Texture recognition by using GLCM and various aggregation functions

Gleb Beliakov, Simon James and Luigi Troiano

***Abstract*— We discuss the problem of texture recognition based on the grey level co-occurrence matrix (GLCM). We performed a number of numerical experiments to establish whether the accuracy of classification is optimal when GLCM entries are aggregated into standard metrics like contrast, dissimilarity, homogeneity, entropy, etc., and compared these metrics to several alternative aggregation methods. We conclude that k nearest neighbors classification based on raw GLCM entries typically works better than classification based on the standard metrics for noiseless data, that metrics based on principal component analysis inprove classification, and that a simple change from the arithmetic to quadratic mean in calculating the standard metrics also improves classification.**

I. Introduction

Texture is a qualitative property we use to describe the touch or visual aspects of surfaces. The problem considered here is the recognition and characterization of visual textures in image processing. Intuitive and often broad notions of texture make it difficult to articulate a specific mathematical definition, however it is generally accepted that it refers to an optical pattern with a number of elements including spatial variations in intensity or wavelength [1]. Smooth textures, for instance, are characterized by subtle and gradual spatial variations, whilst course textures refer to the opposite - drastic variations over short distances. Of course, there are other textural ideas not located on a continuum between these two, such as scaly, furry, bumpy, swirly, and so on. Some of these are only identifiable in terms of touch, others are purely visual notions.

The grey level co-occurrence matrix (GLCM) was introduced by Haralick et al in 1973 [2], [3] as the "grey-tone spatial-dependence matrix" along with 14 metrics derived from it to quantify textures. The GLCM indicates how often different combinations of grey levels occur in a given image. It is a symmetric and square matrix with each element $\{i, j\}$ indicating the probability of a grey level i, neighboring a grey level j in a given direction [1] The size of the GLCM is dependent on the number of grey levels considered, not the size of the image itself. Metrics derived from the GLCM are hence based on the relationship between any two neighboring pixels. They include the GLCM mean, contrast and entropy. In combination, these features can be used to quantitatively evaluate a small window of a digital image and characterize its texture. They are also used for classification, for instance, by k nearest neighbor (kNN) methods.

The mentioned metrics may not discriminate well between various textures. In this contribution we are interested in defining new features that may discriminate certain textures better. We note that most of the mentioned features (see Section II for their definitions) are weighted means of GLCM entries, which are averaging aggregation functions [4]–[7]. Then a valid question is whether there are other aggregation functions that may discriminate textures better. Further, for a chosen broad class of aggregation functions, is it possible to compute its parameters that maximize discrimination?

In this paper we consider the classes of weighted arithmetic means and general linear functions as alternatives for calculating texture features. The next section outlines the details of the GLCM method and provides some basic definitions. In Section III we formulate our problem of calculating new image features. In Section IV we discuss our numerical experiments and present their results. Section V contains the conclusions.

II. Basic definitions

A. GLCM calculation

In order to calculate the GLCM one must first define the window size, number of grey levels and offset direction. Where n is the number of grey levels considered, the GLCM will be an $n \times n$ matrix. As an intermediate step, an $n \times n$ framework matrix can be defined which counts the occurrences of neighboring pixels with the given spatial relationship. For instance, using east as the offset direction with four grey levels (1=black, 4=white), the cell (1,1) would record the number of times black occurs next to black (either moving 1 pixel east or 1 pixel west). The cell (1,3) would record the number of times black occurs next to light grey, or light grey next to black. This will hence be the same entry as (3,1). This framework matrix is then normalized by calculating, for each cell, the occurrences of each grey-level combination divided by the total of all cells. The following metrics are derived from the GLCM P.

Angular Second Moment (ASM)

$$\sum_{i,j=0}^{N-1} (P_{i,j})^2$$

Gleb Beliakov and Simon James are with the School of Engineering and Information Technology, Deakin University, 221 Burwood Hwy, Burwood, 3125, Australia (phone: +613 9251 74754; email: gleb@deakin.edu.au, sjg@deakin.edu.au). Luigi Troiano is with RCOST - University of Sannio, 82100 Benevento, Italy (email: troiano@unisannio.it)

[1]The matrix is made symmetric by calculating the occurrence of neighboring grey levels in one direction and then again in the opposite direction. This way, the probability of i being next to j is the same as j to i. In this paper, we have used the average of the 8 compass directions, however this is not always the case for the GLCM.

978-1-4244-1819-0/08/$25.00©2008 IEEE

Contrast (CON)

$$\sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2$$

Dissimilarity (DIS)

$$\sum_{i,j=0}^{N-1} P_{i,j}|i-j|$$

Homogeneity (HOM)

$$\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2}$$

Maximum Probability (MAX)

$$\max P_{ij}$$

Entropy (ENT) (assume that $0 * \ln(0) = 0$)

$$\sum_{i,j=0}^{N-1} P_{i,j}(-\ln P_{i,j})$$

Energy (ENG) $= \sqrt{ASM}$
GLCM Mean

$$\mu_i = \sum_{i,j=0}^{N-1} iP_{i,j},\ \mu_j = \sum_{i,j=0}^{N-1} jP_{i,j}$$

GLCM Variance

$$var = \sigma_i^2 = \sum_{i,j=0}^{N-1} P_{i,j}(i-\mu_i)^2$$

GLCM Correlation

$$\sum_{i,j=0}^{N-1} P_{i,j}\left(\frac{(i-\mu_i)(j-\mu_j)}{\sigma_i\sigma_j}\right)$$

B. Aggregation functions

Aggregation functions play an important role in several areas, including fuzzy logic, decision making, expert systems, risk analysis and image processing. Recent books [4]–[7] provide a comprehensive overview of aggregation functions and methods of their construction.

The purpose of aggregation functions is to combine several input values into a single output value, which in some sense represents all the inputs. Typically the inputs and outputs are real numbers from $[0, 1]$, although other choices are possible. Notable examples are weighted means, medians, ordered weighted averaging (OWA) functions, discrete Choquet and Sugeno integrals, triangular norms and conorms, uninorms and nullnorms.

Definition 1: An aggregation function is a function of $n > 1$ arguments $f : [0,1]^n \to [0,1]$, with the properties

(i) $f(\underbrace{0,0,\ldots,0}_{n-times}) = 0$ and $f(\underbrace{1,1,\ldots,1}_{n-times}) = 1$.

(ii) $\mathbf{x} \le \mathbf{y}$ implies $f(\mathbf{x}) \le f(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in [0,1]^n$.

The vector inequality is understood componentwise. Aggregation functions may possess various properties, which often classify them into special classes. We are interested in averaging functions. An aggregation function f is called averaging if it is bounded (for all $\mathbf{x} \in [0,1]^n$) by

$$\min(\mathbf{x}) = \min_{i=1,\ldots,n} x_i \le f(\mathbf{x}) \le \max_{i=1,\ldots,n} x_i = \max(\mathbf{x}).$$

This condition is equivalent to idempotency: $f(t,t,\ldots,t) = t$ for ant $t \in [0,1]$.

Weighted arithmetic means (WAM) are the most common aggregation functions. GLCM metrics CON, DIS, HOM and GLCM mean are the WAM (up to a normalizing constant), the root of ASM is a quadratic mean, MAX is also an aggregation function.

In this work we are interested in other means, e.g. weighted power mean

$$M_{r;w}(x) = \left(\sum_{i=1}^{n} w_i x_i^r\right)^{1/r},$$

with $\sum w_i = 1, w_i \ge 0, i = 1,\ldots,n$, as well as other linear functions (not aggregation functions)

$$L_w(x) = \sum_{i=1}^{n} w_i x_i + w_0$$

in which w_i are unrestricted.

III. Calculation of new image features

We saw from Section II that many of the GLCM derived features are weighted means of GLCM entries. Hence our main question is whether other means, or linear functions can be used as features to improve classification.

Consider a multiclass classification problem. We shall use a standard kNN method for classification, with the train and test data given as follows.

- Raw GLCM entries (in total $\frac{n\times(n-1)}{2} + n$ entries (the matrix is symmetric);
- Standard metrics used as features: ASM, CON, DIS, HOM, ENT, and GLCM means, 10 features in total)
- Standard metrics, but with CON, DIS and HOM replaced with power means (r=1.5, 2);
- 10 principal components of the GLCM matrix;
- 12 linear functions of GLCM obtained by linear discriminant analysis (LDA);
- 12 linear functions of GLCM obtained by minimizing a weighted sum of intra- and interclass variations (the method is outlined below).

The number of principal components was chosen to coincide with the number of standard features, and the number of linear functions is the same as the total number of classes we had.

As the experimental data we took the set of 12 Brodatz textures available from `http://sipi.usc.edu/database/database.cgi` (see Figs. 1-6).

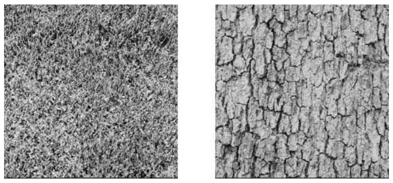
Fig. 1. a) Grass b) Bark

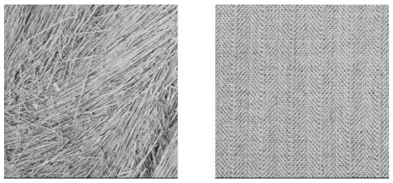
Fig. 2. a) Straw b) Herringbone weave

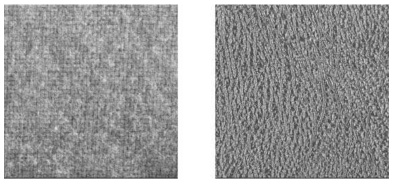
Fig. 3. a) Woolen cloth b) Pressed calf leather

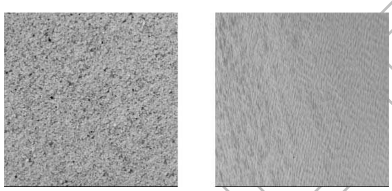
Fig. 4. a) Beach sand b) Water

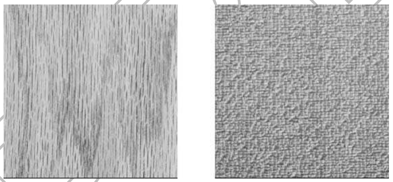
Fig. 5. a) Wood grain b) Raffia

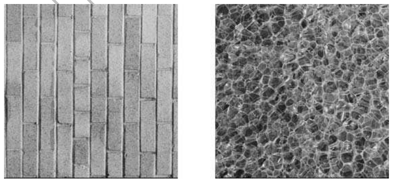
Fig. 6. a) Brick wall b) Plastic bubbles

For each image we calculated several GLCMs with $n = 8$ gray levels[2] using 64×64 window with a random upper left corner. The data set was split into training (600 GLCMs) and test (1800 GLCMs) parts.

From each GLCM in the training data set we calculated various metrics as outlined above. We used the Minitab 12.1 software to perform PCA and LDA on the raw GLCM and calculated coefficients of 10 and 12 linear functions respectively. These linear functions were used to determine and calculate the new metrics for both train and test data sets. We also calculated another set of 12 linear discriminant functions based on minimizing the following criterion

$$\text{minimize } IntraclassSTD_i - P \times InterclassSTD_i \quad (1)$$

$i = 1, \ldots, 12$, $P = 0.5$. The interclass standard deviation involves all classes but the i-th. This criterion is similar in purpose to LDA, in which the ratio of Interclass/Intraclass standard deviations is maximized, but involves a different criterion. We performed optimization by a combination of random start and discrete gradient method [8], [9].

In addition we performed the same type of experiments with noisy test data. The noise was generated by superimposing two textures, the original image and an image with vertical strips, transparency 0.9. This kind of noise cannot be eliminated from the image by standard de-noising methods (see Figs. 7-8).

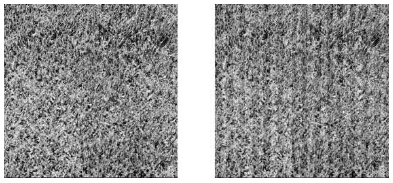
Fig. 7. a) Original (grass) b) With noise

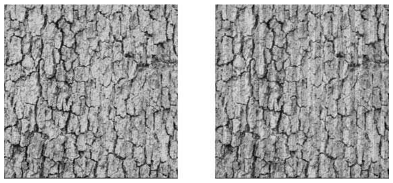
Fig. 8. a) Original (bark) b) With noise

IV. Discussion of the Results

We performed the following experiments using kNN standard classification method on M-class classification problems, $M = 4, 5, 6, 8, 10, 12$. For $M < 12$ we used 10 different combinations of M out of 12 classes to collect statistics. We used $k = 1$ and $k = 5$ and standardized the train and test data to have 0 means and 1 standard deviation.

[2]The original images have 256 gray levels. A GLCM with $n = 8$ gray levels is a symmetric 8×8 matrix. Larger values of n would result in too long feature vectors for some methods as well as a disproportionate number of zero entries in the matrix leading to less informative GLCM calculations.

Tables I-VI summarize our results. We see that the raw GLCM data provides the highest accuracy on noiseless and noisy data. This is not unexpected, as any aggregation method reduces the available information, although for data with noise this may be desired to avoid overfitting. We also observed that $k = 1$ generally provides better accuracy than $k = 5$. Also larger M lead to decreasing accuracy, as expected. Brick was the most common texture to be misclassified, often being confused for Beach sand (Fig. 4a) or Wood grain (Fig. 5a).

The use of principal components improves the accuracy given by the standard metrics. The methods based on LDA and minimizing criterion (1) are marginally worse than the standard metrics, and there was no clear difference in performance of these methods.

Changing the contrast from the arithmetic to power mean did not improve the performance, but changing dissimilarity and homogeneity from arithmetic to power means (with the same weights) systematically improved the accuracy by a relatively small factor.

We note that the standard metrics contain redundant features. We performed an analysis based on removing some of the metrics from calculating distances in kNN, and noticed that the performance stays the same if the GLCM mean, variance and dissimilarity are removed. This leaves us with the important metrics ASM, CON, ENT, ENG, DIS and Correlation. With this reduced set of metrics (7 metrics row in the tables) the accuracy marginally improved.

TABLE I
AVERAGE ACCURACY OF CLASSIFICATION (STANDARD DEVIATION) FOR M=4 CLASS PROBLEM

Method	Noiseless k=1	Noiseless k=5	Noisy k=1	Noisy k=5
GLCM raw	**0.967(0.04)**	**0.951(0.05)**	0.936(0.06)	0.921(0.07)
Metrics	0.957(0.05)	0.947(0.05)	0.940(0.06)	0.943(0.05)
PCA	0.957(0.05)	0.942(0.06)	0.928(0.08)	0.907(0.09)
LDA	0.959(0.04)	0.936(0.05)	0.941(0.07)	0.839(0.08)
Crit.(1)	0.947(0.06)	0.936(0.07)	0.89(0.10)	0.875(0.10)
Quadratic	0.962(0.04)	0.949(0.05)	**0.946(0.05)**	**0.945(0.04)**
7 Metrics	0.957(0.05)	0.947(0.05)	0.94(0.05)	0.94(0.05)

TABLE II
AVERAGE ACCURACY OF CLASSIFICATION (STANDARD DEVIATION) FOR M=5 CLASS PROBLEM

Method	Noiseless k=1	Noiseless k=5	Noisy k=1	Noisy k=5
GLCM raw	**0.973(0.03)**	**0.960(0.03)**	0.923(0.05)	0.914(0.05)
Metrics	0.963(0.04)	0.951(0.04)	0.928(0.06)	0.931(0.05)
PCA	0.967(0.03)	0.954(0.04)	0.919(0.06)	0.904(0.06)
LDA	0.957(0.03)	0.932(0.04)	0.809(0.07)	0.798(0.08)
Crit.(1)	0.958(0.04)	0.945(0.05)	0.879(0.09)	0.868(0.09)
Quadratic	0.967(0.04)	0.952(0.04)	**0.934(0.06)**	**0.932(0.05)**
7 Metrics	0.962(0.04)	0.951(0.04)	0.927(0.06)	0.928(0.05)

TABLE III
AVERAGE ACCURACY OF CLASSIFICATION (STANDARD DEVIATION) FOR M=6 CLASS PROBLEM

Method	Noiseless k=1	Noiseless k=5	Noisy k=1	Noisy k=5
GLCM raw	**0.961(0.03)**	**0.941(0.04)**	**0.915(0.04)**	0.903(0.07)
Metrics	0.947(0.04)	0.935(0.04)	0.894(0.04)	**0.914(0.05)**
PCA	0.951(0.03)	0.930(0.05)	0.901(0.05)	0.877(0.06)
LDA	0.935(0.03)	0.905(0.04)	0.779(0.09)	0.765(0.10)
Crit.(1)	0.928(0.04)	0.922(0.05)	0.866(0.06)	0.855(0.06)
Quadratic	0.952(0.04)	0.935(0.04)	0.911(0.06)	0.913(0.05)
7 Metrics	0.947(0.04)	0.935(0.04)	0.904(0.06)	0.910(0.05)

TABLE IV
AVERAGE ACCURACY OF CLASSIFICATION (STANDARD DEVIATION) FOR M=8 CLASS PROBLEM

Method	Noiseless k=1	Noiseless k=5	Noisy k=1	Noisy k=5
GLCM raw	**0.955(0.02)**	**0.932(0.03)**	**0.885(0.03)**	0.863(0.03)
Metrics	0.939(0.03)	0.926(0.03)	0.860(0.03)	0.883(0.02)
PCA	0.941(0.03)	0.921(0.04)	0.879(0.04)	0.861(0.04)
LDA	0.927(0.03)	0.895(0.04)	0.737(0.04)	0.721(0.04)
Crit.(1)	0.925(0.04)	0.906(0.04)	0.798(0.07)	0.790(0.07)
Quadratic	0.945(0.03)	0.927(0.03)	0.870(0.02)	**0.886(0.02)**
7 Metrics	0.939(0.03)	0.927(0.03)	0.859(0.03)	0.878(0.03)

TABLE V
AVERAGE ACCURACY OF CLASSIFICATION (STANDARD DEVIATION) FOR M=10 CLASS PROBLEM

Method	Noiseless k=1	Noiseless k=5	Noisy k=1	Noisy k=5
GLCM raw	**0.955(0.01)**	**0.931(0.02)**	**0.874(0.02)**	0.848(0.03)
Metrics	0.938(0.02)	0.923(0.01)	0.828(0.02)	0.859(0.01)
PCA	0.943(0.01)	0.920(0.02)	0.87(0.02)	0.847(0.03)
LDA	0.916(0.02)	0.882(0.02)	0.716(0.02)	0.701(0.03)
Crit.(1)	0.926(0.02)	0.904(0.02)	0.782(0.04)	0.776(0.04)
Quadratic	0.944(0.02)	0.925(0.02)	0.839(0.02)	**0.862(0.02)**
7 Metrics	0.938(0.02)	0.923(0.01)	0.827(0.02)	0.855(0.02)

TABLE VI
ACCURACY OF CLASSIFICATION FOR M=12 CLASS PROBLEM

Method	Noiseless k=1	Noiseless k=5	Noisy k=1	Noisy k=5
GLCM raw	**0.951**	**0.926**	**0.858**	0.832
Metrics	0.932	0.919	0.807	0.843
PCA	0.937	0.914	0.855	0.833
LDA	0.909	0.874	0.697	0.680
Crit.(1)	0.919	0.896	0.753	0.752
Quadratic	0.938	0.92	0.818	**0.847**
7 Metrics	0.933	0.92	0.807	0.838

V. CONCLUSIONS

The main conclusions we can make from our analysis are the following.

- Some of the standard metrics derived based on GLCM are redundant.
- Dimensions found by PCA are better suited to discrim-

inate between classes.

- Methods based on optimizing inter and intra-class variance are generally not better than the standard metrics.
- Standard metrics can be improved by using weighted power means rather than arithmetic means.
- Without any noise, or with a small noise level, raw GLCM entries better discriminate classes with the nearest neighbor method.

Of course, when using each of the approaches, one needs to take into account the balance between the accuracy and computational effort. While dimensions found by PCA improve the accuracy, their computation may not be always warranted. In contrast, a simple change from the arithmetic to quadratic mean does not involve much extra effort, but systematically improves the accuracy, and can be suggested for routine computations.

Our next goal is to analyze the image segmentation problem based on texture. This is a more challenging problem because the neighborhood of each pixel may contain several textures. Further, we will investigate the use and aggregation of the directional GLCMs (at the moment we simply average them). This may help to eliminate various anisotropic effects (such as image shrinking in one dimension). Alternative aggregation methods of non-redundant standard metrics is also part of this line of research.

References

[1] R. M. Pickett, "Visual analysis of texture in the detection and recognition of objects," *Picture Processing and Psychopictorics*, pp. 298–308, 1970.

[2] R. M. Haralick, K. S. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–620, 1973.

[3] R. M. Haralick and K. S. Shanmugam, "Combined spectral spatial processing of erts imagery data," *Remote Sensing of Environment*, vol. 3, pp. 3–13, 1974.

[4] G. Beliakov, A. Pradera, and T. Calvo, *Aggregation Functions: A Guide for Practitioners*. Heidelberg, Berlin, New York: Springer, 2007.

[5] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap, *Aggregation Functions*. to appear: Elsevier, 2008.

[6] Y. Torra, V. Narukawa, *Modeling Decisions. Information Fusion and Aggregation Operators*. Berlin, Heidelberg: Springer, 2007.

[7] T. Calvo, G. Mayor, and R. Mesiar, Eds., *Aggregation Operators. New Trends and Applications*. Heidelberg, New York: Physica-Verlag, 2002.

[8] A. Bagirov, "A method for minimization of quasidifferentiable functions," *Optimization Methods and Software*, vol. 17, pp. 31–60, 2002.

[9] G. Beliakov and J. Ugon, "Implementation of novel methods of global and non-smooth optimization: GANSO programming library," *Optimization*, vol. 56, pp. 543–546, 2007.