Indexing Images for Content Based Retrieval

Tianyi Zhuang
2014012056
THSS 41
zty0826@gmail.com

Songzhou Yang
2014013422
THSS 41
yangsongzhou.tsinghua@gmail.com

ABSTRACT
Image retrieval is one of the most important research scopes in recent days on computer science. In this paper, we compare different methods of image retrieval with different feature extraction methods, and evaluate the efficiency of r-tree.

Keywords
Image retrieval, feature extraction, r-tree, image histogram, HSV

1. IMAGES
We use two different color models, RGB and HSV. The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The HSV color model stands for hue, saturation, and value, and is also often called HSB (B for brightness). Compared with the RGB model, the HSV model may be more effective.

2. FEATURES
We use the color histogram with different implementation. And this part use MATLAB as a tool.

2.1 RGB histogram
This is a naive implementation of the color histogram, we calculate color histograms of RGB without any preprocess. And finally we get a vector of 768-dimension of every image.

2.2 HSV histogram
Consider that the HSV is more ocular than the RGB, so firstly we convert RGB images to HSV images before calculate histograms. The result is also a vector of 768-dimension of every image.

2.3 Gray histogram
In order to diminish the dimension of feature, we calculate a gray image for every image. With this process, we will get a vector of 256-dimension of every image.

2.4 Advanced RGB histogram
Even a 256-dimension vector seems to be of high dimension, so we try to diminish deeply. We re-divide the RGB model for 6Rs, 2Gs and 2Bs, and calculate histogram again. Finally, we get a 10-dimension vector and a 20-dimension vector.

2.5 Advanced HSV histogram
RGB works not as well as what we expected, so we do the same operation in HSV images. And got another sets of 10-dimension vector and 20-dimension vector.

In order to get a better results, we normalize every vector and get another sets of features.

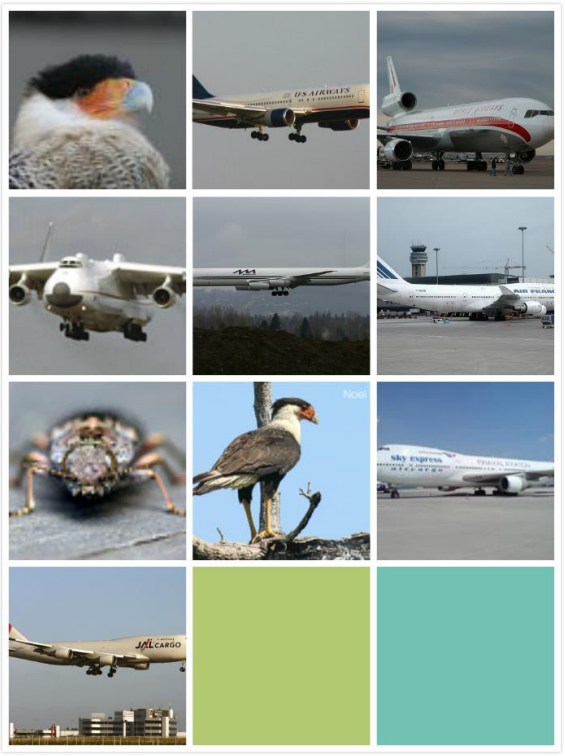
Table 1. Accuracy of different features

Features	Dimension	Accuracy
RGB Hist 10d	10	23.9106%
HSV Hist 10d	10	33.4759%
RGB Hist 20d	20	23.9337%
HSV Hist 20d	20	35.8952%
Standardized RGB Hist	10	24.0477%
Standardized HSV Hist	10	33.2924%
Standardized RGB Hist	20	24.0834%
Standardized HSV Hist	20	35.9202%
Color Moment	9	26.2890%

From this table we can see that our own HSV features can reach a higher accuracy compared with TA's. Because HSV model can show image more ocular than color moment and RGB model.

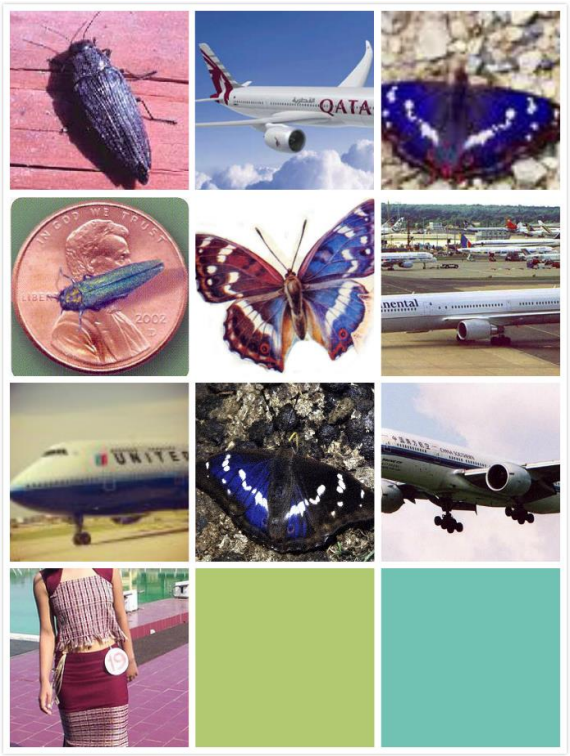
3. EXPERIMENT RESULTS
We chose different features, calculate Euler distance between the query feature and stored features and return the nearest 10 results. These are some of our results.

3.1 Using Standardized 10-dimension RGB Hist-vector

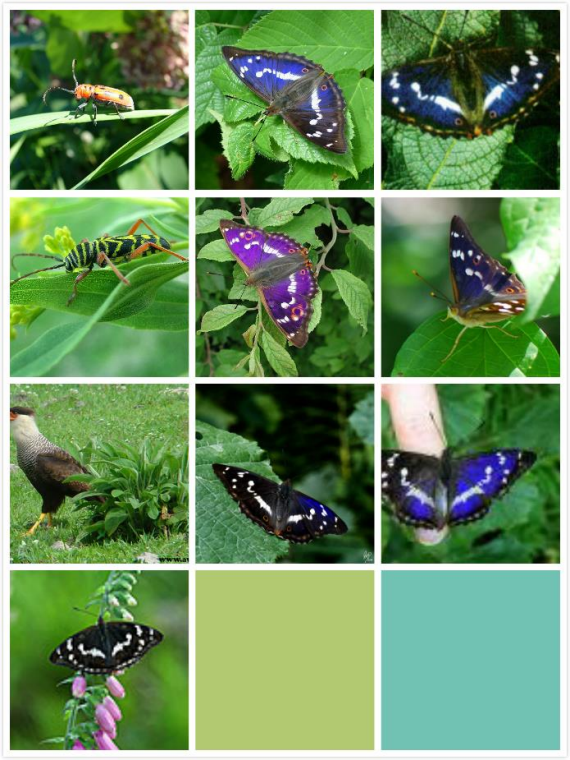


3.2 Using Standardized 10-dimension HSV Hist-vector

3.3 Using Standardized 20-dimension RGB Hist-vector



3.4 Using Standardized 20-dimension HSV Hist-vector



4. THE EFFICIENCY OF R-TREE

Because data in this experiment is low-scale, we store data in RAM rather than in disk. In this satiation, we use the node access times to imitate the disk access times.

We test with different parameters about the max number of data stored in one node, the number of data and the dimension of data.

In order to evaluate r-tree entirely, we generate some new features with different dimensions.

Here are results.

Table 2. Efficiency of R-Tree

The max number of data stored in one node: 4

	1000	2000	3000	4000	5000
4	47936	87915	121693	166713	195338
8	51577	90046	126545	159694	198996
12	63514	118011	160204	227214	266957
16	53704	106543	148847	185232	217470
20	63388	119121	171383	205733	271546

The max number of data stored in one node: 8

	1000	2000	3000	4000	5000
4	30419	55734	81127	99879	128094
8	31692	59636	77159	97291	119555
12	43640	77800	111082	134741	160101
16	41510	77214	106083	137250	175426
20	41098	76838	109882	141383	181638

The max number of data stored in one node: 12

	1000	2000	3000	4000	5000
4	25520	41892	54439	71425	84177
8	24009	41132	59130	77239	87362
12	29267	58100	75902	98288	130296
16	32104	61472	90132	114418	139251
20	32829	65151	96135	119793	150430

From above tables, we will know that the node access times are concerned with the max number of data stored in one node, the number of data and the dimension of data. The more the number of data stored, the fewer the number of data is, so the node access times is fewer. The more the dimension of data is, the more complex the query space is, so the node access times are more.

5. CONCLUSIONS

The efficiency of r-tree is concerned with many factors, we can take different methods to enhance r-tree on the basis of different usages. In general, the more dimension the data has, the bad efficiency the r-tree has, so there is a large optimization space. Aimed at the image retrieval problem, the accuracy seems to be lower, so it's important to modify feature extraction methods even rebuild a framework.

6. REFERENCES

[1] Y Rui, TS Huang, SF Chang. 1999 Image Retrieval: Current Techniques, Promising Directions, and Open Issues. Journal of Visual Communication and Image Representation Volume 10, Issue 1, 1 March 1999, Pages 39–62

[2] Anil K. Jain, Aditya Vailaya. 1996 Image retrieval using color and shape. Pattern Recognition Volume 29, Issue 8, August 1996, Pages 1233–1244

[3] A. W. M. Smeulders ; Amsterdam Univ., Netherlands ; M. Worring ; S. Santini ; A. Gupta. 2002 Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume:22 , Issue: 12)