Therefore, these two images will be seen to users as similar images. Similarly, we can observe that I_2 and I_n have one common texture T_2 . These bit vector indexes are used to support fast texture based retrievals.

	T_1	T_2	T_3	T_4	T_5	T_6	T_t	<i>classification</i>
I_1	1	0	0	0	0	0	1	<i>texture image</i>
I_2	0	1	0	0	0	0	0	<i>texture image</i>
I_3	0	0	1	0	0	0	0	<i>texture image</i>
I_4	0	0	0	0	0	0	0	<i>non-texture image</i>
I_n	0	1	0	0	0	0	1	<i>texture image</i>

Figure 3. Bit vector index for textured images

3.2 Indexing on image colors

Web images have various colors. The average RGB/HSI values or their color histograms are usually used for content based retrievals on colors. The histogram based retrieval is quite slow because of the computing overhead by dimensionality curses. To lessen the computing time, it is essential to lower the histogram dimensions. Figure 4 shows an example lowering the image dimension from 256×256 to 8×8 .

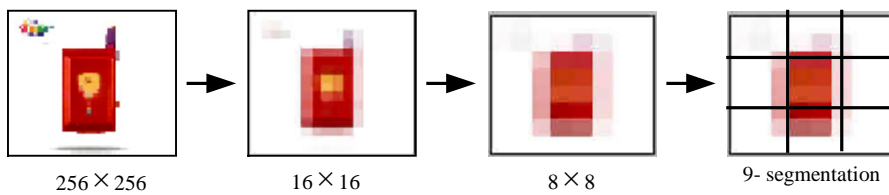


Figure 4. Transforming the image into the lower dimensions

We divide the resulting low resolution image into $m \times m$ positional blocks (segments) to provide queries on color regions. For each C_j ($1 \leq j \leq m \times m$), the average RGB value and average HSI value of the corresponding color regions are stored. Figure 5 shows the set of colors of the i -th image, $\text{color}(I_i) = \{C_1, C_2, \dots, C_9\}$, where $m = 3$. The color search ranges are expressed as (x-range, y-range). For example, to find images having color search conditions on region $C_4, C_5,$ and C_6 , the color search range is expressed as (2, 1:3).

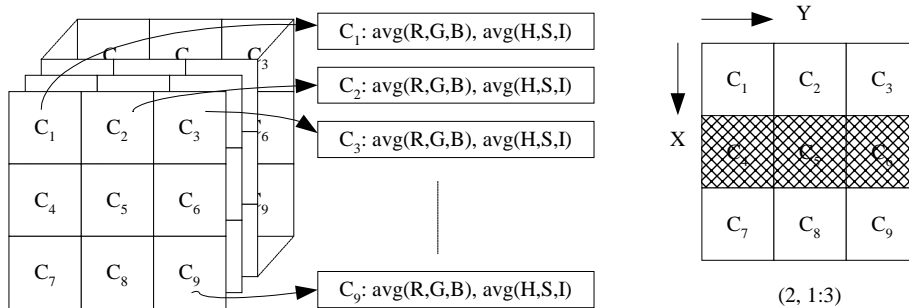


Figure 5. Color information on positional image blocks

For an image, its image identifier, the average RGB value, the average HSI value, and average RGB/HSI values of each color regions are stored in the database (Figure 6). The search conditions can be specified using the appropriate color positions. The system searches the matching images by comparing the average color of the whole image and the average colors of the specified color regions simultaneously.

4. INTEGRATED WEB IMAGE RETRIEVAL

The IWIR supports various search interfaces on web images. It is possible to present queries by texture, color, keyword, or by combining some or all of these factors. The search patterns of individual users are recorded in the system log table.

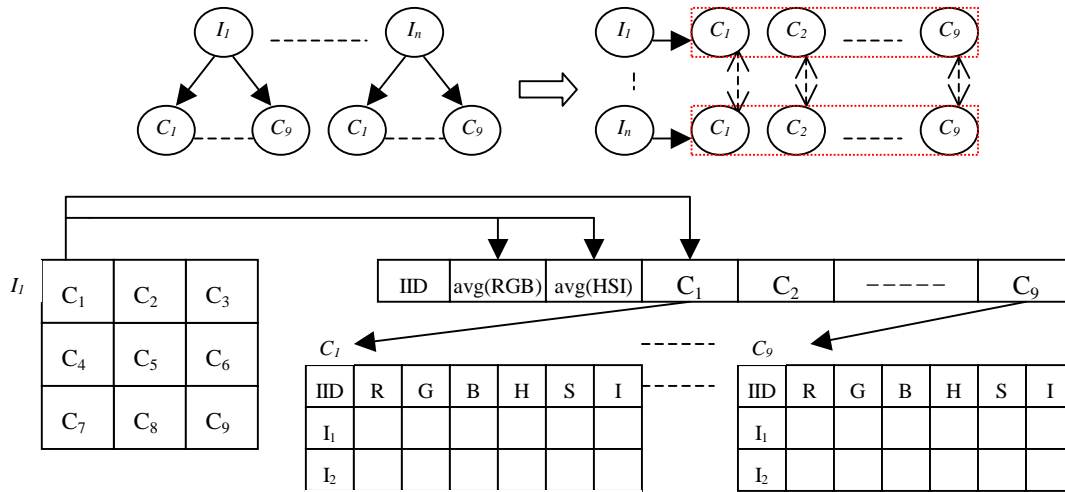


Figure 6. Storage structure of color information

4.1 Searching by texture

Users can select zero or more texture patterns from the query by texture interface. The user id and his selected texture values are recorded in the log table for UUP processing (see section 5). We apply the exclusive-OR operation on the texture bit vectors of query image and each stored image. In this operation, we exclude the case where the values of both bit positions are 0. Then, we complement its result, and apply bitwise OR operations on the complemented result. The final value 1 means two images are matched. Figure 7 summarizes this procedure. To retrieve the matched images, the appropriate SQL query is generated and its results are returned to users. Figure 8 shows an example of query by texture.

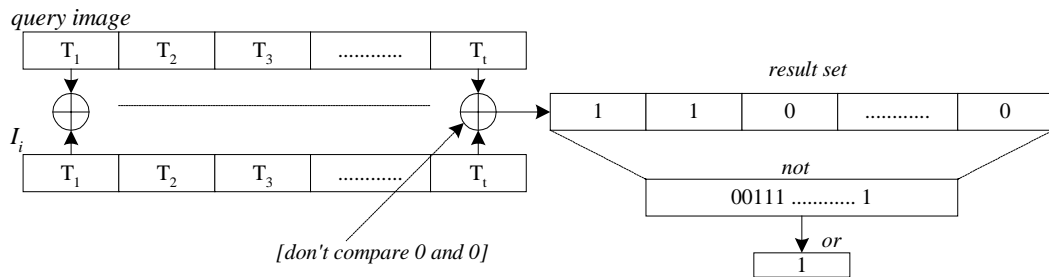


Figure 7. Matching by texture information

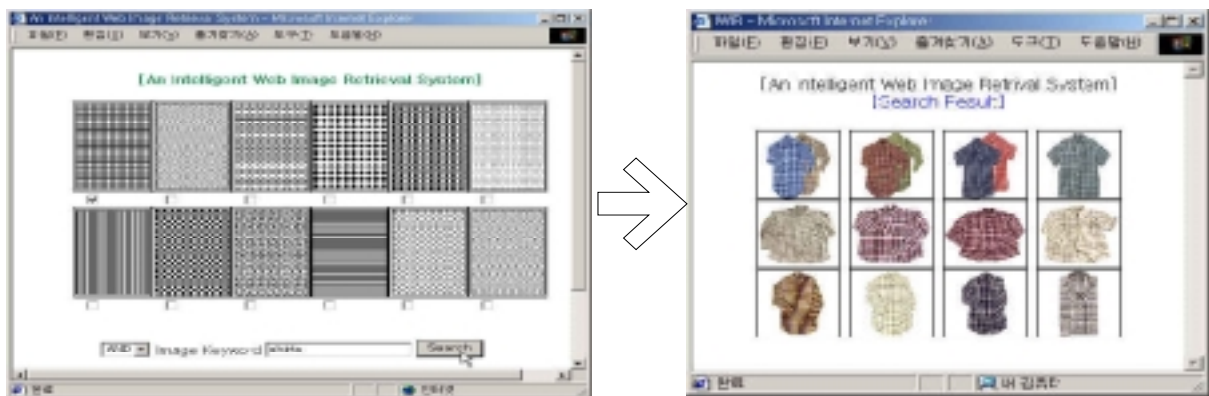


Figure 8. An example of query by texture

4.2 Searching by color

Users can specify RGB or HSI color values on the interested color regions through the query by color interface. Figure 9 shows an example of query by color. Alternatively, the color query can be given by uploading a sample image. The user id and his selected color values are recorded in the log table. The average values of the whole image or each color regions are extracted and the appropriate SQL query is generated. The retrieval procedure for query by color is as follows (here, $m=3$).

```

procedure Query_by_color (var r_color:avg(R);g_color:avg(G);b_color:avg(B);
                           h_color:avg(H);s_color:avg(S);i_color:avg(I) );
  var rowcount : integer; j : integer;
  database connection.open;
begin
  for j :=1 to 9 do
    if (Cj =Null) then set R,G,B and H,S,I values as null;
    else // set of color range
      open.recordset
      Rcolor_p := Cint (rs("r_color")) + 15; Rcolor_n := Cint (rs("r_color")) - 15; //tolerance ±15%
      Gcolor_p := Cint (rs("g_color")) + 15; Gcolor_n := Cint (rs("g_color")) - 15;
      Bcolor_p := Cint (rs("b_color")) + 15; Bcolor_n := Cint (rs("b_color")) - 15;
      Hcolor_p := Cint (rs("h_color")) + 15; Hcolor_n := Cint (rs("h_color")) - 15;
      Scolor_p := Cint (rs("s_color")) + 15; Scolor_n := Cint (rs("s_color")) - 15;
      Icolor_p := Cint (rs("i_color")) + 15; Icolor_n := Cint (rs("i_color")) - 15;
      close.recordset
    endif

    retrieve ID from color index;
    if (rs.EOF := true) then query_result := Null;
    else do while not rs.EOF
      rowcount := rowcount + 1;
      rs.movenext
    loop
  endif
  database connection.close // database disconnection
  set rs.nothing
end;
end Query_by_color
  
```

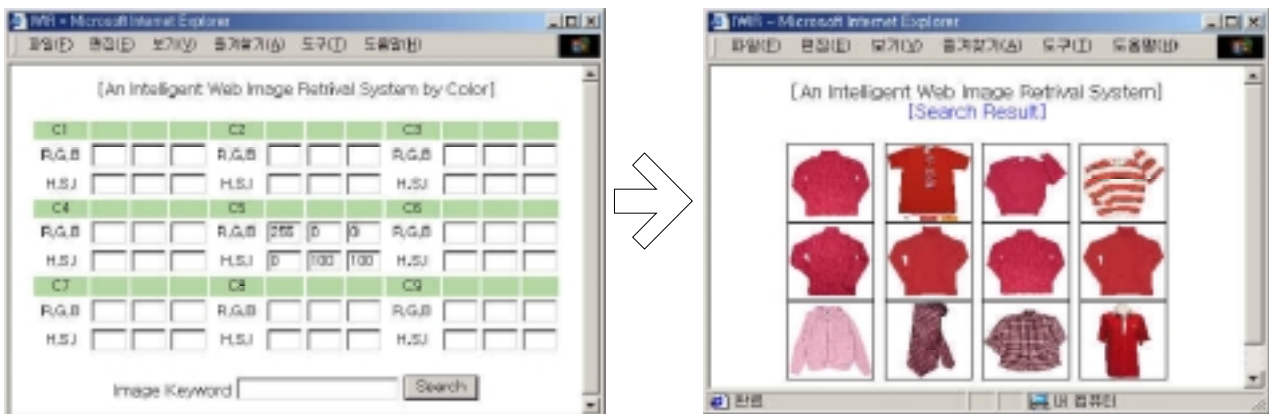


Figure 9. An example of query by color.

4.3 Integrated web image searching

Users can specify queries having search conditions related with keyword, texture, and color. Figure 10 shows the query processing procedure. The images are matched using texture and color indexes and the appropriate SQL query is generated using texture ids, color ids, and keywords. Figure 11 shows an example of query having search conditions on keyword, texture, and color.

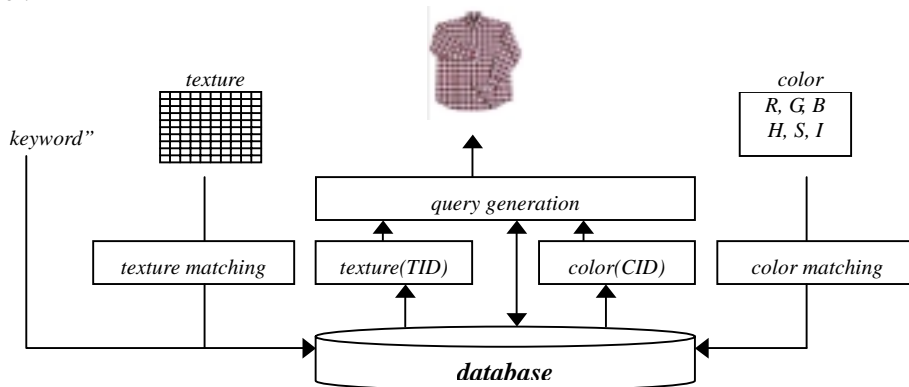


Figure 10. Integrated search procedure

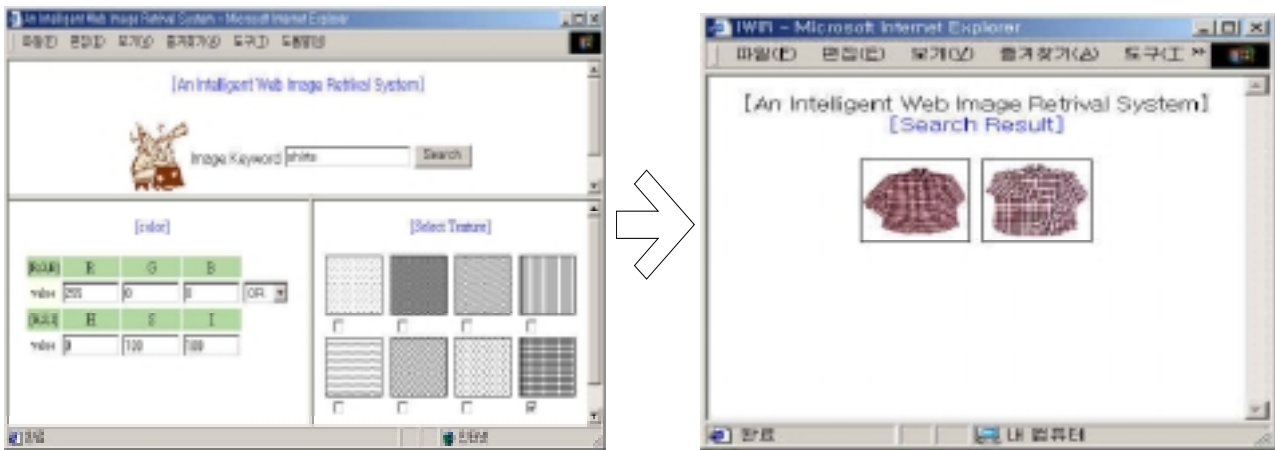


Figure 11. An example of query by keyword + texture + color

5. MINING OF USER USAGE PATTERNS

The IWIR remembers user's preferences, so called user usage patterns. Some users may be interested in 'red' shirts having 'check' patterns. We utilize the log information on the previous queries to improve user's satisfaction on the later query results. The previous query logs are accumulated and used for texture and color mining. Figure 12 shows the outline of query log generation and mining procedure.

For the texture mining, user id and retrieved texture ids for each search transaction are stored in the Log_T table. By applying the association mining procedure[11,12] on the log table, the frequent item sets with their referenced count values can be generated. We choose the texture having the maximum count value as the preferable texture of the user. The texture mining procedure is as follows.

```

procedure Texture_log_analysis ( $T_i := T_{id}$ );
  var i := integer;
  begin
    database connection.open;
    set of Log_T;
  
```

```

for  $i := 1$  to  $n$  do
  select  $T_{id}$  from  $Log\_T$  where  $T_{id}$  like  $T_i$  and  $(U_{id} := session(Uid))$ ; // scanning the  $T_i$  item
  if  $(T_i \neq Null)$  then
    create  $temp\_table\_log\_t$ 
    finding frequent item sets;
    for  $i := 1$  to  $n$  do
      update  $temp\_table\_log\_t$  set  $count := count + 1$  where  $T_{id} = \{T_i\}$ ; // increment count
    endif

  //find maximum number
  select  $Max(count)$  from  $temp\_table\_log\_t$ 
  if  $(Count(rs(0)) > 1)$  then // more than 1 item
    select random item;
  else
    select  $T_i$  item;
  endif
  database connection.close
  set  $rs.nothing$ 
end;
end  $Texture\_log\_analysis$ 

```

Figure 13 shows an example of texture based mining. For the user U_1 , his previous queries show the texture patterns $\{T_1, T_1, T_2, T_2, T_3, T_1, T_3, T_1\}$. By choosing the maximum count value from the frequent item sets, we can find the user U_1 prefers the texture T_1 .

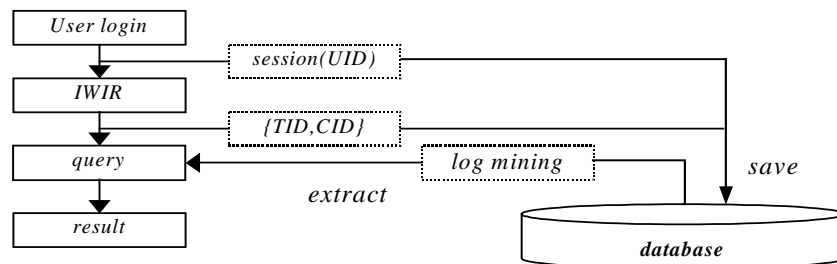


Figure 12. Query log generation and mining procedure

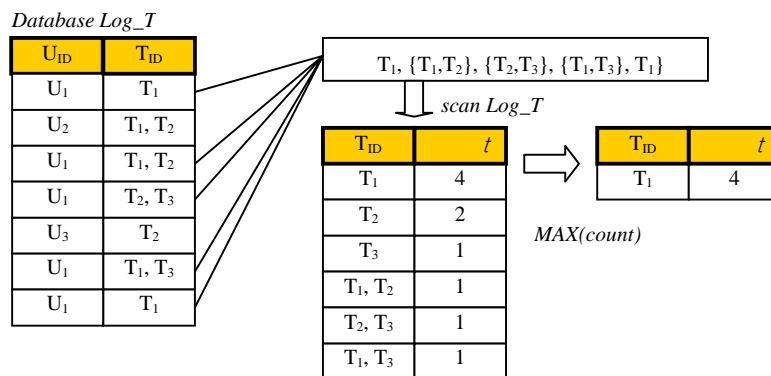


Figure 13. An example of texture mining

The color mining procedure is similar with the texture mining procedure. This procedure is as follows.

```

procedure Color_log_analysis (  $C_j := C_{id}$ ;  $I_i := I_{id}$ );
  var  $j := integer$  ;  $i := integer$ ;
  begin
    database connection.open;
    set of Log_C ;
    for  $j := 1$  to  $n$  do
      select  $I_{id}$ ,  $C_{id}$  from Log_C where  $U_{id} := session(Uid)$  ; // scanning the  $I_i$ ,  $C_j$  items
      create temp_table_log_c
      if ( $C_j \neq Null$ ) then
        result into temp_table_log_c ;
        finding frequent item sets;
        for  $j := 1$  to  $j$  do
          for  $i := 1$  to  $i$  do // increment count
            update temp_table_log_c set count := count + 1 where ( $C_{id} := \{C_j\}$  and  $I_{id} := I_i$ );
          endif
        //find maximum number
        select Max(count) from temp_table_log_c
        if (Count(rs(0)) > 1) then // more than 1 item
          select random item;
        else
          select  $C_j$  item;
        endif
      database connection.close
      set rs.nothing
    end;
  end Color_log_analysis

```

Figure 14 shows an example of color based mining. For the user U_1 , his previous queries show the color patterns $\{C_1, C_1, C_2, C_1\}$ on the image I_1 . and $\{C_2, C_3, C_1, C_3\}$ on the image I_2 . By choosing the maximum count value from the frequent item sets, we can find the user U_1 prefers the color C_1 .

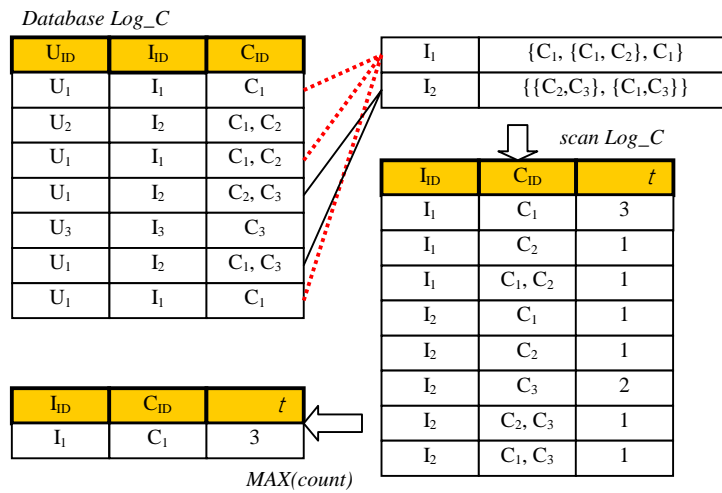
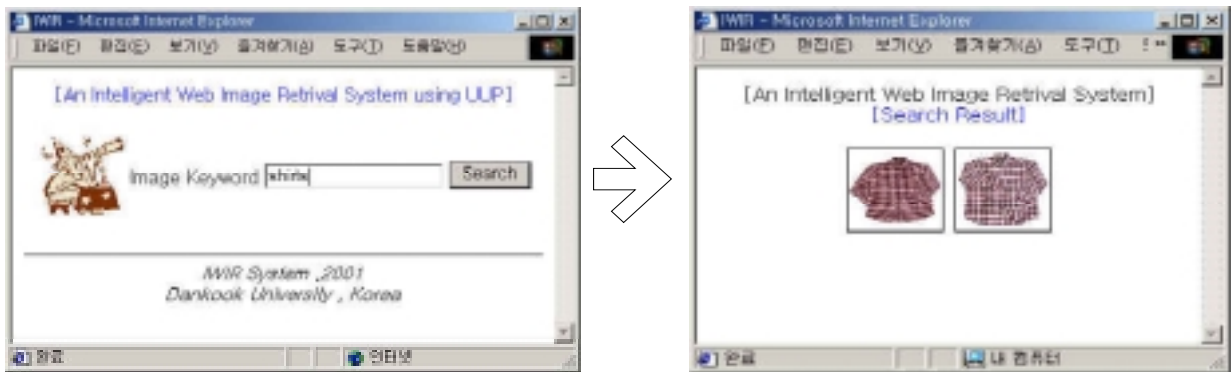


Figure 14. An example of texture mining

These mining results are used to filter the query results. Figure 15 compares the results of query by keyword 'shirts' before and after the user log mining. After the query log mining is applied, the number of matched images is reduced, thus resulting in shorter response time. This will be quite useful to the e-commerce site users, suffering from lots of uninterested search results.



(a) Query by keyword without log mining



(b) Query by keyword reflecting user's preferences

Figure 15. Effects of log mining

6. EXPERIMENTAL RESULTS

We have developed the prototype system on the Pentium-III, Windows 2000 platform using SQL Server 2000. This system was implemented by ASP and Java. We have used 500 web images. Table 1 shows performance results on various search combinations measured in terms of recall and precision. This table shows that the results combining keyword, texture, and color search conditions are better than single feature queries. Also, by applying the user usage pattern, the degree of retrieval correctness has been quite improved even for the case of query by keyword.

Table 1. Performance measures in recall and precision

no	Keyword	Color(RGB,HIS)	Texture	UUP	Recall	Precision
1	"shirts"	-	-	-	72%	42%
2	-	(253,0,0) (100,0,0) ±1.5%	-	-	83%	86%
3	-	-	100...000	-	75%	90%
4	-	(253,0,0) (100,0,0) ±1.5%	100...000	-	83%	96%
5	"shirts"	(253,0,0) (100,0,0) ±1.5%	100...000	-	83%	99%
6	"shirts"	-	-	0	83%	99%

7. CONCLUSIONS

In this paper, we presented an intelligent web image retrieval system IWIR. We have proposed the texture and color based image indexing techniques. To support fast retrieval, we utilized the bit vector indexing for textures representing the presence of each texture pattern by 1 bit. For positional color search, we divided given images into region blocks and stored the average color of each block in system databases. The query can be given by providing keywords, by selecting one or more sample texture patterns, by assigning colors within positional color blocks, or by combining some or all of these factors. The system is designed to remember user's preferences by mining user query patterns, so that it can add more feature information automatically to subsequent user queries.

We are planning to develop an integrated web image search engine by combining the image crawling agent and automatic image caption extractor module. There should be further researches to improve the performance of the feature extractor modules and the matching modules on massive amount of web images.

ACKNOWLEDGEMENTS

This work was funded by KOSEF(Korea Science and Engineering Foundation) under contract no. 98-0102-06-01-3.

REFERENCES

1. D.-H. Lee, D.-Y. Seo, N.-H. Kim, J.-Y. Lee, "Discovery and Application of User Access Patterns in the World Wide Web," in *Proc of the 4th World Congress on Expert Systems*, March 16-20, 1998, pp.321-327.
2. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker and C. Faloutsos, "The QBIC Project: Querying images by content using color, texture, and shape," in *Proc. SPIE Storage and Retrieval for Image and Video Database*, February 1993, pp.173-187.
3. V. N. Gudivada and V. V. Raghavan, "Content-Based Image Retrieval Systems," *IEEE Computer*, 28(9), 1995, pp.18-22.
4. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equiz, "Efficient and Effective Querying by Image Content," *Journal of Intelligent Information System(JIIS)*, 3(3), July 1994, pp.231-262.
5. J.R. Smith and S.-F. Chang, "VisualSEEK: A Fully Automated Content-Based Image Query System," *ACM Multimedia 96*, Boston, MA, 1996.
6. J.Li, J.Z. Wang, G. Wiederhold, "Integrated Region Matching for Image Retrieval," in *Proc. of the 2000 ACM Multimedia Conf., Los Angeles, October, 2000*.
7. J.Li, J.Z. Wang, G. Wiederhold, "Classification of texture and non-textured images using region segmentation," in *Proc. of the Seventh International Conference on image Processing, Vancouver, BC, Canada, September, 2000*.
8. Jia. Wang, et.al., "Color Clustering Techniques for Color Content-B Image Retrieval from Image Database," in *Proc. of the International Conference on Multimedia Computing and Systems*, June 1997, pp. 442-449.
9. J. Han and Y. Fu, "Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Database," in *Proc. AAAI'94 Workshop on Knowledge in Databases (KDD'94)*, Seattle, WA, July 1994, pp.157-168.
10. R. Agrawal and R. Srikant, "Mining Sequential Patterns," in *Proc. of the Int'l Conference on Data Engineering (ICDE)*, Taipei, Taiwan, March, 1995.
11. R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," in *Proc. VLDB*, Santiago, Chile, Sept. 1994, pp.487-499.
12. J. Han, Y. Huang, N. Cercone and Y. Fu, "Intelligent Query Answering by Knowledge Discovery Techniques," *IEEE Transactions on Knowledge and Data Engineering*, 8(3), 1996, pp.373-390.
13. O. Zaiane and J. Han, "WebML: Querying the World-Wide Web for Resources and Knowledge," in *Proc. Int'l Workshop on Web Information and Data Management (WIDM'98)*, Nov. 1998, pp. 9-12.