Image Categorization Using Texture Features

Aya Soffer *
Computer Science and EE Department
University of Maryland Baltimore County
5401 Wilkens Avenue
Baltimore, MD 21228-5398 and
CESDIS, Goddard Space Flight Center
E-mail: soffer@cs.umbc.edu

Abstract

A method for categorizing images using texture features is presented. The goal of categorization is to find all images from the same category as a given query image. Some example categories are floor plans, music notes, satellite images, houses, and fingerprints. The hypothesis that two images that have the same texture features are likely to belong to the same category, is examined. A new texture feature termed $N \times M$*-gram* is presented in detail. It is based on N-grams , a technique that is commonly used for determining similarity of text documents. Intuitively an $N \times M$-gram is a small pattern in an image. The notion of $N \times M$-grams is defined and the process of computing an image profile in terms of its $N \times M$-grams is explained. Three similarity measures for comparing images based on their $N \times M$-gram profiles are proposed. Results of an experiment on images from various categories are presented. The three similarity measures for $N \times M$-grams are compared to each other as well as to results of categorization using other well known texture features such as cooccurrence matrices, local standard deviation, Laws texture features, and grey level distribution features. The results show that for our test images $N \times M$-gram based methods were more successful in finding images from the same category as a given query image than other texture features.

Kewords: retrieval by content, image categorization, texture, N-grams , image similarity, image databases, document analysis

1 Introduction

With the proliferation of systems that store electronic images, there has recently been much interest in similarity based image retrieval [9]. The goal is to retrieve all images that are similar to a given query image. The type of similarity varies from system to system. The most common form of similarity is based on color [18, 19]. The underlying assumption is that often colors are used to

*The support of USRA/CESDIS and NASA Goddard Space Flight Center is gratefully acknowledged.

encode functionality, and thus images with similar color distributions have similar themes. Another form of similarity is based on texture [5, 13]. Shape based similarity is also used in some systems [6]. In QBIC [14], users must outline objects in the image in order to use shape features, thus this method cannot be used non-interactively. In some cases when the domain of images is limited automated indexing based on shape is also possible [17].

Testing whether a particular method for image similarity is useful has proven to be an elusive task. One of the difficulties is that in many cases it is not clear even to the evaluator which images are most similar to a given query image. In our work, we have thus chosen to define the goal of similarity retrieval in a slightly different manner. Rather than trying to find the images that are most similar to a given query image, we assume that the images in our database can be divided into well defined categories, and that the goal of retrieval is to find other images from the same category as the query image. Some example categories are floor plans, music notes, satellite images, houses, and fingerprints. With this goal in mind, it is easier to assess how well the system performs. While categorization may not be relevant for all image archives, we believe that for large image archives such as the world wide web, having the ability to find for example pictures of houses is useful. Furthermore, once all images from the same category are retrieved, a domain specific technique tailored for the specific category can be used to refine the results.

In this paper we explore whether images can be categorized based on texture features. We hypothesize that images from similar categories have similar texture features. In particular, we present a method for categorizing images using a new texture feature termed $N \times M$*-gram* . This method is based on the N-gram technique that is used for determining similarity of text documents [2]. Our approach for categorizing images is based on extending the definition of N-grams to 2D. Intuitively an $N \times M$-gram is a small patch or pattern in an image. The hypothesis that we examine in this paper is that two images that have the same recurring patterns are likely to belong to the same category. We define the notion of $N \times M$-grams and show how to compute an image profile in terms of its $N \times M$-grams termed an $N \times M$*-gram vector*. We propose three similarity measures to compare images based on their $N \times M$-gram vectors. We present results of an experiment that we conducted on images from various categories. We compare the results using $N \times M$-grams with these similarity measures to each other as well as to results of categorization using other well known texture features such as cooccurrence matrices [7], local standard deviation, Laws texture features [11], and to a method based on color distribution features [18]. Our results show that for our test images $N \times M$-gram based methods were more successful in finding images from the same category as a given test image than other texture features or color distribution based methods.

The rest of this paper is organized as follows. In Section 2 we review the definition of N-grams and define $N \times M$-grams. Related work is also described in this section. Our method for generating $N \times M$-gram profiles of images is presented in Section 3. Several similarity measures for comparing two images based on their $N \times M$-gram profiles are given in Section 4. In Section 5 we describe a method for categorizing images based on grey level distribution features as well as methods based on other texture features. Our evaluation procedure and experimental results are discussed in Section 6. Section 7 contains concluding remarks.

2 Background and Definitions

An *N-gram* is a sequence of N consecutive characters in a document. N-grams are generated from a document by sliding a "window" N characters wide across the document's text, moving it 1 character at a time. The word "image", for example, consists of three tri-grams "ima", "mag", and "age". The *N-gram frequency profile* of a document consists of a frequency count of each possible N-gram in the language. For example, for the English case-insensitive alphabet, there are 26^3 possible tri-grams. Thus, the N-gram frequency profile of a document written in English would be the frequency of each of these 26^3 possible tri-grams in the document. The N-gram frequency profile can also be viewed as a vector of dimension 26^3 termed an *N-gram vector*. A similarity measure between the N-gram vector of two documents determines the similarity of these two documents. The assumption is that two documents whose N-gram vectors are "close" are likely to deal with related matter. One measure that is commonly used is the normalized dot product of the two N-gram vectors.

0	0	1	1
0	1	1	0
1	1	0	0
1	0	0	0

Figure 1: Example image pixel values. The corresponding 3×3-grams are "001011110", "011110100", "011110100", and "110100000".

Our approach for categorizing images is based on extending the definition of N-grams to 2D. An *$N \times M$-gram* is a sequence of $N \times M$ pixel values. $N \times M$-grams are generated from an image by sliding a "window" $N \times M$ characters wide across the image moving it 1 pixel at a time in a row-wise manner. For example, the image whose pixel values are shown in Figure 1 consists of four 3×3-grams, "001011110", "011110100", "011110100", and "110100000". The *$N \times M$-gram frequency profile* of an image consists of a frequency count of each possible $N \times M$-gram. For example, for binary images there are $2^{3\times3} = 512$ possible 3×3-grams. Thus, the 3×3-gram frequency profile of a binary image would be the frequency of each of these 512 possible 3×3-grams in the image. Note that if we consider the number of occurrences of each $N \times M$-gram rather than the frequency, then the $N \times M$-gram vector is the histogram of $N \times M$-gram values. A similarity measure between the $N \times M$-gram vector of two images determines the similarity of these two images. The assumption is that two images whose $N \times M$-gram vectors are "close" are likely to belong to the same category.

Some other systems that use N-gram techniques for indexing images include CANDID [10] which employs probability density functions in an approach that closely resembles the N-gram work for textual data. The general idea is to first compute several features (local color, texture, and/or shape) at every pixel in the image, and then compute a probability density function that describes the distribution of these features. This approach is analogous to the N-gram approach in that the image is described by a global signature and image similarity is determined by comparing these signatures. However, color and texture features are still used for the signature rather than extending N-grams to 2D. 2D pixel trigrams have been suggested for locating copies of pictures in an image database [8]. A variation of $N \times M$-grams called local binary patterns has been shown to be effective for classification of Brodatz textures [15].

3 Generating $N \times M$-gram vectors

Since the goal of using $N \times M$-grams is to implicitly index on the structure of the image and not on the color or greyscale, and since we wanted to limit the range of possible $N \times M$-grams , we first transform the images into a binary representation. Currently we perform this "binarization" using a global threshold operation on the input images. Other binarization methods such as gradient images and edges will be used in the future. The $N \times M$-grams are calculated for the binary image by sliding a window of size $N \times M$ over the entire image. The pixel values of the N rows of M pixels in this window are concatenated. The number of occurrences of each $N \times M$-gram in the image is counted. This number is divided by the total number of $N \times M$-grams in the image to compute the $N \times M$-gram frequency. We store both the absolute count and the frequency of each $N \times M$-gram for each image. In addition, we compute the average frequency and average count of each $N \times M$-gram for the entire corpus of images.

4 Comparing $N \times M$-gram vectors

The similarity measure commonly used to compare two N-gram vectors of text documents is the normalized dot product of the two vectors. We use a similar measure for images. Let T denote the highest $N \times M$-gram value (e.g., 511 for binary images and 3×3-grams), and let $f(j, I)$ denote the frequency of $N \times M$-gram j in image I. For two images A and B the similarity score $S_{dp}(A, B)$ using the dot product similarity measure is given in Equation 1.

$$S_{dp}(A,B) = \frac{\sum_{j=0}^{T} f(j,A) * f(j,B)}{\sqrt{\sum_{j=0}^{T} f(j,A)^2 * \sum_{j=0}^{T} f(j,B)^2}} \tag{1}$$

$S_{dp}(A, B)$ is essentially the cosine of the angle between the $N \times M$-gram vector of images A and B, its range of values lie between -1 and 1, with 1 given for an exact match (i.e., identical images). In work on text documents [2], it was observed that the similarity can be improved by subtracting the average N-gram vector for a corpus of documents from the N-gram vector of the query and database documents. The average N-gram vector is simply the average frequency of each N-gram over all of the documents that are stored in the database. By subtracting this average the weight of N-grams that are very common and hence not very useful in distinguishing between documents is reduced. We have adopted this idea for images. Let $af(j)$ denote the average frequency of $N \times M$-gram j over all images in the database. The similarity score $S_{dpa}(A, B)$ with this modification is given in Equation 2.

$$S_{dpa}(A,B) = \frac{\sum_{j=0}^{T}(f(j,A) - af(j)) * (f(j,B) - af(j))}{\sqrt{\sum_{j=0}^{T}(f(j,A) - af(j))^2 * \sum_{j=0}^{T}(f(j,B) - af(j))^2}} \tag{2}$$

The second similarity measure that we use takes the bin to bin difference of the $N \times M$-gram frequency vector. That is, the difference in frequency for each $N \times M$-gram value. The similarity between two images A and B using bin to bin difference $S_{bd}(A, B)$ is given by Equation 3.

$$S_{bd}(A,B) = \sum_{j=0}^{T} |f(j,A) - f(j,B)| \tag{3}$$

In order to reduce the weight of common $N \times M$-grams, we again adjust by the average. In this case, we multiply the absolute difference between the bins by $\frac{1}{af(j)}$, thus giving $N \times M$-grams with a lower overall frequency a larger weight. The similarity between two images A and B using bin to bin difference normalized by the average frequency of each $N \times M$-gram in the whole database $S_{bda}(A, B)$ is given by Equation 4.

$$S_{bda}(A, B) = \sum_{j=0}^{T} \frac{|f(j, A) - f(j, B)|}{af(j)} \tag{4}$$

The third similarity measure that we use tries only to gauge the similarity between images in terms of the number of common $N \times M$-grams. The motivation for this is the assumption that images from similar categories may have many common $N \times M$-grams while still having other areas of the image that are different. This case is penalized by the dot product similarity measure. For this measure we use the absolute count rather than the frequency of each $N \times M$-gram. The $N \times M$-gram vector is thus the histogram of $N \times M$-grams , and the similarity measure is the histogram intersection [19]. Let $t(j, I)$ denote the total number of occurrences of $N \times M$-gram j in image I. The similarity between two images A and B using this measure $S_{int}(A, B)$ termed *$N \times M$-gram vector intersection* is given by Equation 5.

$$S_{int}(A, B) = \frac{2 * \sum_{j=0}^{T} min(t(j, A), t(j, B))}{\sum_{j=0}^{T}(t(j, A) + t(j, B))} \tag{5}$$

$min(t(j, A), t(j, B))$ is the number of occurrences of $N \times M$-gram j in both images A and B. Thus, $S_{int}(A, B)$ is the number of common $N \times M$-grams in the two images normalized by the total number of $N \times M$-grams in them. In order to reduce the weight of common $N \times M$-grams, we again adjust by the average. Let $at(j)$ denote the average number of occurrences of $N \times M$-gram j in the database. The similarity measure $S_{inta}(A, B)$ with this modification is given in Equation 6.

$$S_{inta}(A, B) = \frac{2 * \sum_{j=0}^{T} \frac{min(t(j,A), t(j,B))}{at(j)}}{\sum_{j=0}^{T}(\frac{t(j,A)}{at(j)} + \frac{t(j,B)}{at(j)})} \tag{6}$$

5 Other Texture and Grey Level Distribution Features

In this section we describe some other texture and grey level features that we use for categorizing images.

5.1 Grey level features

In this method we categorize the images based on the grey level values of the input images. We use two different similarity measures. The first compares the histograms of the images using histogram intersection. Let G denote the highest grey level (i.e., all pixel values range between 0 and G), and let $n(g, I)$ denote the number of pixels with grey level g in image I. The similarity score between two images A and B using the histogram of grey levels $S_{grlh}(A, B)$ is given in Equation 7.

$$S_{grlh}(A, B) = \frac{2 * \sum_{g=0}^{G} min(n(g, A), n(g, B))}{\sum_{g=0}^{G}(n(g, A) + n(g, B))} \tag{7}$$

The second similarity score uses statistics on the grey level distribution rather than the histograms. It is based on a method that has been used for image similarity retrieval [18]. Using this method, we compute five global features that describe the grey level distribution throughout the image. Let P denote the number of pixels in the image, and let $I(p)$ denote the value of pixel p in image I. Note that although images are usually treated as 2-dimensional arrays (with K rows and L columns), and a pixel is given by its row and column (k, l), in this case since we are interested in the distribution of intensity values over all of the pixels in the image we view it as a 1-dimensional array with $P = K \times L$ pixels. We are only using grey scale images, thus $I(p)$ is the intensity of pixel p. For color images we would need to compute these features for each band.

We use the following five features, *df(f)* denotes distribution feature f.

df(1): $mean = \frac{1}{P} * \sum_{p=0}^{P-1} I(p)$

df(2): $stddev = \sqrt{\frac{1}{P-1} * \sum_{p=0}^{P-1} (I(p) - mean)^2}$

df(3): $RMS = \sqrt{\frac{1}{P} * \sum_{p=0}^{P-1} I(p)^2}$

df(4): $skew = \frac{1}{P} * \sum_{p=0}^{P-1} (\frac{I(p) - mean}{stddev})^3$

df(5): $kurtosis = \frac{1}{P} * \sum_{p=0}^{P-1} (\frac{I(p) - mean}{stddev})^4 - 3$

Each image is thus represented by a five dimensional feature vector. The similarity of two images using this method is determined by the weighted Euclidean distance between their corresponding feature vectors. Let F denote the number of features used to describe the color distribution, and let $df(f, I)$ denote the value of distribution feature f for image I. The similarity score using these statistical distribution features on the grey level values $S_{grls}(A, B)$ between two images A and B is given in Equation 8.

$$S_{grls}(A, B) = \sqrt{\sum_{f=1}^{F} w_f (df(f, A) - df(f, B))^2} \qquad (8)$$

w_f is the weighting factor of f^{th} feature of the feature vector. The weighing factor is computed so that features with a smaller variance have a larger weight as described in [1].

5.2 Local standard deviation

The local standard deviation texture feature calculates the standard deviation in a local window of size $dx \times dy$. The output of this process is an image, in which each pixel p is replaced by the standard deviation in the window of size $dx \times dy$ with p in its center. The same two similarity measures that are used to compare the grey scale levels are applied to the output of the local standard deviation operation. Let S denote the highest local standard deviation, and let *sdv(I)* denote the output of applying the local standard deviation operator to image I. The similarity score between two images A and B using the histogram of local standard deviations $S_{sdvh}(A, B)$ is given in Equation 9.

$$S_{sdvh}(A, B) = \frac{2 * \sum_{s=0}^{S} min(n(s, sdv(A)), n(s, sdv(B)))}{\sum_{s=0}^{S}(n(s, sdv(A)) + n(s, sdv(B)))} \tag{9}$$

Similarly, we compute the five distribution features over the local standard deviation values, and $S_{sdvs}(A, B)$ is the weighted Euclidean distance between these 5-dimensional feature vectors.

5.3 Grey level cooccurrence

This texture feature is based on the grey level cooccurrence or two-dimensional spatial dependency of of the grey levels for a fixed distance and/or angular spatial relationship. The size of the cooccurrence matrix is $G \times G$, where G is the number of grey levels. We compute the cooccurrence matrix in the horizontal and vertical directions with distance 1. Thus, each entry (g_1, g_2) in the cooccurrence matrix of an image I counts how many times the grey levels g_1 and g_2 appear next to each other in the horizontal and vertical direction in image I. We then compute eight features from the cooccurrence matrix (GLCM) of image I as described in [7]. These features are the contrast, the correlation, the inverse difference moment, the entropy, the variance, the cluster prominence, the cluster shade, and the diagonal moment of the GLCM. The similarity score between two images A and B using the grey level cooccurrence texture feature $S_{glcm}(A, B)$ is the weighted Euclidean distance between the 8-dimensional GLCM feature vector of A and B.

5.4 Laws texture features

This method extracts texture features from an image by convolving the original image with a spatial filter. The spatial filters are comprised of 5×5 kernels derived from center-weighted vectors defined by the LAWS texture measure [11]. The five center-weighted vectors are denoted by L5, E5, S5, W5, and R5 and each have 5 components. Each 5×5 kernel is derived from multiplying a particular combination of two of theses vectors. This results in 25 possible 5×5 kernels. For example, L5E5 is formed by multiplying the vectors, L5 with E5. It's transpose E5L5 is similarly formed by multiplying the vectors, E5 with L5. One of these kernels is more sensitive to horizontal changes in texture, while the other is sensitive to vertical changes in texture. We use the L5E5, L5S5, L5R5, E5L5, E5E5, S5L5, W5W5, and R5L5 kernels. This is a mixture of horizontal, vertical, and diagonal sensitive kernels. The original images are convolved with each one of these kernels. We then compute the mean, variance, RMS, skew, and kurtosis (the same features used by the GRL and SDV methods), over the grey level values of the convolved images. Once more the similarity between two images A and B using Laws texture features $S_{laws}(A, B)$ is the weighted Euclidean distance between these 5-dimensional feature vectors.

6 Implementation and Experimentation

At this point we have tested our method on 200 images from 22 different categories. These categories are finger prints, floor plans, comic strips, music notes, tables, mug shots (passport photo type pictures), pictures of houses, aircraft, animal statues, topographic maps, road maps, satellite images, aerial photos, Arabic documents, Chinese documents, English print (mainly Times-Roman and Helvetica font), English song lyrics (a large font with large spaces), English handwritten documents, Hebrew print without vowels (various dots and lines that appear mainly underneath the letters), Hebrew print with vowels, Hebrew script (a computer script font), Hebrew handwritten (script written by numerous people). Most of these images were downloaded from various web sites. Some of the English and Hebrew documents were scanned by us.

We computed the $N \times M$-gram profile for each image with $N = 3$ and $M = 3$ (i.e., 3×3-grams). In addition, we computed all the other texture and grey level features for each image, as described in Section 5. To test the retrieval accuracy, for each image I in our database we run the system with I as the query image, in effect ranking all of the other images with respect to I. We repeated this process using $N \times M$-grams with the following similarity measures as described in Section 4: dot product (*dp*), dot product normalized by averages (*dpa*), bin to bin difference (*bd*), bin to bin difference normalized by averages (*bda*), $N \times M$-gram vector intersection (*int*), $N \times M$-gram -vector intersection normalized by averages (*inta*). We also computed similarity scores using grey level histogram intersection (*grlh*) and grey level distribution statistics features (*grls*) as described in Section 5.1, local standard deviation histogram intersection (*sdvh*) and distribution statistics features (*sdvs*) as described in Section 5.2, grey level cooccurrence matrix features (*glcm*) as described in Section 5.3, and Laws texture features (*laws*) as described in Section 5.4.

6.1 Evaluation procedure

Retrievals based on exact match are usually evaluated using precision and recall statistics [16]. However, systems where the result of retrieval is a list of items sorted by similarity, need a more elaborate evaluation metric that reflects the positions in which the set of relevant items appear in the retrieved set. One such measure is the *normalized recall* metric [16], which measures how close to the top of the list of retrieved items does the set of relevant items appear compared to the ideal retrieval in which all N relevant objects appear in the top N positions. For the image similarity problem (i.e., find all images similar to the query image) it is necessary to first determine what is the set of all relevant items, a task which is usually not clear to the person evaluating the system. However, for the categorization problem (i.e., find all images from the same category as the query image), it is only necessary to know the list of images from each category for the test set. Since we assume that the images fall into well defined categories, this information is readily available.

In order to calculate the the normalized recall we define two measures [3]:

- $IAVRR$ - ideal average rank of relevant items. Ideally all images from the same category should be ranked at the top. Thus, if C_n is the number of images in category C, then since the ideal rankings are $1 - C_n$ the ideal average ranking of category C is given by

$$IAVRR(C) = \frac{\sum_{n=1}^{C_n} n}{C_n}$$

.

- $AVRR$ - average rank of relevant items for a particular retrieval. For a particular retrieval seeking images from the same category as a query image Q the average rank of relevant items for Q is given by

$$AVRR(Q) = \frac{\sum_{I \in cat(Q)} R(I)}{cat(Q)_n},$$

where $R(I)$ denotes the ranking of image I in the result, $cat(Q)$ is the category of query image Q, and $cat(Q)_n$ is the number of images in the category of query image Q.

The ratio of AVRR to IAVRR is a measure of the deviation from the ideal ranking. $\frac{AVRR}{IAVRR} = 1$ for a perfect retrieval (i.e., all images from the same category as the query image were ranked at the top). $\frac{AVRR}{IAVRR} > 1$ for all other cases. In general, a lower ratio implies better results. For each category C we compute the average normalized recall metric for C by taking the average of the results of $\frac{AVRR(Q)}{IAVRR(C)}$ for each image Q in category C. In addition, we compute the average normalized recall value over the whole database as the average AVRR to IAVRR ratio for all categories, and the total normalized recall as $\frac{AVRR(Q)}{IAVRR(cat(Q))}$ for each image Q in the database.

6.2 Results

	DP	DPA	INT	INTA	BD	BDA
fingerprints	1.28	1.79	1.66	1.73	2.15	2.17
floorplans	4.41	2.54	2.19	1.16	2.97	3.47
comics	3.44	4.51	2.47	1.69	3.31	3.67
music	3.51	2.30	3.31	1.07	3.40	3.05
tables	1.63	1.42	1.30	1.26	1.32	1.26
mug shots	4.14	1.18	2.38	1.88	1.84	1.84
house pictures	9.51	11.92	1.80	2.02	11.35	11.98
aircraft	9.65	10.50	11.71	9.53	10.50	11.58
animals	1.07	1.52	1.90	2.30	1.10	1.07
topo maps	2.01	2.68	1.54	1.92	3.35	2.99
road maps	6.11	9.62	7.17	5.85	8.65	8.94
satellite	3.24	4.03	4.40	3.60	6.03	5.51
aerial	5.23	3.88	4.28	5.12	11.42	10.98
Arabic	8.30	8.65	3.68	2.64	7.41	5.76
Chinese	1.00	1.00	4.55	1.07	1.00	1.00
Eng. print	7.14	5.30	6.27	4.50	8.35	8.86
Eng. lyrics	1.00	1.00	3.05	1.00	1.00	1.00
Eng. hand	4.48	3.27	2.76	1.80	4.05	3.90
Heb. script	1.33	2.50	1.17	1.11	1.22	1.11
Heb. hand	13.65	10.75	7.40	4.42	10.97	12.53
Heb. print1	1.00	1.00	1.39	1.00	1.00	1.00
Heb. print2	3.19	4.13	2.68	2.15	5.44	5.20
total	3.68	3.47	2.83	2.14	3.83	3.86
average	4.23	4.34	3.48	2.60	4.73	4.95

Table 1: Normalized recall results for each category using 3×3-grams of binarized original images (normalized recall$= 1$ for perfect retrieval, > 1 otherwise).

Tables 1– 2 summarize the normalized recall (NR) results for each image category for the various similarity criteria that we used. The average and total normalized recall over the whole database using each similarity criteria is also presented. Figures 2– 6 show the first 16 ranked images for one image from five of the twenty two categories. Each figure has three parts (a), (b), and (c) reporting the results using 3×3-gram with vector intersection score normalized by averages (*inta*), local standard deviation with histogram intersection (*sdvh*), and grey levels with distribution statistics

	GRLS	GRLH	SDVS	SDVH	GLCM	LAWS
fingerprints	3.35	1.14	3.46	1.08	1.55	3.00
floorplans	3.14	4.70	4.11	3.03	2.84	4.48
comics	4.57	2.68	2.01	1.85	3.08	1.75
music	15.52	16.35	15.79	12.53	17.55	15.89
tables	8.31	1.80	4.34	1.40	5.53	2.80
mug shots	9.39	9.60	4.41	2.67	7.02	5.17
house pictures	7.97	16.74	3.44	5.52	7.45	5.76
aircraft	9.44	5.53	4.11	3.45	3.70	4.58
animals	2.23	1.00	1.43	1.00	1.05	1.02
topo maps	3.10	1.23	4.88	2.21	3.87	5.67
road maps	3.27	9.83	2.50	7.81	7.89	3.43
satellite	4.92	11.88	4.84	4.08	5.04	6.77
aerial	5.71	20.00	4.77	4.66	8.66	8.60
Arabic	11.05	6.65	15.53	8.07	12.43	15.52
Chinese	6.00	5.96	4.63	3.87	2.45	3.23
Eng. print	4.31	4.86	3.20	4.37	3.40	4.57
Eng. lyrics	8.82	4.95	2.20	1.85	2.15	11.75
Eng. hand	3.15	8.03	4.34	7.56	7.14	4.23
Heb. script	1.11	1.00	1.17	1.00	1.00	1.94
Heb. hand	5.22	7.62	2.25	5.83	2.98	3.58
Heb. print1	1.44	1.00	1.00	1.00	1.00	1.00
Heb. print2	7.47	3.64	6.77	2.77	2.17	5.65
total	4.65	5.25	4.06	3.46	4.48	4.40
average	5.70	6.40	4.44	3.85	4.82	5.28

Table 2: Normalized recall results for each category using grey level features, local standard deviation features, grey level cooccurrence matrix features, and Laws texture features (normalized recall= 1 for perfect retrieval, > 1 otherwise).

features (*grls*), respectively. The first image out of the 16 is the query image. The names of all images that are from the same category as the query image are displayed in reverse video.

From these results it is apparent that overall the $N \times M$-gram based methods perform better than the grey level distribution and other texture feature methods. In addition, between the similarity measures for $N \times M$-gram vectors, the $N \times M$-gram vector intersection score normalized by averages (*inta*) is best with an average NR of 2.6 and a total NR of 2.14. The NR for the bin to bin (*bd*) and dot product (*dp*) similarity measures are very similar to each other and slightly higher than those of *inta*. Among the other texture feature based methods the local standard deviation with histogram intersection (*sdvh*) did best (average NR of 3.85 and total NR of 3.46), while the grey level histogram intersection did worst (average NR of 6.4 and total NR of 5.25).

Note, however, that this performance was not observed for all categories. The local standard deviation with histogram intersection (*sdvh*) did best in 4 categories: fingerprints, aircraft, animal statues, and Hebrew script. An example of the superiority of *sdvh* in the aircraft category can be seen in Figure 2 where 7 out of 8 of the images from the aircraft category in our database are among the top 16 using *sdvh*, as opposed to 3 for *grls* and *inta*. The local standard deviations with distribution statistic features (*sdvs*) performed best in 3 categories: roadmaps, English print, and Hebrew handwriting. The grey level histogram intersection (*grlh*) did best in the topographic map category with a NR of 1.23. Note, however,r that *int* and *inta* also have relatively low NR values for this category (1.54 and 1.92, respectively).

Figure 2: Results for image from the aircraft category (a) 3×3-gram with vector intersection score normalized by averages (*inta*). (b) local standard deviation with histogram intersection (*sdvh*). (c) grey levels with distribution statistics features (*grls*).

The dot product similarity measure for $N \times M$-grams performed particularly well in the satellite images and aerial photo categories. Finally the $N \times M$-gram vector intersection normalized by averages method (*inta*) was best in 11 categories. However, its overall normalized recall is much better than the other methods. It performed particularly well compared to the other methods in the floorplans, comics, music, tables, mugshots, Chinese, Arabic and house picture categories. It was weakest in the aircraft and roadmap categories.

Figures 3, 4, 5, and 6 are examples where *inta* is clearly better for the Chinese, music, English handwriting, and mug shots categories, respectively. The *inta* method ranked the 9 Chinese documents in our database 1–9 (a perfect normalized recall), 8 of these 9 Chinese documents are ranked in the top 16 using *sdvh*, while only 2 are in the top 16 using *grls*. In the music category, *inta* again ranked the 9 musical scores in our database perfectly, *sdvh* found 6 in the top 16 and ranked them relatively lower, and *grls* only found 3 in the top 16 and also ranked them low. In the English handwriting category 11 out of the 15 images from this category in our database are ranked in the

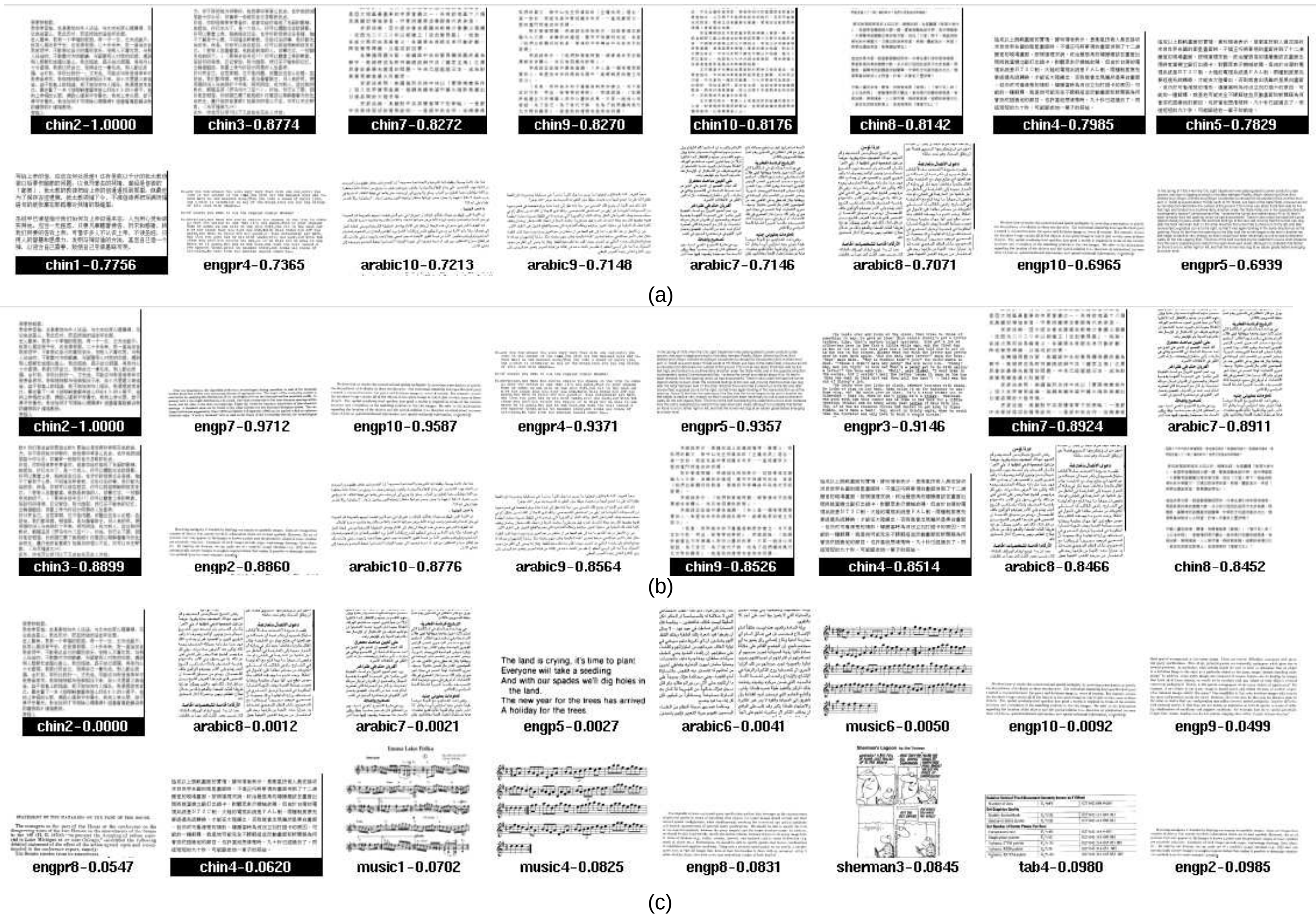

Figure 3: Results for image from the Chinese category (a) 3×3-gram with vector intersection score normalized by averages ($inta$). (b) local standard deviation with histogram intersection ($sdvh$). (c) grey levels with distribution statistics features ($grls$).

top 16 using $inta$, $sdvh$ found 6 in the top 16, and $grls$ found 5 in the top 16. Finally, in the mug shots category 10 out of the 11 mug shots in our database are in the top 16 using $inta$, 10 mug shots are also in the top 16 using $sdvh$, however they are ranked lower, and only 5 are in the top 16 using $grls$.

7 Concluding Remarks

As can be seen from the results, $N \times M$-grams work very well on simple documents such as floorplans, music notes, comics, and tables. The $N \times M$-gram method was however weak on richer images such as aircraft, aerial photos, and road maps. There are numerous modifications that will most likely improve these results. We computed the $N \times M$-grams on the images following a threshold operation on the original greyscale levels. It may be more useful to compute these on the gradient image or on the images following some edge detection. Furthermore, the choice of 3×3-grams for the experiments

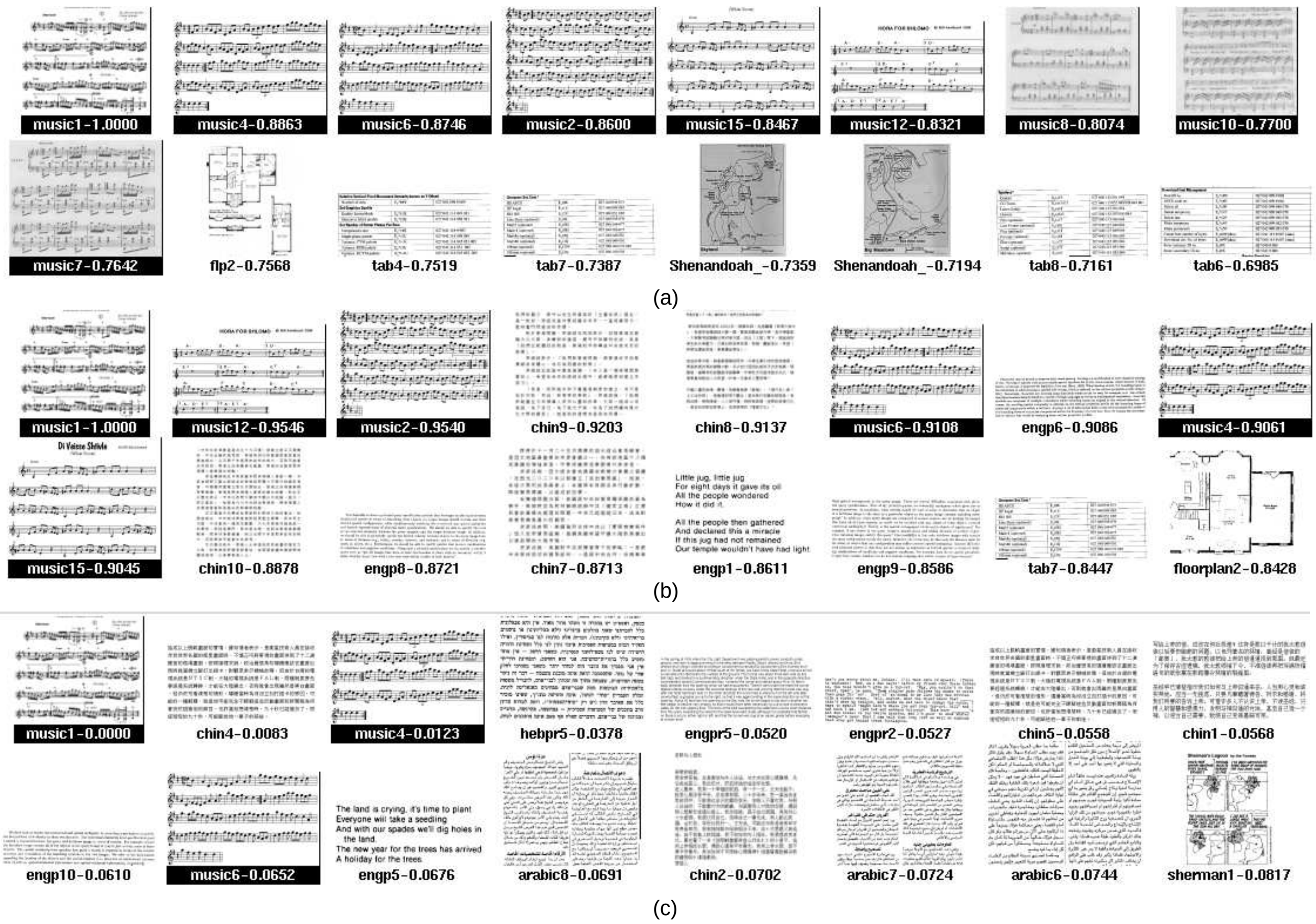

Figure 4: Results for image from the music category (a) 3×3-gram with vector intersection score normalized by averages ($inta$). (b) local standard deviation with histogram intersection ($sdvh$). (c) grey levels with distribution statistics features ($grls$).

was quite arbitrary and may in fact be too small a window. Using a larger window may yield better results, but will increase the storage as the dimensionality of the $N \times M$-gram vectors will be higher. One solution for both problems is to use multiple window sizes by applying the same size window to various resolutions of the image. We intend to use the wavelet transform [12] for this purpose. By computing $N \times M$-grams on both the low pass and the high pass components of the wavelet decomposition of the images we will in effect get both edge information and multi-resolution images.

The experiments that we conducted rank all of the images in the database with respect to one particular image. Another interesting experiment is to try to cluster the images using the three similarity measures. This could either be done totally automatically using some form of k-means clustering or by first providing some examples for each category as a training set and then clustering the remaining images according to these initial clusters.

In our discussion we did not consider the running time and indexing associated with the methods that we compared. The methods that use statistical distribution features such as grey level statistics,

Figure 5: Results for image from the English handwriting category (a) 3×3-gram with vector intersection score normalized by averages ($inta$). (b) local standard deviation with histogram intersection ($sdvh$). (c) grey levels with distribution statistics features ($grls$).

and local standard deviation value statistics only require a relatively small number of features per image, and thus it is straightforward to index these feature vectors for fast searches using an appropriate multi-dimensional indexing structure [4]. In contrast, it is not as easy to index $N \times M$-gram vectors since their dimensionality is very high. However, there is much room for improvement of $N \times M$-gram based methods in terms of efficiency (e.g., by use of dimensionality reduction techniques). If the results of $N \times M$-gram based methods are significantly better in terms of retrieval accuracy, then the extra computing time will be well worth it.

8 Acknowledgments

I thank Jacqueline Le Moigne for her comments on this manuscript. I have benefited from discussions with Ethan Miller.

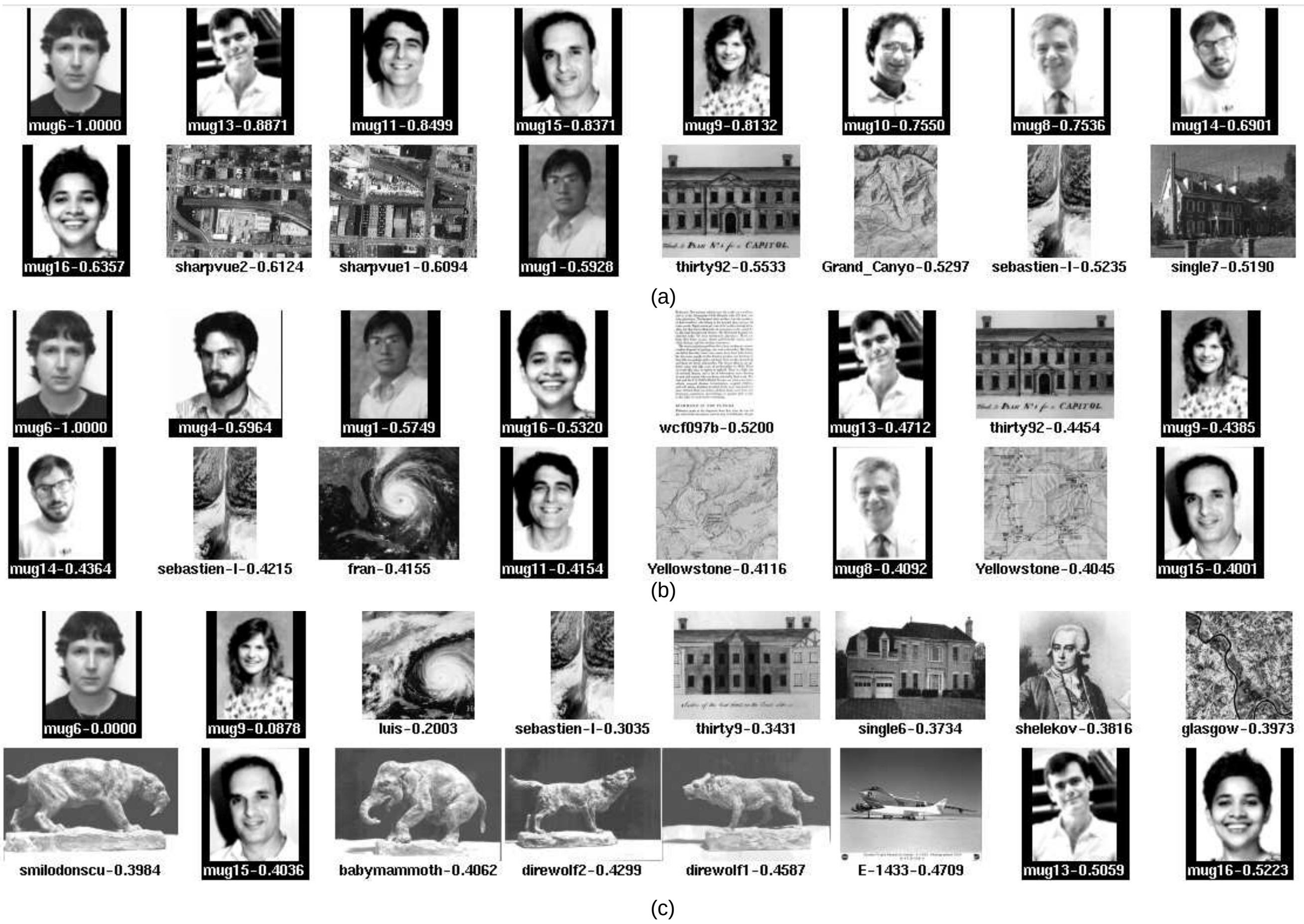

Figure 6: Results for image from the Mug shots category (a) 3×3-gram with vector intersection score normalized by averages ($inta$). (b) local standard deviation with histogram intersection ($sdvh$). (c) grey levels with distribution statistics features ($grls$).

References

[1] G.L Cash and M. Hatamian. Optical character recognition by the method of moments. *Computer Vision, Graphics, and Image Processing*, 39(3):291–310, September 1987.

[2] M. Damashek. Gauging similarity with n-Grams: Language-independent categorization of text. *Science*, 267:843–848, February 1995.

[3] C. Faloutsos, R. Barber, W. Equitz, M. Flickner, W. Niblack, and D. Petkovic. Efficient and effective quering by image content. *Journal of Intelligent Information Systems*, pages 231–62, 1994.

[4] J.H. Friedman, J.L. Bentley, and R.A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1977.

[5] G. L. Gimel'farb and A. K. Jain. On retrieveing textured images from an image database. *Pattern Recognition*, 29(9):1461–1483, September 1996.

[6] W. I. Grosky and R. Mehrotra. Index-based object recognition in pictorial data management. *Computer Vision, Graphics, and Image Processing*, 52(3):416–436, December 1990.

[7] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-3:610–621, 1973.

[8] D. P. Huijsmans, S. Poles, and M. Lew. 2D pixel trigrams for content-based image retrieval. In A. W. M. Smeulders and R. Jain, editors, *Proceedings of the 13th International Conference on Pattern Recognition*, pages 139–145, Amsterdam, The Netherlands, August 1996.

[9] R. Jain. NSF workshop on visual information management systems. *SIGMOD RECORD*, 22(3):57–75, September 1993.

[10] P.M. Kelly, T.M. Cannon, and D.R. Hush. Query by image example: the CANDID approach. In *Proceeding of the SPIE, Storage and Retrieval of Still Image and Video Databases III*, volume 2420, pages 238–248, San Jose, CA, February 1995.

[11] K. I. Laws. Rapid texture identification. In *Proceeding of the SPIE*, volume 238, pages 376–380, 1980.

[12] S. Mallat. A theory for multiresolution signal decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1995.

[13] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, August 1996.

[14] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yanker. The QBIC project: Querying images by content using color, texture, and shape. In *Proceeding of the SPIE, Storage and Retrieval of Image and Video Databases*, volume 1908, pages 173–187, San Jose, CA, February 1993.

[15] T. Ojala, M. Pietikäinen, and D. Harwood. A comparitive study of texture measures with clssification based on feature distributions. *Pattern Recognition*, 29(1):51–59, January 1996.

[16] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, 1st edition, 1983.

[17] A. Soffer and H. Samet. Retrieval by content in symbolic-image databases. In *Proceedings of the SPIE, Storage and Retrieval of Still Image and Video Databases IV*, volume 2670, pages 144–155, San Jose, CA, February 1996.

[18] M. Stricker and M. Orengo. Similarity of color images. In *Proceeding of the SPIE, Storage and Retrieval of Still Image and Video Databases III*, volume 2420, pages 381–392, San Jose, CA, February 1995.

[19] M. Swain. Interactive indexing into image databases. In *Proceeding of the SPIE, Storage and Retrieval for Image and Video Databases*, volume 1908, pages 95–103, San Jose, CA, February 1993.