Related Image Retrieval using Color and Gabor Filter Histogram Featurization

Dandre Allison
Computer Science Department
Pomona College
185 E. Sixth Street
Claremont, CA 91711
dra02007@mymail.pomona.edu

Martin Wintz
Computer Science Department
Pomona College
185 E. Sixth Street
Claremont, CA 91711
mw002007@pomona.edu

ABSTRACT

Content-based image retrieval (CBIR) is one of the most researched problems in Information Retreival. Rather than using metadata, CBIR uses computer vision techniques to extract information directly from the image file in order to computer similarity with other images. In this paper we explain the steps involved in a fully functional image retrieval system based on color and texture features. We examine feature extraction methods, such as color histograms and Gabor wavelet histograms. We also explain our method of index construction, as well as multiple dissimilarity measures. We then discuss methods of generating a "ground truth" corpus, and test the performance of our system using various parameters on this corpus.

1. INTRODUCTION

CBIR is significantly more difficult than text retreival, due to the form of the data used in the task. In text retreival, words provide an obvious feature with which to classify documents by similarity. Since human languages have finite dictionaries, small enough for modern computers to process, one can create postings-lists for each word which hold information about which documents the word appears in. In image retreival, the data comes in form of a two-dimensional matrix which contains information for each pixel. Due to the continuous nature of the data, it becomes harder to extract individual features, and due to the mathematical nature of the features, it becomes more or less impossible to classify them into postings-lists.

The first step in image retrieval is therefore feature extraction. Feature extraction is the process of extracting meaningful data – in the sense that it can actually describe something about an image which would be helpful in image retreival – into a form which is helpful in quickly comparing images. Usually, these features will correspond to qualities of an image that correspond to human vision, such as color, texture, etc... Common featurization methods break down elements like color and texture into histograms which are then used to computer some sort of similarity measure. For our image retreival system, we use La*b* color histograms and Gabor wavelet filters. Color histograms are the usual tool for color comparision, and we convert the standard RBG values to La*b* values, which more closely represent human perception of color. Gabor wavelet fitlering is a popular method for texture extraction, which involves a Gabor filter: a linear filter which is the product of a Gaussian function and a complex sinusoid.

Once the features are extracted into histograms, the next task is constructing an index so that images can be quickly retreived. Since postings-lists cannot be used because of the nature of histogram features, each image needs to be compared to each other image before search in order to keep retrieval time down. We compute a two-dimensional matrix of similarity measures using different histogram-dissimilarity measures, such as L1 measure, etc...

2. RELATED WORK

Since image retrieval is such a well researched field, part of the challenge in finding useful related works is parsing through a large body of literature. We based the structure of our system on a first paper which briefly goes through every step of the image retrieval process[3]. Although the paper's main goal is to examine the performance of different dissimilarity measures, it briefly describes color histograms and gabor wavelet. All our histogram dissimilarity measures are drawn from this paper.

Since gabor filters require complex math, we used mutliple sources in order to implement them. We used a tutorial on gabor filters and an online demo[2] in addition to a paper on gabor filter extraction[1] and an example implementation from open-source computer vision code.

3. METHOD

3.1 Featurization

As previously mentioned, most important and difficult problem in image retrieval is feature extraction. We discuss two popular methods of feature extraction for color and texture features respectively. The first, color histogram extraction, involves dividing up the color information of a picture into a set amount of bins. The second is Gabor wavelet feature extraction, which uses Gabor wavelets filters of different scale and rotation to extract texture information into a histogram based on filter transformations.

3.2 Color Histogram

Creating a color histogram consists of collecting information from each pixel of the image and quantizing it into a set amount of bins. First, each pixel's coordinates in the color space are extracted. Typically this information will come in the RBG colorspace, however, the La*b* color space was designed to more accurately represent differences in color according to human perception, where the L is the lightness component and a* and b* describe the color. The non-linear conversion from RGB to L a*b*is described by the equation:

$$
\begin{aligned}
L &= 116f(Y/Y_n) - 16 \\
a* &= 500[f(X/X_n) - f(Y/Y_n)] \\
b* &= 200[f(Y/Y_n) - f(Z/Z_n)]
\end{aligned}
$$

where

$$
f(t) = \begin{cases} t^{1/3} & t > (6/29)^3 \\ \frac{1}{3}(\frac{29}{6})^2 t + \frac{4}{29} & \text{otherwise} \end{cases}
$$

This information is then evenly divided into a set number of bins for fast comparison. For our image retrieval system, we used 256 bins.

3.3 Gabor Wavelet Histogram

A Gabor filter is the result of the multiplication of a complex sinusoid and a Gaussian function:

$$
g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) cos\left(2\pi \frac{x'}{\lambda} + \psi\right)
$$

where

$$
\begin{aligned}
x' &= xcos\theta + ysin\theta \\
&\text{and} \\
y' &= -xsin\theta + ycos\theta
\end{aligned}
$$

The five parameters are the wavelength of the cosine factor (λ) which affects the scale of the wavelength, the orientation (θ), the phase offset (ψ), the sigma of the Gaussian envelope (σ), and the spatial aspect ratio (γ), which affects how wide the wavelet is. Based on this function, we can create a dictionary of Gabor filters to test for responses against a picture. We generate a dictionary by modulating over λ and θ to obtain filters of different size (m different sizes) and orientation (n different orientations), and then perform the Gabor transform on the image $I(x, y)$ using each filter which is defined by

$$
w_{mn}(x, y) = \int (I(x_1, y_1) g_{mn} * (x - x_1, y - y_1) \, dx_1 dy_1
$$

We then create a histogram of size $m*n*2$ with each bin alternating between representing the mean μ_{mn} and the standard deviation σ_{mn} defined as:

$$
\begin{aligned}
\mu_{mn} &= \int\int |w_{mn}(xy)| \, dxdy, \\
&\text{and} \\
\sigma_{mn} &= \sqrt{\int\int (|w_{mn}(x, y)| - \mu_{mn})^2 \, dxdy}
\end{aligned}
$$

For our image retrieval system, we used 5 different scales over 6 different orientations.

3.4 Histogram Dissimilarity

Once collected for all the pictures, the histograms serve as a point of comparison for dissimilarity. Different dissimilarity measures have different advantages depending on the image retrieval task – for example, color vs. texture. Currently we have implemented *L1* dissimilarity measure, which is defined as the sum of the absolute value of the difference between two bins in a histogram over all bins.

4. EVALUATION

Our evaluation techniques were directly drawn from one of the papers we studied.

4.1 Color Evaluation

For color we used a Landscape database of 80 images. The premise of this evaluation is that randomly sample pixels from an image should, upon query time, get their original images as well as other other randomly sampled pixel sets from the same image. back as a result. To test this, we create disjoint randomly generated pictures from pictures, by sampling random pixels from that picture. There were 16 disjoint pictures per set size, generated from each image. The sizes where 4, 8, 16, 32, and 64, making a total of 80 query images for each original image. The small sizes of the samples ensure that our system can retrieve images that are only moderately similar.

4.2 Texture Evaluation

Our method of texture evaluation was similar. We used the texture database of 58 textures. The premise of this evaluation is that sub-images of the full texture image should, upon query time, get their original image back as a result. Unlike in color evaluation, the sampled pixels are adjacent rather than randomly sampled from the image. We generated a corpus of smaller images, sampling from the original picture. There were 16 pictures per set size, generated for each image. The sizes where 4, 8, 16, 32, and 64, making a total of 80 query images for each original image.

5. RESULTS

Although on individual sources, it was visible that our system performed well, our evaluation methods yielded some strange results

For color evaluation, we measured precision at 10 and 20. We tried two methods: using the original image as a query, and using one of the extracted pixel sets as a query. Because our quantizing function for color histograms was not well adapted to La*b*, and was generally not a very "smart" quantization method, results were heavily affected by the size of the image. When searching on one of the large originals, precision at 10 was about 2%, and precision at 20 was around 4%. However, when searching on one of the samples, other samples from the same image would always be returned, and precision at both 10 & 20 was a full 100%

For texture evaluation, we similarely measured precision at 10 and 20, recieving extremely positive results. Precision at both 10 was about 94%, and precision at 20 was about 91%.

6. CONCLUSION

In conclusion, although our system worked reasonably well for images which were close in size, we were limited by time to implement better evaluation methods. Evaluation in CBIR is an extremely difficult task, since collecting a ground truth of data involves either long hours of manual labor, or some sort of cleverly auto-generated ground truth. We used techniques described in one our the papers we studied, but they might have been better fit to the paper's purpose (comparing dissimilarity measures) than to measure the performance of our system on its own, where they were clearly biased for our data. Our normalization method also needs to be improved, but we weren't able to find anything specifically about quantizing La*b* values.

7. REFERENCES

[1] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.

[2] N. Petkov. Gabor filter for image processing and computer vision. http://matlabserver.cs.rug.nl/edgedetectionweb/web/edgedetection_params.html, 2008.

[3] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *Comput. Vis. Image Underst.*, 84(1):25–43, 2001.