# EECS332 Report

EECS332 Report 1

1 My Personal views 2

2 ImageGoogle 5

2.1 Introduction 5

2.2 Color-based retrieval 6

2.2.1 RGB Color Model 7

2.2.2 Color Histogram 7

2.2.3 Implementation 8

2.3 Edge-based retrieval 9

2.3.1 Edge Histogram 9

2.3.2 Implementation 9

2.4 Gist Descriptor 10

2.4.1 Implementation 11

2.5 Similarity Calculation 11

2.5.1 Euclidean Distance 11

2.5.2 Histogram Intersection Distance 12

2.6 Image Database 13

3 Results 14

3.1 ImageGoogle 14

3.2 Query Results 15

3.2.1 Global Color Histogram 15

3.2.2 Block-based Color Histogram 16

3.2.3 Edge Direction Histogram 17

3.2.4 Gist Descriptor 18

3.2.5 Euclidean Distance 19

3.2.6 Histogram Intersection Distance 19

3.2.7 Conclusion 20

3.3 Future Improvement 21

4 Course Feedback 21

Reference: 23

# My Personal views

EECS332 Digital Image Analysis is a class of teaching image processing on digital images. It introduces a basic understanding of fundamentals and applications of processing digital images.

I learned a lot of basic algorithms for image process, such as, edge detector, morphological operations, color segmentation and so on. I implemented these techniques through C++ with the Opencv library. Although these techniques has already been implemented and been optimized greatly, we still implement in our own way. Definitely, my algorithm maybe not efficient enough and sometimes it performs not very well, but I got profound understanding of these techniques. Taking the canny edge detection as an example, I implement the progress step by step according to the lecture. Firstly, conducting Gaussian smoothing, then, calculating the gradient of the image… Finally, I made it. I was so excited. The canny edge detection in the image processing is not easy, it gave me a lot of confidence in the image-processing field.

I saw one project, which could recognize users’ gesture in my undergraduate study. I was shocked at that time. I was always wondering how the system could recognize and analyze it? Due to my school had no such class of image analysis for undergraduate students, so I did not step into that fields. However, after I saw this Digital Image Analysis class, I choose this class decidedly. I want to see how can people make that.

Besides, there is also another significant reason that grasping this technology can expands the area of job hunting. Currently, there are a great number of corporations dealing with the image processing. In other words, image processing can be applied in numerous fields of our daily life, such as, medical, robot, transportation, and industry. If I am mastered at this technology, it can increase my opportunity of find a good job greatly.

I want to learn more techniques about object recognizing. I want to learn how to let the system recognize and analyze objects. The reason is that I am interested in the robot area. As a result, I want to apply such technology on the robots. So that, robots can recognize the scenery around it, and analyze it by itself while they are moving. Robots can look around and analyze which area he can go. For a simple example, if there is a stone on the way, robot can recognize it and bypass it. I believe that is an interesting field, and if robots have that ability we can let such robots to do many works, which we human cannot do. For a daily and small example, we can let such robots clean the floor, which can make out life easier and more relax. What’s more, we also can send these robots to other planet to conduct some scientific research. I believe this is a big developing direction in the future.

# ImageGoogle

## Introduction

In the past, there are a great number of image retrieval systems the Blobworld System developed at U.C. Berkeley [1] and the IBM QBIC system developed at the IBM Almaden Research Center. Inspired with the Google Image search, which can query image through one key word, our group (Feng liu, Xiaofeng Zhu, and me) believe query images through one image will play a significant role in the future. As a consequence, we choose the ImageGoogle as our final project. We use Java to implement the user interface of our system and choose Matlab as the backend to do the image processing.

As a matter of face, the fundamental technology of these systems is that they extract the signature of each image and establish a rule for comparing images. By according to this fundamental theory, we employ the content-based image retrieval (CBIR) as the main idea to implement our ImageGoogle system. CBIR is the application of [computer vision](http://en.wikipedia.org/wiki/Computer_vision) techniques to the [image retrieval](http://en.wikipedia.org/wiki/Image_retrieval) problem, that is, the problem of searching for [digital images](http://en.wikipedia.org/wiki/Digital_image) in large [databases](http://en.wikipedia.org/wiki/Database) [2]. There are three most common characteristics for CBIR is color, shape and texture. In this project we mainly focus on the color and shape comparison between images.

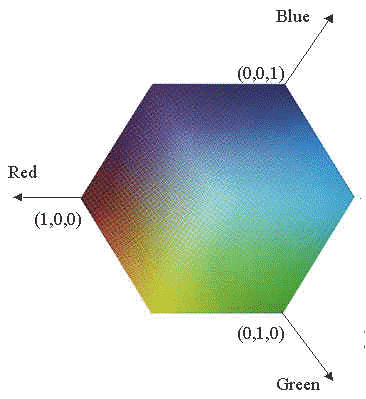
For the color character, we using global color histogram and block-based color histogram to extract the color information of each image. For the shape character, we extract the edge direction histogram of every image as the measurement. Besides, we also employ the gist descriptor to compare images. In the following passage, we will introduce the implementation and compare the outcome of these techniques. After extract the characters of images, we need to calculate the similarity according to these feature. So we decided to use Euclidean distance and histogram intersection distance to calculate the similarity between images.

## Color-based retrieval

There are three basic methods for retrieving color information from images: histogram, color layout and region-based search. In this project, we only focus on the histogram in the RGB space to extract the color feature from images. 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 space. A color component, or a color channel, is one of the dimensions. And a histogram search characterizes an image by its color distribution.

### RGB Color Model

The RGB color model consists by three primary colors: Red, Green and Blue. The choice of primary colors is related to the physiology of the [human eye](http://en.wikipedia.org/wiki/Human_eye); good primaries are stimuli that maximize the difference between the responses of the [cone cells](http://en.wikipedia.org/wiki/Cone_cells) of the human retina to light of different wavelengths, and that thereby make a large [color triangle](http://en.wikipedia.org/wiki/Color_triangle) [3].



1. RGB color model

### Color Histogram

A color histogram represents the distribution of colors in an image, because it counts the number of pixels of each color in one image. The color histogram is defined by,



where A , B and C represent the three color channels (R,G,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 [4].

In this project, we use two color histograms, one is global histogram and the other is block-based color histogram. It is obvious that global histogram means construct one histogram of the whole image. And block-based histogram means we divides one whole image into several parts and calculate the histogram of each parts respectively.

### Implementation

As we all know, there are three channels in the RGB space, and every channel contains 256 colors. As a result, the color histogram has 256\* 256\*256 = 16777216 colors. It is impossible for us to calculate. So we decided to quantize all colors into 64\*64\*64 = 262144 colors. To begin with, we divide the image into three channels and traverse all pixels one by one and determine the range of each pixel locates. Then, we calculate the total number pixels of each range. Finally, we merge the three channel histograms into one color histogram.

For the block-based color histogram, we just divide one image into four parts equally and calculate the color histogram respectively.

## Edge-based retrieval

Considering the drawbacks of simply using color histogram to query images, we also implement the edge direction histogram. Edge in images is an important low-level feature. It indicates the shape of the object in images. It also can describe the texture of the image.

### Edge Histogram

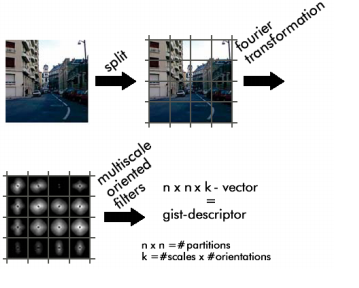
Similar to the color histogram, the edge histogram represents the edge distribution in one image. The edge histogram focuses on the shape of the object instead of the color. As a result, the edge histogram can help to find images, which contain similar objects with the query image.

### Implementation

We employ the canny detector to extract the edge histogram. The procedure is similar to the canny edge detection. First of all, we convert the color image into gray scale. Then use Gaussian filter to smooth the gray scale image. After that we calculate the gradient of it. Then we count every pixel to build the histogram of the direction range. Finally, we quantized that histogram.

## Gist Descriptor

Gist descriptor is created for recognizing similar scenes, like mountains, tall buildings, streets and so on. It is a structure without color information. By using the gist descriptor, we can extract the gist of each image easily, then, we can compare the gist of each image to determine the similarity of them.



(b) Gist Descriptor Procedure

### Implementation

The idea of the gist descriptor is easy. It will divide the image the image into several parts and anchor the structure of the sub images to their location in the image. After that, we use the fourier transformation and multiscale oriented filters to get the gist descriptor. We just use the build-in function provided by the Matlab called “LMgist” to compute the gist descriptor. Before that we need to determine the parameter of this function. After getting the gist descriptor of one image, we need to visualize it. So we use the “showGist()” function to show the gist descriptor of the images.

## Similarity Calculation

After extracting features of images, we need to figure out a rule to measure the similarity of image based on these features. Also there are some distance formulas for measure the similarity of color and edge histograms. In this project, we only focus on the following two distance calculations: Euclidean distance and Histogram intersection distance.

### Euclidean Distance

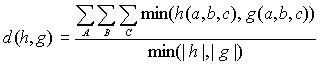
In [mathematics](http://en.wikipedia.org/wiki/Mathematics), the Euclidean distance or Euclidean metric is the "ordinary" [distance](http://en.wikipedia.org/wiki/Distance) between two points that one would measure with a ruler, and is given by the[Pythagorean formula](http://en.wikipedia.org/wiki/Pythagorean_theorem) [5]. We assume that h and g represent two color histograms. The Euclidean distance between the color histograms h and g can be computed as:



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 cross-wise. All bins contribute equally to the distance.

### Histogram Intersection Distance

The color histogram intersection was proposed for color image retrieval by [6]. The formula of the histogram intersection distance is as following:



where |h| and |g| gives the magnitude of each histogram, which is equal to the number of samples. Colors not present in the user's query image do not contribute to the intersection distance. This reduces the contribution of background colors. The sum is normalized by the histogram with fewest samples.

## Image Database

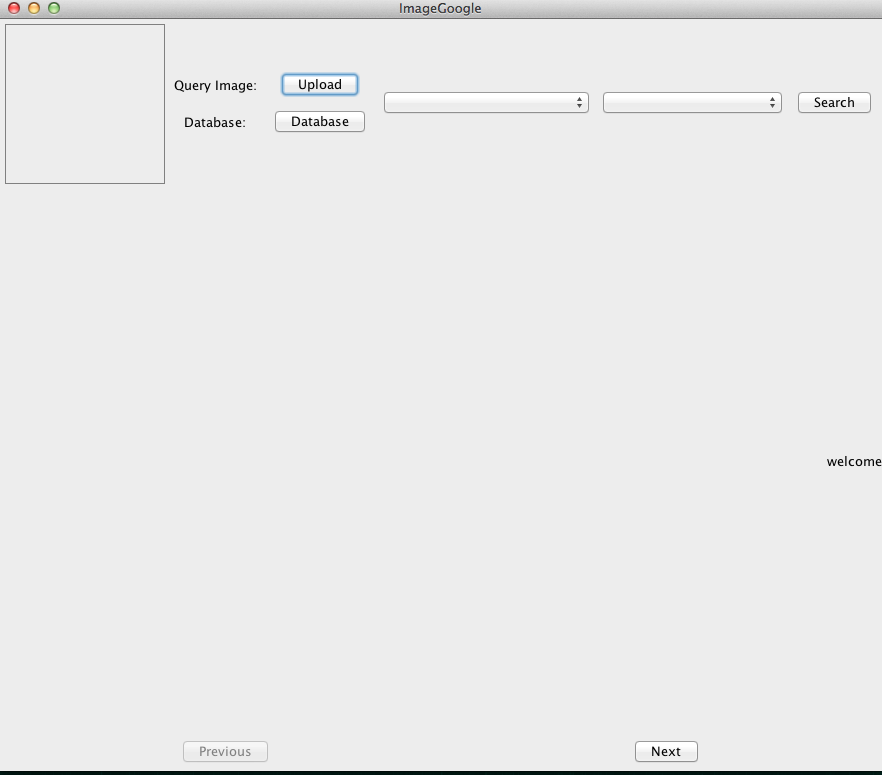
For our image database, we download 1000 sample images from the Internet [7] which is used to test the performance of the CBIR system. These images contain many categories, such as, people, cars, dinosaur, and landscape.

As a matter of fact, we do not establish a real database, such as Mysql, to store these images. We just store these images in one folder. Due to we employ four image feature-extracting methods: global color histogram, block-based color histogram, edge direction histogram and the gist descriptor. We use our own Matlab methods to extract the color and edge feature of every image in the database and store these features separately into mat files. As a result, our system does not need to extract features from images in the database every time when the user query images in our system. In other words, we just load the corresponding mat file and compare the feature extracted from the query image. This method will increase the speed of our system dramatically.

# Results

## ImageGoogle

This is the screenshot of our ImageGoogle system.

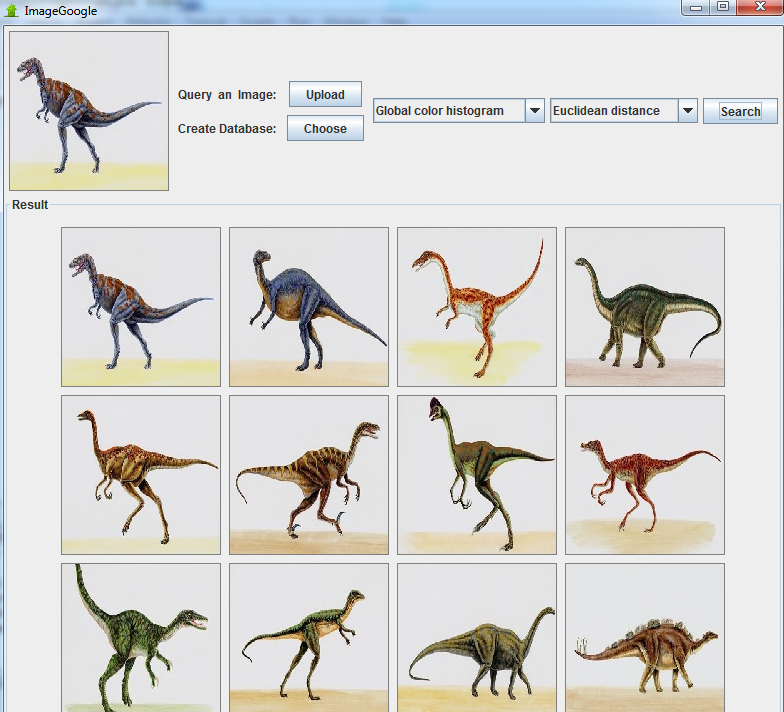


The user can click the upload button to upload a query image. Then, he needs to choose one method among global color histogram, block-based color histogram, edge direction histogram and gist descriptor to extract one specific query feature. Also he needs to choose one method he wants to use to calculate the similarity from Euclidean distance and histogram intersection distance. After pressing the search button, the system will provide 24 similar images for the user. Besides, users also can choose their own image database. They can click the database button to choose one file as the image database.

## Query Results

### Global Color Histogram

The advantage of using global color histogram is that the object in the image you want to query would not the affected if the object is translated or rotated in other images.



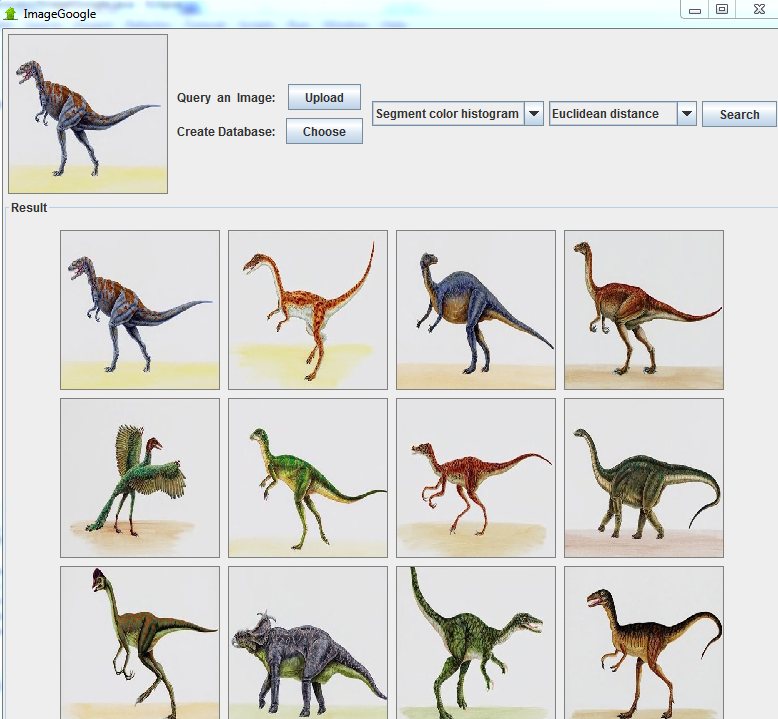
According to the above screenshot of the result using color histogram, we can query similar images, which are rotated. We can see that the dinosaur in the query image faces to left. Using this method, we can get similar images whose dinosaurs face to the right. What’s more, in most cases, colors often represent the content of images. In other words, color histogram can solve most common query images.

However, the drawback of this method is that global color histogram only focus on the color distribution of images, and it ignores the object’s location, shape and texture. As a result, this method may potentially be identical for two images with totally different object content which happens to share the same color information.

### Block-based Color Histogram

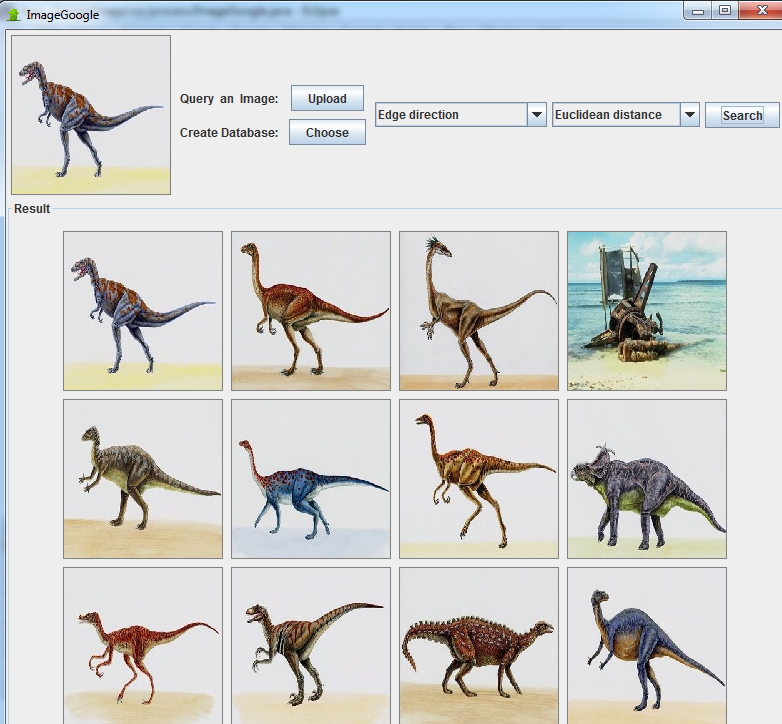
To improve the problem mentioned above, we implement the block-based color histogram. Obviously, the block-based color histogram has better performance then the global color histogram.

But the drawback is that it also increases the calculation in the meantime.



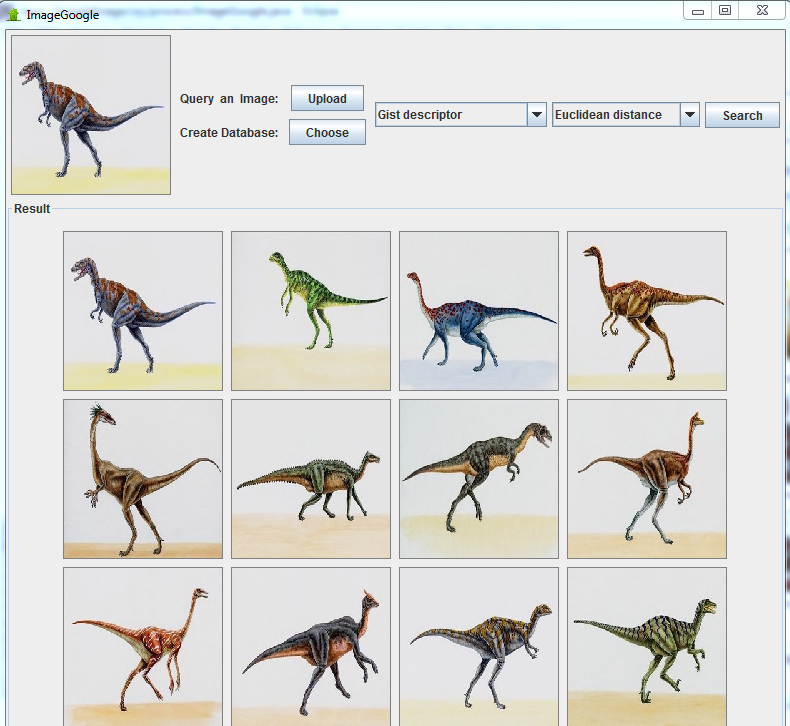
### Edge Direction Histogram

The edge direction histogram only pays attention to the shape of the objects in images. Sometimes, it will also take the texture of images into consideration. As we can see, if we use the edge direction, sizes of similar images are nearly as same as the query image. And they face to the same direction. But colors of these dinosaurs are far different from the query image.



### Gist Descriptor

Gist Descriptor is a structure without color information. It will record the gist of the image. It is easy to store, easy to handle and fast. The drawback is that it is difficult to implement. We can see that the gist descriptor mainly focuses on the background of images. All similar images and the query image have nearly same background color.

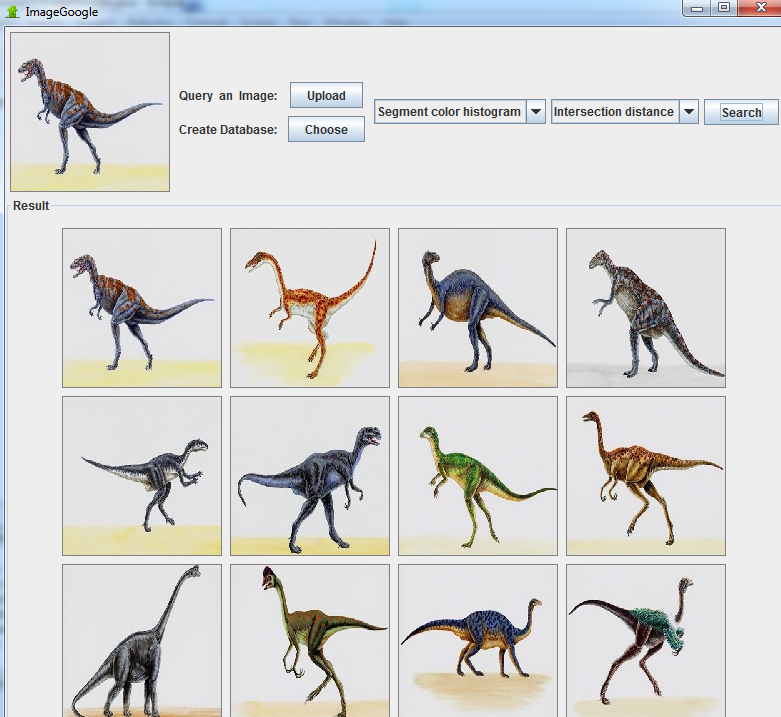
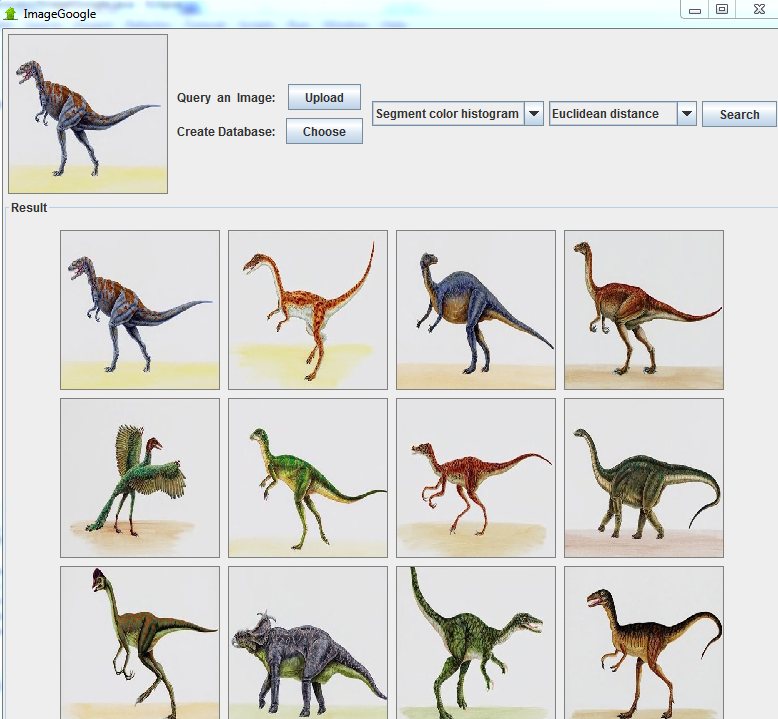


### Euclidean Distance

It compares all identical bins in respective images. In other words, all bins contribute equally to the distance. The scope of the Euclidean distance result is from 0 to 1. If the result nears the 0, then it means that image is similar to the query image. If the result is nearing 1, then it indicates that image is not similar with the query image.

### Histogram Intersection Distance

In this method, colors not present in the user's query image do not contribute to the intersection distance. This reduces the contribution of background colors. The sum is normalized by the histogram with fewest samples. The result is opposite to the previous Euclidean distance. The result scope of this method is from 0 to 1,too. But if the result nears to 1, then it means these two images are very similar. If the result nears to 0, then it indicate these two images are not similar.



Comparing the outcome of one same feature extracting methods with tow different similarity calculation methods. We can see that the histogram intersection distance performs better than the Euclidean distance.

### Conclusion

After compare result of all methods, we can figure out a conclusion that, if the query image has single and outstanding object, like the dinosaur queried in this report, then using block-based color histogram combines with histogram intersection distance is a better choice. If the query image contains complicated and colorful objects, like the landscape, then the gist descriptor or edge direction histogram is a better.

## Future Improvement

In the current version, we just separate the color-based retrieval methods and edge-based retrieval method completely. As a result, the ImageGoogle system can only focus on one specific feature of images so the result may not good enough. In the future, we can combine these query methods together. So that our system will focus on both color and shape of objects in images. Definitely, it will increase the performance.

Besides, currently, our database only contains 1000 pictures, and when the user search one query image among in the database. The system still needs to take a bit of time to search. I think we need to make an index of the database, so that it can increase the speed dramatically.

# Course Feedback

It is a fundamental course of image processing. It is suitable for students who have no experience in that area before. However, this course provides a lot of linear algebra and probability. To be honestly, I am not good at mathematics. So sometimes, I would feel boring of the class. I think professor can make the teaching more vividly and interesting. What’s more, professor can introduce us more about some interesting related projects to us. I believe it will attract students’ eyes.

Reference:

1. James Z. Wang, "Integrated Region-Based Image Retrieval", Boston, Kluwer Academic Publishers, 2001

1. [Content-based Multimedia Information Retrieval: State of the Art and Challenges](http://www.liacs.nl/home/mlew/mir.survey16b.pdf), [Michael Lew](http://en.wikipedia.org/wiki/Michael_Lew), et al., [ACM Transactions on Multimedia Computing, Communications, and Applications](http://en.wikipedia.org/wiki/ACM_Transactions_on_Multimedia_Computing,_Communications,_and_Applications), pp. 1–19, 2006.
2. R. W. G. Hunt (2004). The Reproduction of Colour (6th ed.). Chichester UK: Wiley–IS&T Series in Imaging Science and Technology. [ISBN](http://en.wikipedia.org/wiki/International_Standard_Book_Number) [0-470-02425-9](http://en.wikipedia.org/wiki/Special:BookSources/0-470-02425-9).

1. <http://scien.stanford.edu/pages/labsite/2002/psych221/projects/02/sojeong/#dct>

1. <http://en.wikipedia.org/wiki/Euclidean_distance>
2. M. J. Swain and D. H. Ballard. "Color indexing", International Journal of Computer Vision, 7:1 1991.
3. http://wang.ist.psu.edu/docs/related/