Image Retrieval System (RGB)

by

Zarith Suraya binti Zahir

Dissertation submitted in partial fulfillment of
The requirements for the
Bachelor of Technology (Hons)
(Information Communication and Technology)

DECEMBER 2005

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Image Retrieval System (RGB)

by

Zarith Suraya binti Zahir

A project dissertation submitted to the
Information Communication and Technology Programme
Universiti Teknologi PETRONAS
In partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION COMMUNICATION AND TECHNOLOGY)

Approved by,

(Mr Ahmad Izuddin b Zainal Abidin)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
December 2005

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

ZARITH SURAYA BINTI ZAHIR

ABSTRACT

In this project, histogram-based search method is investigated in RGB color space. Histogram search characterizes an image by its color distribution but the drawback of a global histogram representation is that information about object location, shape and texture is discarded. Thus, this project showed that images retrieved by using the global color histogram may not be semantically related even though they share similar color distribution in some results. An image retrieval demo system was built to make it easy to test the retrieval performance and to expedite further algorithm investigation. In general, histogram-based retrievals in Euclidean Distance algorithm showed better performance than in other algorithm.

ACKNOWLEDGEMENT

Upon completing the Final Year Project (FYP) of Image Retrieval System focusing on RGB, the author would like to praise to the Al-Mighty Allah for giving the chance to finish the project. Special appreciation and thankful are also dedicated to the FYP supervisor, Mr. Ahmad Izuddin bin Zainal Abidin for his supervision, commitment, professionalism, advice and guidance for completing this project. Also special thank to the author's friend, Mr. Shahrul Zarif for being very helpful in supporting and giving the ideas to the author in finding and researches the project.

The author also wants to give deepest gratitude to Information Technology Department for the support and not forgetting to IT lecturers, technicians, the author's parents and colleagues for all the encouragement. Last but not least, to those who help directly or indirectly in completing this project. Thank you very much, your contributions are highly regarded and would be remembered.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Over the last few decades, a wide variety of government and industrial organizations have been collecting photographic and image data. For example, NASA's EOSDIS project has collected tremendous amount of data pertaining to the earth. Likewise, hospitals regularly collect x ray and CAT scan data, leading to very large banks of such images. For the most part, many of these images are stored in non electronic forms, such as 4 x 6-inch photographs or x-rays. It is recently, with the drop in prices of digitization equipment, that these images have started being digitized. As digitization becomes more and more widespread, we're confronted with a greater and greater need to manage such images efficiently.

In the relational model of data, we are typically confronted with a situation where queries are posed textually. In other words, the user types in a query and obtains an answer in response. However, consider the case of the police investigator may have in from of him a surveillance photograph of someone whose identity he may not know, but wishes to determine. Thus, he may wish to ask a query of the form "Here's a picture of a person. Can you retrieve all pictures from the image database that are similar to this person and tell me the identities of the people in the pictures you return to me?" This query is fundamentally different from ordinary queries for two reasons. First, the query includes a picture as part of the query. Second, the query asks about "similar" pictures and hence uses a notion of "imprecise match," whose definition needs to be precisely articulated. (It is possible to reason precisely about imprecise data!).
(James Z. Wang, 2001)

1.2 Problem Statement

This project falls in Information Retrieval Area. As we all know, if we wanted to retrieve text information, we will find an exact matching to it because we can use SQL statement. What about image retrieval? Will it be the same like text retrieval? We do not have specific query language for retrieving image.

Querying an image is different from querying textual database. In relational model of data, we are typically confronted with a situation where queries are posed textually. In other words, the user types in a query and obtains an answer in response. It also needs some calculation together with some algorithm to find the exact match of the query image.

Moreover, large image database are difficult to browse with text search if the user wants to search images which have similar color content.

Apart from that, it requires large space of storage to save image files. And the file will be saved manually, means either in a non-electronic form or save it in one large server. This will cause to a low quality time and less convenience for user to find the file one by one.

In addition, most conventional image databases are text-annotated. As a result, image retrieval is based on keyword searching. Text annotated images are simple and easy to manipulate. However, there are two major problems with this method. Firstly, in creating a keyword for a large amount of images is time consuming. Furthermore, the keywords are inherently subjective and not unique. For large collections, it is time consuming to create and maintain. Also, text-annotated only provides hit or miss type searching. If the user does not specify the right keywords, the desired images may be forever unreachable.

Therefore, a new smart system for retrieving image database focusing on RGB is required. With the Image Retrieval System (RGB), image retrieval will be more effective. By using a query image, it will help find any other candidate images that are particularly similar to the query image based on the threshold specified. It is a suitable system that capable for overcoming this problem. Initially, all other images will be collected and stored in database using Microsoft Access during the first time of data-entry.

1.3 Objective

The objectives of this project are as follows:

- To build a prototype to test retrieval performance
- To solve the problem of image retrieval
- To ensure the achieved time within a reasonable time
- To make sure the matching is as close as 100%

Image retrieval demo was built to make it easy to test the retrieval performance and to expedite further algorithm investigation through developing engine that enables image to be retrieved and employ one technique which the algorithm I used.

1.4 Scope of Study

The timeframe provided for this project is limited to only 15 weeks. Therefore there are certain limitations that had to be specified in order to help narrow the nature of research.

A few assumptions are going to be made about variables used through the project. They are:

- The images must be in JPG form as it will save the computation time
- The images must be resizing to 120 X 160; also to save the computation time
- The retrieval method is only focusing on RGB; it will eliminate other signature of the image
- Only Histogram Euclidean Distance Algorithm is used to calculate and compare the image histogram

With the assumptions mentioned above, it will make the scope of this project narrower. With few adjustments and enhancements in the future, the author hopes that this project can be expanded to be used in the real industry out there. It might sound a little bit carried away, but the author imagined that this project / product might be useful in the recognition technology.

The area of area itself is very huge and large, but with today's changing demands, effective computer vision system become more and more essential. Application of this program could be in the biological entities class, such as classification of unknown cancer disease.

CHAPTER 2
LITERATURE REVIEW

Taken from the project are the following keywords and algorithm. Basically these are what contained, used during the whole project development:

2.1 Histogram – Based Image Search

The color histogram for an image is constructed by counting the number of pixels of each color. In these studies the developments of the extraction algorithms follow a similar progression: (1) selection of a color space, (2) resizes the image, (3) computation of histograms, (4) derivation of the histogram distance function, (5) comparing the image in database. Each of these steps may be crucial towards developing a successful algorithm (J.R. Smith & S. –F. Chang, 1995).

An image histogram refers to the profitability mass function of the image intensities. This is extended for color images to capture the joint probabilities of the color channels. More formally, the color histogram is defined by,

$$h_{A, B, C}(a, b, c) = \mathrm{N}.\mathbf{Pr}ob(A = a, B = b, C = c)$$

Where A, B and C represent the three color channels (R, G and B) and N is the number of pixels in the image. Computationally, the color histogram is formed by discretizing the colors within an image and counting the number of pixels of each color.

2.2 Color Space Red, Blue & Green (RGB)

A color space is defined as a model for representing color in terms of intensity values. Typically, a color space defines a one- to – four dimensional spaces. A color component, or color channel, is one of the dimensions. A color dimensional space (i.e. one dimension per pixel) represents the gray – scale space.

The RGB color model is composed of the primary colors Red, Green and Blue. This system defined the color model that is used in most color CRT monitors and color raster graphics. They are considered the "additive primaries" since the colors are added together to produce the desired color. The RGB model uses the Cartesian coordinate system as shown in Figure 1.(a). Notice the diagonal from (0, 0, 0) black to (1, 1, 1) white which represents the grey-scale. Figure 1.(b) is a view of the RGB color model looking down from "White" to origin.

The RGB color space (Red, Green and Blue) is well supported on image processing programs thus it is not difficult to make a program that reads an image from a disk to the computer's RAM.
(Petteri Kerminen & Moncef Gabbouj – 2001)

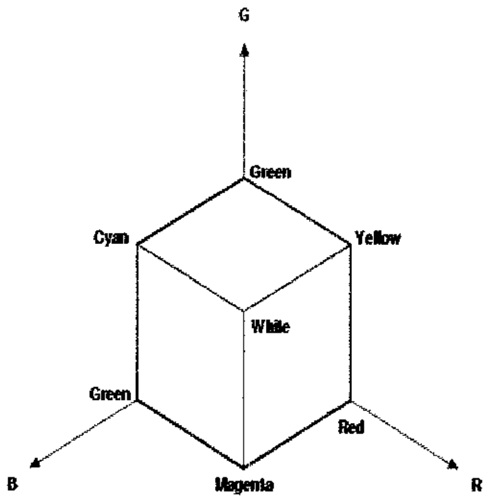


Figure 1.(a) RGB coordinates system

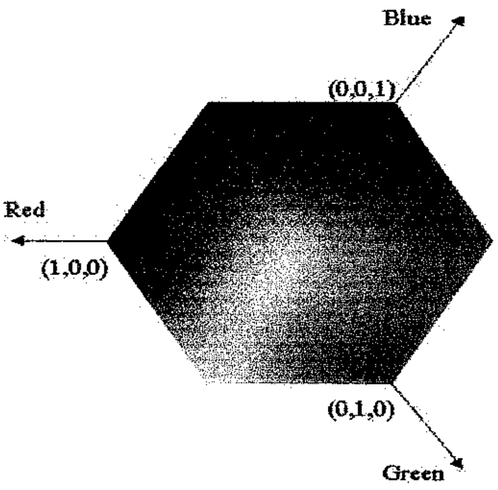


Figure 1.(b) RGB color model

2.3 Histogram Euclidean Distance Algorithm

There are several distance formulas for measuring the similarity of color histograms. In general, the techniques for comparing probability distributions such as the *kolmogoroff – smirnov* test are not appropriate for color histograms. This is because visual perception determines similarity rather than closeness of the probability distributions. Essentially, the color distance formulas arrive at a measure of similarity between images based on the perception of color content. The distance formula that has been used for image retrieval is histogram Euclidean distance.

According to (M. J. Swain & D. H. Ballard, 1991)

There is only comparison between the identical bins in the respective histograms. Two different bins may represent perceptually similar colors but are not compared crosswise. All bins contribute equally to the distance.

Let h and g represent two color histograms. The Euclidean distance between the color histograms h and g can be computed as:

$$d^2(h, g) = \Sigma_A \ \Sigma_B \ \Sigma_C \ (h(a,b,c) - g(a,b,c))^2$$

In this distance formula, there is only comparison between the identical bins in the respective histograms. Two different bins may represent perceptually similar colors but are not compared crosswise. All bins contribute equally to the distance.

2.4 Generation of image DB

The image data used in this experiment were captured by the author and some are taken from website. However, in order to reduce the computation time of the whole process, only 15 images were used and their sizes were reduced to 120 x 160 pixels. Those 15 images have numbers as their names from 1 with JPEG – compressed for convenience and for saving storage.

CHAPTER 3
METHODOLOGY / PROJECT WORK

3.1 Methods of Development

3.1.1 System Development Life Cycle

System Development Life Cycle (SDLC), which contains four core stages, will be used in the project. The four basic stages are Planning, Analysis, Design and Implementation. Each of the stages will have its own specifications and task lists.

Table 1 : Suggested Activities Based on SDLC

Phase	Activity
Plan	• Do research based on project title, gather necessary information • Plan scope of time and scope of study
Analysis	• Analyze available data and information • Segregate information according to necessary class
Design	• Design interface and working prototype • Prepare documentation • Do testing on product
Implementation	• Finalize the product • Product presentation

For the product part, the author plan to use a prototype model. The model will help during the development, as series of testing can be done onto it. This is because the author wants to make sure the prototype will be completed and can work as planned.

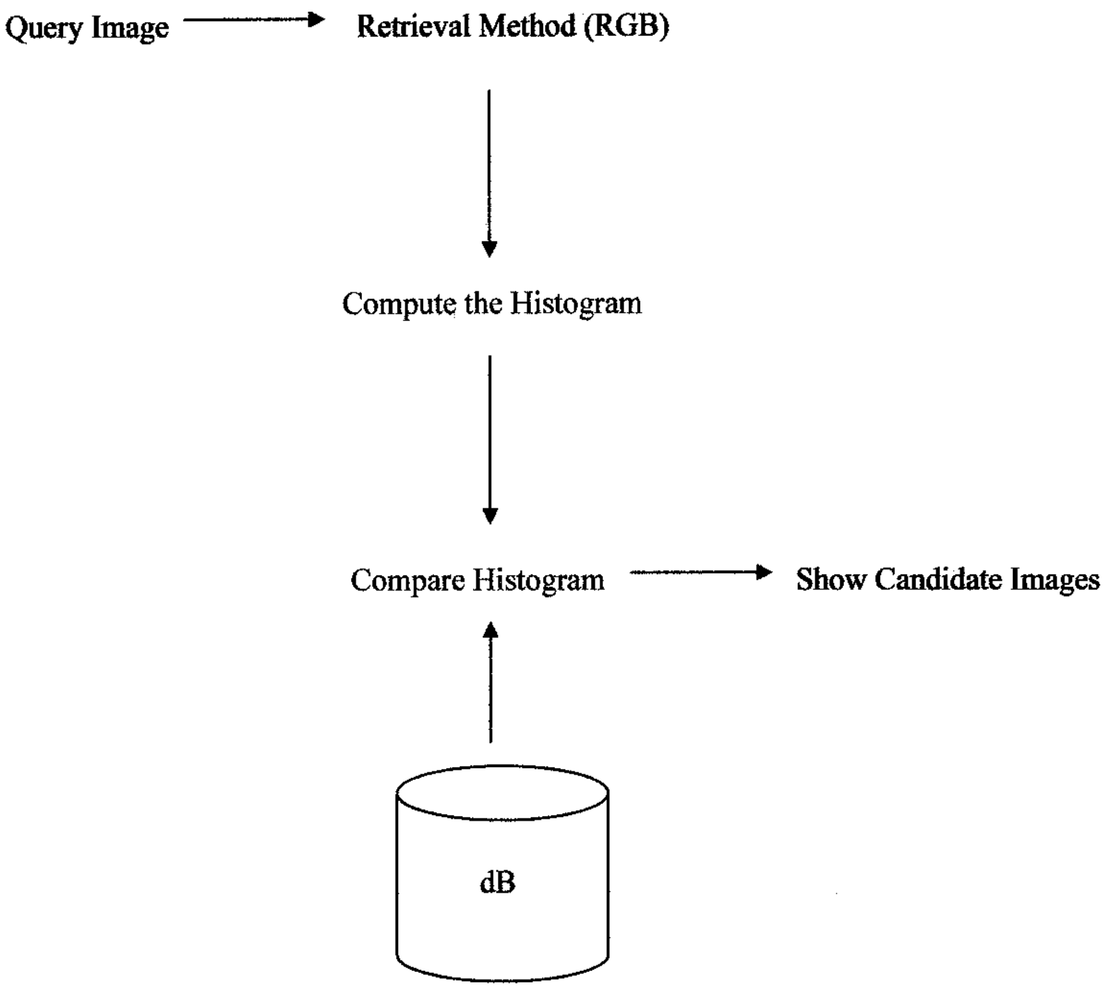


Figure 2 : Prototype Methodology

Following is the procedure for image querying:

Steps 1 : The user loads a query image, adjusts the size and sends the query image.

Step 2 : When the query server receives the query image, it employs the RGB retrieval method

Step 3 : It then computes the histogram of the query image and also the images in the database using the Histogram Euclidean Distance algorithm pixel by pixel.

Step 4 : The results of the computed histogram of the query image, then is compared with the histogram of the images in the database

Step 5 : The mechanism ranks the most similar database images as query image according to the similarity order that the user specified in the threshold

Step 6 : Finally, the query server sends the recommended candidate images from image database to the query interface at the user side

The difference from ordinary queries is that:

- It includes a picture as part of a query
- The query asks about "similar" picture and hence uses a notion of "imprecise match"

The only possibility to build a histogram is exhaustive pixel by pixel. Once it has been done and saved on the disk, it can be used for analysis. Comparison between images is not possible, if the histograms don't have the same number of bins and are not converted to the same color space.

Once got the title of the Final Year Project, the author had done some researches to study and understand the background of this project. This also to find the most desire algorithm that will be used in this project based on their accuracy, precision and time. Since the author is new with the Microsoft Access and Visual Basic, it took much time to understand and get familiar with the software. After that, the database of images has been created using Microsoft Access and the Graphical User Interface is designed using Visual Basic.

The main part of this project is to create the coding for querying, computing, matching and retrieving candidate images. This is done according to the user interface to make sure the project work successfully. The last part is to analyze and test the system by conducting an experiment within 25 images.

3.2 Tools Used

3.2.1 Hardware

There is no specific hardware that is being used for project other that the basic peripherals such as monitor, CPU and mouse.

3.2.2 Software

3.2.2.1 Microsoft Visual Basic 6.0

Microsoft Visual Basic is an object-oriented/event driven programming language that is easy enough for a nonprogrammer to use, yet sophisticated enough to be used by professional programmers. With Visual Basic it takes just a few clicks of the mouse to include standard Windows object such as buttons, list boxes, scroll bars and icons in Windows application. Once object are created, the programmer then concentrates on writing the specific instructions telling each objects how to respond when clicked, double-clicked, scrolled and so on.

In this project, the Visual Basic is used to create Graphical User Interface to retrieve the data from database.

3.2.2.2 Microsoft Access 2003

Microsoft Access 2003 provides a powerful set of tools that are sophisticated enough for professional developers and easy to learn for new users. Access 2003 supports a variety of data formats including Extensible Markup Language (XML), Object Linking and Embedding (OLE) and Open Database Connectivity (ODBC).

Using Access 2003, the database will be created. The database application manages the data in a record and field format. A field is a column of data inside a file. The database does not necessarily store the data in a table-like format but the database makes the data to appear in that format in the program. This will be completed by applying the Microsoft Visual Basic to retrieve data in the record and field format. This software stores all related database field in a single global file in .mdb format. In the file, the individual groups of records and fields are called tables. A table is a logical collection of data in database.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Data Gathering and Analysis

During the early stage of project, a lot of data was gathered for the purpose of training. Apart from finding materials such as study, journals, reference books, the author also looked and tried a few set of images. All images are first saved in Microsoft Access in a trial – and – error mode. The outputs are monitored.

As Microsoft Access is capable of saving a lot of image file type, the author did gathered a few types of images as sample data. Some of them were Joint Photographic Expert Group (JPEG) type, Standard Windows Bitmap (BMP) and Graphic Interchange Format (GIF). But the author only use JPG image to save computation time.

In the experiment, an image retrieval demo system in Figure 3 was built to test image retrieval efficiently. If the user selects and image from file, the retrieval system finds the best five to six candidate images based on the threshold specified by user including the same image as the query image. Internally, it has information about all sorted indexes of images in DB in the order of increasing histogram distance. But, it shows only a few best candidates based on the threshold.

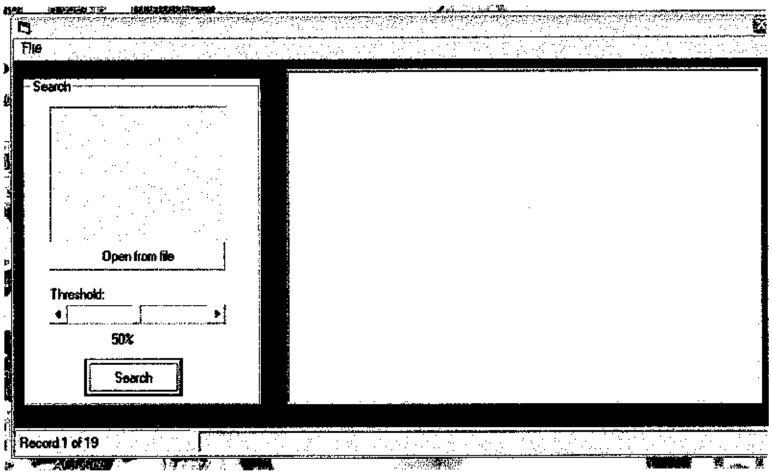


Figure 3: System Interface

4.1.1 Computational Complexity

The color histogram for an image is constructed by counting the number of pixels of each color. Color histogram is defined by,

$$h\ A, B,C\ (a,\ b,\ c) = \mathrm{N}.\mathbf{Pr}ob(A = a, B = b, C = c)$$

where A, B and C represent the three color channels and N is the number of pixels in the image. Using histogram-based search methods, it characterizes an image by its color distribution. Computationally, the color histogram is formed by discretizing the colors within an image and counting the number of pixels of each color.

Many histogram distances have been used to define the similarity of two two color histogram representation. Euclidean Distance and its variations are the most commonly used.

Let h and g represents the two color histogram

$$d\,2(h,\ g) = \Sigma A\ \ \Sigma B\ \ \Sigma C\ (h(a,b,c) - g(a,b,c))2$$

Euclidean Distance Algorithm only compare between the identical bins in the respective histograms.

4.1.2 Simulation Results

VISUAL BASIC 6.0 simulations have been done for all images on Pentium (R) 4 CPU 2.80 GHz with 448 MB of RAM and Windows XP Workstation operation system. And histograms have been built for 15 images (size 120 x 160)

Table 2 : Time taken to compute 15 images

Color space	Total time	Time / image
RGB	1:20:18	5.3secs

From this table I can conclude that, these times were proportional to the computational complexity analysis.

The histogram build process will get much more speed when it is implemented with C – Language.

4.1.3 Visual Simulation Result

Important : the visual color similarity of retrieved images

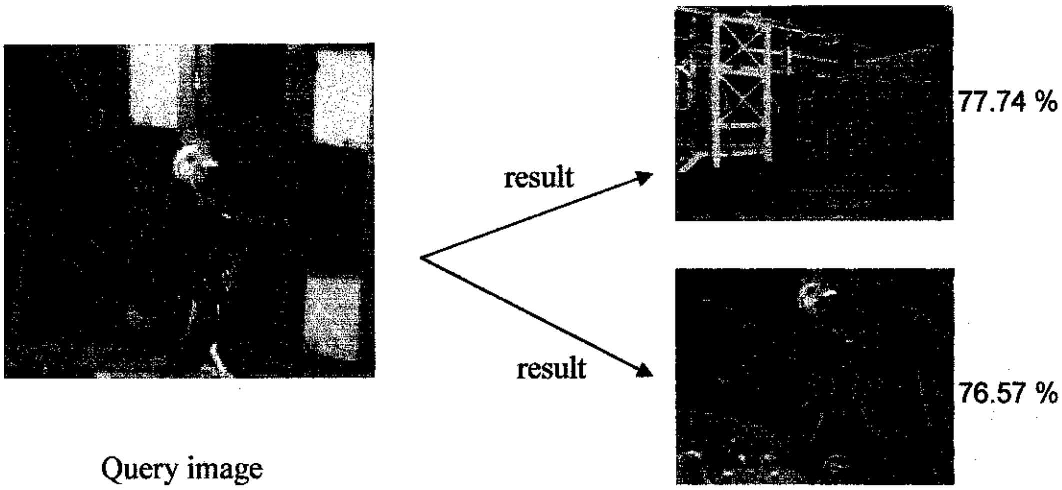
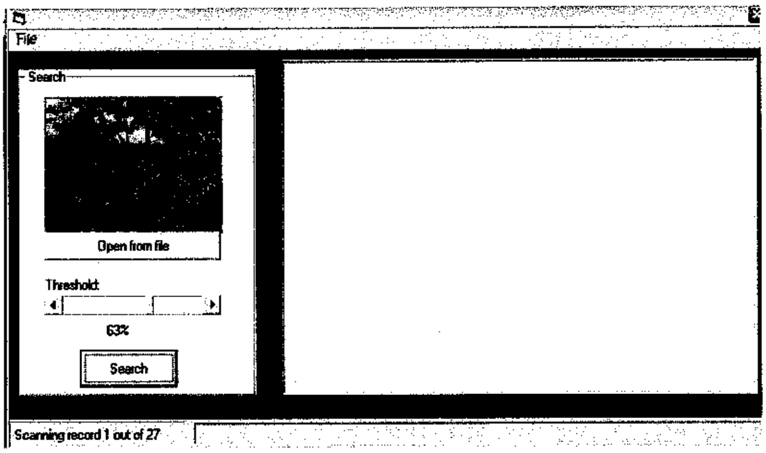


Figure 4 : Examples of results from query image

4.2 Discussion

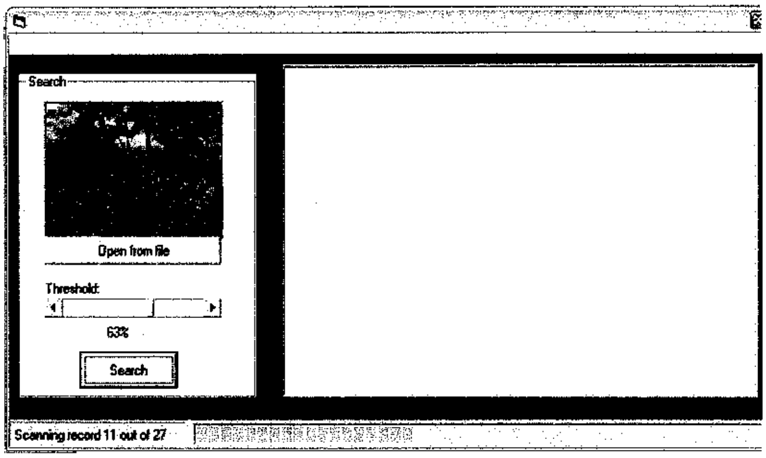


Threshold:
63%

Scanning record 1 out of 27

Figure 5 : System initializing the query image and threshold specified

The query server will start initializing the record images in the database as shown in Figure 5. On Figure 6 it will then starts scanning all the record to find the similar matching based on the threshold user has specified before.



Threshold:
63%

Scanning record 11 out of 27

Figure 6 : System scanning the database for candidate images

4.2.1 Outcome of Testing

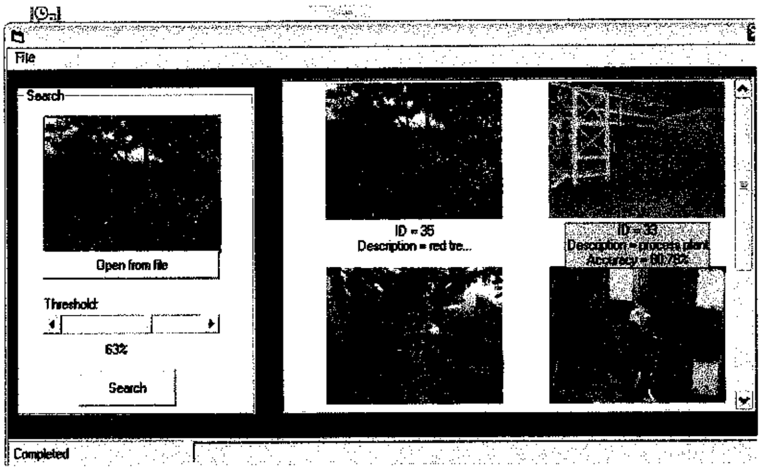
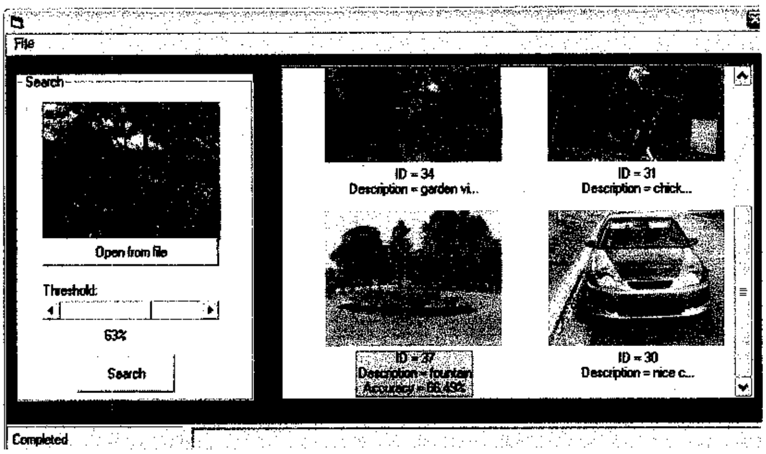


Figure 7 (a) and (b) : Results of the retrieval system



On Figure 7 (a) and (b) the server send the recommended candidate images from image database to the query interface at the user side.

In order to measure the retrieval effectiveness of an image retrieval system, a threshold is used. Although this effectiveness measure is useful to test retrieval system objectively, it cannot exclude subjectivity inherent in image retrieval system. It's because the category itself was classified by humans. However, it is still most popular to test an information retrieval system. In the experiment, the threshold will be stated by the user and the candidate images will also list out the percentage of RGB of that image.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The main focus of this project has been on RGB color spaces. I have described the method for extracting from a color image a small set of color – texture features that has the desirable property of preserving perceptual similarity, i.e., representing perceptually similar images by indices that lie close together in the feature space.

As the histogram search characterized an image by only its color distribution, it will leads to some drawbacks like stated earlier. The information about the object location, shape and texture is discarded. The results are not semantically related even though they share similar color distribution in some results.

5.2 Recommendation for Future Enhancements

The project can be considered as an attempt to venture into computer vision area. However, it is done on a small-scale level. The author would like to recommend enhancement of the project in terms of:

5.2.1 Interface

The interface of the prototype is very minimal at this time. This is because the prototype does not have a lot of function other than applying the searching and saving databases. If there is to be an expansion of the project, maybe the interface can be modified in terms of the functions addition. Other than that, studies on other area such as Human-Computer Interaction (HCI) may be included as well.

5.2.2 Expansion of Functions

Currently, the only function that is available in the project is to search in the database for the similar image or imprecise image of the query one. In the future, the author would like to look into other tools of image preprocessing analysis, such as texture and shape, edge / line detection, segmentation and etc. And that expansion will be very beneficial to users or industry later.

5.2.3 Widen the Scope and Limitations

As stated in the Chapter 1, there are 4 limitations applied to the variables used. This is to make the scope of the project smaller, other than to ensure the success of the project. In the future, it will be a lot more meaningful if the scope can be expanded. For example, image used in the project are not only for JPG image, but any image format.

Other than that, the author would be very delightful if the project can be translated into another language such as C or C++. It will provide more ability to cross-platform, especially for users who have not used VISUAL BASIC before. Furthermore, the codes can be shared as open-source information. And the calculation of the algorithm might take lesser time to compute.

REFERENCES

(James Z. Wang, 2001) James Z. Wang, *Integrated Region – Based Image Retrieval*, Boston, Kluwer Academic Publishers,

(Smith, J.R. and Chang, S-F., 1995 - 1996), *Automatic Image Retrieval Using Color and Texture*,
http://www.ctr.columbia.edu/jrsmith/html/pubs/PAMI/pami_final_1.htm

(M. J. Swain & D. H. Ballard, 1991) *Color Indexing*, International Journal of Computer Vision

(Ponyton, Charles A., 1999) Ponyton, Charles A.; *Frequently Asked Questions about Color ("colorspace-faq");*
"Frequently Asked Question about Color". Maintained and available on the Internet in text (ASCII), Postscript and Adobe Acrobat formats

(David Travis, 1991) *Effective Color Displays, Theory and Practice*, Harcourt, Brace & Javanovich Publishers

(C. Faloutsos, et. Al., August 3,1993) *Efficient and Effective Querying by Image Content*, IBM RJ 9453(83074)

APPENDICES

form database

```
Dim WithEvents adoPrimaryRS As Recordset
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean



Private Sub Form_Load()
  Dim db As Connection
  Set db = New Connection
  db.CursorLocation = adUseClient
  db.Open "PROVIDER=Microsoft.Jet.OLEDB.3.51;Data Source=" & DBaseName & ";"

  Set adoPrimaryRS = New Recordset
  adoPrimaryRS.Open "select ID,DESCRIPTION,IMAGES from tbl_images", db,
adOpenStatic, adLockOptimistic

  Dim oText As TextBox
  'Bind the text boxes to the data provider
  For Each oText In Me.txtFields
    Set oText.DataSource = adoPrimaryRS
  Next
  Dim oPic As PictureBox
  'Bind the ole controls to the data provider
  For Each oPic In Me.picFields
    Set oPic.DataSource = adoPrimaryRS
  Next

  mbDataChanged = False
End Sub

Private Sub Form_Resize()
  On Error Resume Next
  lblStatus.width = Me.width - 1500
  cmdNext.Left = lblStatus.width + 700
  cmdLast.Left = cmdNext.Left + 340
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
  If mbEditFlag Or mbAddNewFlag Then Exit Sub

  Select Case KeyCode
    Case vbKeyEscape
      cmdClose_Click
    Case vbKeyEnd
      cmdLast_Click
    Case vbKeyHome
      cmdFirst_Click
    Case vbKeyUp, vbKeyPageUp
      If Shift = vbCtrlMask Then
        cmdFirst_Click
      Else
        cmdPrevious_Click
      End If
    Case vbKeyDown, vbKeyPageDown
      If Shift = vbCtrlMask Then
        cmdLast_Click
    Else
     cmdNext_Click
    End If
  End Select
End Sub

Private Sub Form_Unload(Cancel As Integer)
  Screen.MousePointer = vbDefault
End Sub

Private Sub adoPrimaryRS_MoveComplete(ByVal adReason As
ADODB.EventReasonEnum, ByVal pError As ADODB.Error, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
  'This will display the current record position for this recordset
  lblStatus.Caption = "Record: " & CStr(adoPrimaryRS.AbsolutePosition)
End Sub

Private Sub adoPrimaryRS_WillChangeRecord(ByVal adReason As
ADODB.EventReasonEnum, ByVal cRecords As Long, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
  'This is where you put validation code
  'This event gets called when the following actions occur
  Dim bCancel As Boolean

  Select Case adReason
  Case adRsnAddNew
  Case adRsnClose
  Case adRsnDelete
  Case adRsnFirstChange
  Case adRsnMove
  Case adRsnRequery
  Case adRsnResynch
  Case adRsnUndoAddNew
  Case adRsnUndoDelete
  Case adRsnUndoUpdate
  Case adRsnUpdate
  End Select

  If bCancel Then adStatus = adStatusCancel
End Sub

Private Sub cmdAdd_Click()
  On Error GoTo AddErr
'picfields(2).
lblPic.Visible = True
  With adoPrimaryRS
    If Not (.BOF And .EOF) Then
      mvBookMark = .Bookmark
    End If
    .AddNew
    lblStatus.Caption = "Add record"
    mbAddNewFlag = True
    SetButtons False
  End With


  Exit Sub
AddErr:
  MsgBox Err.Description
End Sub

Private Sub cmdDelete_Click()
```

```
  On Error GoTo DeleteErr
  With adoPrimaryRS
    .Delete
    .MoveNext
    If .EOF Then .MoveLast
  End With
  Exit Sub
DeleteErr:
  MsgBox Err.Description
End Sub

Private Sub cmdRefresh_Click()
  'This is only needed for multi user apps
  On Error GoTo RefreshErr
  adoPrimaryRS.Requery
  Exit Sub
RefreshErr:
  MsgBox Err.Description
End Sub

Private Sub cmdEdit_Click()
  On Error GoTo EditErr

  lblStatus.Caption = "Edit record"
  mbEditFlag = True
  SetButtons False
  Exit Sub

EditErr:
  MsgBox Err.Description
End Sub
```

```
Private Sub cmdCancel_Click()
 On Error Resume Next
lblPic.Visible = False
 SetButtons True
 mbEditFlag = False
 mbAddNewFlag = False
 adoPrimaryRS.CancelUpdate
 If mvBookMark > 0 Then
  adoPrimaryRS.Bookmark = mvBookMark
 Else
  adoPrimaryRS.MoveFirst
 End If
 mbDataChanged = False
cmdOpen.Visible = False
End Sub

Private Sub cmdUpdate_Click()
 On Error GoTo UpdateErr
lblPic.Visible = False
 With adoPrimaryRS
SaveBitmap adoPrimaryRS, "IMAGES", strfilepath
 adoPrimaryRS.UpdateBatch adAffectAll
End With
 If mbAddNewFlag Then
  adoPrimaryRS.MoveLast           'move to the new record
 End If

 mbEditFlag = False
 mbAddNewFlag = False
 SetButtons True
 mbDataChanged = False
```

```
  Exit Sub
UpdateErr:
  MsgBox Err.Description
End Sub

Private Sub cmdClose_Click()
  Unload Me
End Sub

Private Sub cmdFirst_Click()
  On Error GoTo GoFirstError

  adoPrimaryRS.MoveFirst
  mbDataChanged = False

  Exit Sub

GoFirstError:
  MsgBox Err.Description
End Sub

Private Sub cmdLast_Click()
  On Error GoTo GoLastError

  adoPrimaryRS.MoveLast
  mbDataChanged = False

  Exit Sub

GoLastError:
  MsgBox Err.Description
End Sub

Private Sub cmdNext_Click()
  On Error GoTo GoNextError

  If Not adoPrimaryRS.EOF Then adoPrimaryRS.MoveNext
  If adoPrimaryRS.EOF And adoPrimaryRS.RecordCount > 0 Then
    Beep
    'moved off the end so go back
    adoPrimaryRS.MoveLast
  End If
  'show the current record
  mbDataChanged = False

  Exit Sub
GoNextError:
  MsgBox Err.Description
End Sub

Private Sub cmdPrevious_Click()
  On Error GoTo GoPrevError

  If Not adoPrimaryRS.BOF Then adoPrimaryRS.MovePrevious
  If adoPrimaryRS.BOF And adoPrimaryRS.RecordCount > 0 Then
    Beep
    'moved off the end so go back
    adoPrimaryRS.MoveFirst
  End If
  'show the current record
  mbDataChanged = False
```

```
 Exit Sub

GoPrevError:
 MsgBox Err.Description
End Sub

Private Sub SetButtons(bVal As Boolean)
 cmdAdd.Visible = bVal
 cmdEdit.Visible = bVal
 cmdUpdate.Visible = Not bVal
 cmdCancel.Visible = Not bVal
 cmdDelete.Visible = bVal
 cmdClose.Visible = bVal
 cmdRefresh.Visible = bVal
 cmdNext.Enabled = bVal
 cmdFirst.Enabled = bVal
 cmdLast.Enabled = bVal
 cmdPrevious.Enabled = bVal
End Sub

Private Sub lblPic_DblClick()
Call picFields_DblClick(0)
End Sub

Private Sub picFields_DblClick(Index As Integer)
If mbAddNewFlag = True Or mbEditFlag = True Then
Call OpenFile(picFields(2), Me)
End If
End Sub
```

form search

```
Private WithEvents adoPrimaryRS As Recordset
Private DbaseLoaded As Boolean

Private Sub cmdOpen_Click()
Call OpenFile(picKey, Me)
End Sub

Private Sub Command1_Click()
If Not DbaseLoaded Then
MsgBox "No databse loaded", vbInformation, "No Database"
Exit Sub
End If

'Set ListView1.ListItems = Nothing
ListView1.ListItems.Clear
ImageList1.ListImages.Clear

adoPrimaryRS.Close
adoPrimaryRS.Source = "SELECT * FROM tbl_images"
adoPrimaryRS.Open
Set picRef.DataSource = adoPrimaryRS

Dim key() As PIXEL
Dim result() As PIXEL
ReDim key(picKey.ScaleHeight, picKey.ScaleWidth)
ReDim result(picRef.ScaleHeight, picRef.ScaleWidth)

Dim keyHistogram() As HISTOGRAM
ReDim keyHistogram(Red To Blue)
```

```
Dim resultHistogram() As HISTOGRAM
ReDim resultHistogram(Red To Blue)

Dim record As Integer



Dim sum(Red To Blue) As Long


Dim recResult() As RGBRECORD
ReDim recResult(1 To adoPrimaryRS.RecordCount)



Dim temp() As Long
ReDim temp(0 To 255)

StatusBar1.Panels(1).Text = "Initializing source"
'Get color value for each pixel and break into its component(RGB)
key() = calculate(picKey)

Dim i As Color
DoEvents
ProgressBar1.Min = 0
ProgressBar1.Max = Blue
For i = Red To Blue 'for color component from Red to Blue

'Calculate the histogram of each colr component and return an array of frequency from 0-
255
temp() = computeHistogram(key, i, picKey.ScaleWidth, picKey.ScaleHeight)

'copy the array
Call CopyArray(temp, keyHistogram(i).Value)
```

```
ProgressBar1.Value = ProgressBar1.Value + 1
Next i

'reset back progress bar
ProgressBar1.Min = 0
ProgressBar1.Max = adoPrimaryRS.RecordCount
ProgressBar1.Value = 0

For record = 1 To adoPrimaryRS.RecordCount
   StatusBar1.Panels(1).Text = "Scanning record " & record & " out of " &
adoPrimaryRS.RecordCount
   result() = calculate(picRef)

   For i = Red To Blue

      temp() = computeHistogram(result, i, picRef.ScaleWidth, picRef.ScaleHeight)
      Call CopyArray(temp, resultHistogram(i).Value)
      sum(i) = calculateDistance(keyHistogram(i).Value, resultHistogram(i).Value)

   Next i



   recResult(record).record = adoPrimaryRS.Fields("ID")
   recResult(record).sum = (sum(Red) + sum(Green) + sum(Blue)) ^ (1 / 2)
   adoPrimaryRS.MoveNext
   ProgressBar1.Value = ProgressBar1.Value + 1
Next record

Call Sort(recResult)
StatusBar1.Panels(1).Text = "Completed"
ProgressBar1.Value = 0
```

```
Dim k As Integer
Dim counter As Integer
Dim itmx As ListItem
Dim similarity As Long
For k = LBound(recResult) To UBound(recResult)

  If ((FAILURE - recResult(k).sum) / FAILURE) * 100 > sbarThresh.Value Then
    adoPrimaryRS.Close
    adoPrimaryRS.Source = "SELECT * FROM tbl_images WHERE ID=" &
recResult(k).record
    adoPrimaryRS.Open
    Set picRef.DataSource = adoPrimaryRS
    counter = counter + 1
    ImageList1.ListImages.Add counter, , picRef.picture
    Set ListView1.Icons = ImageList1
    Set itmx = ListView1.ListItems.Add(, , , counter)
    ListView1.ListItems(counter).Text = "ID = " & adoPrimaryRS.Fields("ID") &
vbNewLine & _
                    "Description = " & adoPrimaryRS.Fields("DESCRIPTION") &
vbNewLine & _
                    "Accuracy = " & Format(((FAILURE - recResult(k).sum) /
FAILURE) * 100, "0.00") & "%"
    ListView1.Refresh
  End If
Next k

MsgBox ImageList1.ListImages.Count & " images matched"
End Sub
Private Sub Form_Load()
DbaseLoaded = False
```

```
sbarThresh.Value = 50

End Sub

Private Sub mnuDBase_Click()
On Error GoTo errhandler

DBaseName = GetDbase(Me)


Dim db As Connection
  Set db = New Connection
  db.CursorLocation = adUseClient
  'Connection String
  'Definition for MS ACCESS database - comment this if using SQL Server
  db.Open "PROVIDER=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=" & DBaseName 'App.Path & "\db1.mdb;"

  'Definition for SQL Server - uncomment if using SQL Server
  'struid = Admin 'username
  'strpwd = Admin 'password
  'db.Open "PROVIDER=MSDataShape;Data PROVIDER=MSDASQL;" & _
    "dsn= Your_DSN_Name;uid=" & _
      struid & ";pwd=" & _
      strpwd & ";database=Your_Database_Name;"

  Set adoPrimaryRS = New Recordset
  adoPrimaryRS.Open "tbl_images", db, adOpenStatic, adLockOptimistic
```

```
 'Bind the ole controls to the data provider

 Set picRef.DataSource = adoPrimaryRS
DbaseLoaded = True
mnuEdit.Enabled = True
errhandler:
End Sub

Private Sub mnuEdit_Click()
Load frmDbase
frmDbase.Show
End Sub

Private Sub mnuExit_Click()
End
End Sub

Private Sub sbarThresh_Change()

lblThresh.Caption = sbarThresh.Value & "%"
End Sub

Private Sub sbarThresh_Scroll()
lblThresh.Caption = sbarThresh.Value & "%"
End Sub
```