Use of Band Ratioing for Color Texture Classification

Rubén Muñiz[1], Santiago García[1], Fermín González[2], José Antonio Corrales[1]

[1] Department of Computer Science. University of Oviedo, Gijón, SPAIN
{rubenms,carbajal,ja}@lsi.uniovi.es
[2] Indra, Madrid, SPAIN
fermingm@indra.es

Abstract. In the recent years, many authors have begun to exploit the extra information provided by color images to solve many computer vision tasks. In the field of texture classification, this lead us to develop new feature extraction algorithms or to modify the existing ones. In this paper a new method based on Gabor filters and band ratios is presented. For testing purposes, 30 color textures have been selected from the Vistex database. We will perform a number of experiments over those texture set, and we will classify them using a standard supervised learning algorithm and a GA-P classifier.

1 Introduction

Texture classification is a key field in many computer vision applications, ranging from quality control to remote sensing. Briefly described, there are a finite number of texture classes we have to learn to recognize. In the first stage of the development of such kind of systems, we extract useful information (features) from a set of digital images, known as the training set, containing the textures we are studying. Once this task has been done, we proceed to classify an unknown texture into one of the known classes. This process can be summarized in the following steps:

1. Image (texture) acquisition and preprocessing
2. Feature extraction
3. Feature selection (optional)
4. Classification

Since the earlier approaches to the problem, grayscale images has been widely used, primarily due to acquisition hardware limitations and/or limited processing power. In the near past, much effort has been done to develop new feature extraction algorithms (also known as texture analysis algorithms) to take advantage of the extra information contained in color images. On the other hand, many classical grayscale algorithms has been extended to process color textures [1][2][3].

Texture analysis algorithms can be divided into statistical and spectral ones. The former methods extract a set of statistical properties from the spatial distribution of intensities of a texture. Common examples of this approach are the histogram method and the family of algorithms based on coocurrence matrices [4][5]. The latter techniques, on the other hand, computes a number of features obtained from the analysis of the local spectrum of the texture. In the following section we will see in detail one of the most commonly used spectral methods.

2 Gabor filters

Gabor filters have been extensively used for extracting features from grayscale and color textures [6][1][3][7]. These filters are optimally localized in both space and spatial frequency and allows to get a set of filtered images which correspond to a specific scale and orientation component of the original texture. There are two major approaches to texture analysis using Gabor filters. First, one can look for specific narrowband filters to describe a given texture class, while the other option is to apply a bank of Gabor filters over the image and process its outputs to obtain the features that describe the texture class.

In this paper, we will use the second approach to extract a number of features from a database of 30 color textures, which will be presented later.

2.1 2D Gabor filterbank

The Gabor filter bank used in this work is defined in the spatial domain as follows:

$$f_{mn}(x,y) = \frac{1}{2\pi\sigma_m^2} \exp^{-\frac{x^2+y^2}{2\sigma_m^2}} \times \cos 2\pi(u_m x \cos\theta + u_m y \sin\theta) \ . \quad (1)$$

where m and n are the indexes for the scale and the orientation, respectively, for a given Gabor filter. Depending on these parameters, the texture will be analyzed (filtered) at a specific detail level and direction. The half peak radial and orientation bandwidths [6] are defined as follows:

$$B_r = \log_2\left(\frac{2\pi\sigma_m u_m + \sqrt{2\ln 2}}{2\pi\sigma_m u_m - 2\sqrt{2\ln 2}}\right) \ . \quad (2)$$

$$B_\theta = 2\tan^{-1}\left(\frac{\sqrt{2\ln 2}}{2\pi\sigma_m u_m}\right) \ . \quad (3)$$

As in [1] we define a filterbank with three scales and four orientations. The bandwidth B_θ is taken to be 40° in order to maximize the coverage of the frequency domain and minimize the overlap between the filters.

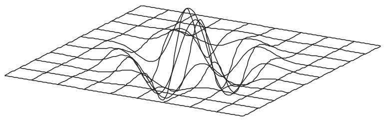

Fig. 1. Gabor filter computed by $f_{1,45^\circ}$

2.2 Gabor features

To obtain texture features we must filter the texture images using the generated filters. This is achieved by convolution on the frequency domain (4), due to the size of the filters used. For each filtered image, we extract a single feature μ_{mn} which represents its energy, as shown below.

$$G_{mn}(x,y) = I(x,y) * f_{mn} \ . \tag{4}$$

$$\mu_{mn}(x,y) = \sqrt{\left(\sum_{x,y} G^2_{mn}(x,y)\right)} \ . \tag{5}$$

This approach is only valid when grayscale images are used. If we want to filter a color image, we have to preprocess it before this method can be applied. The more obvious solution for this problem is to transform the image by a weighted average of the three color bands.

$$I(x,y) = aR(x,y) + bG(x,y) + cB(x,y) \ . \tag{6}$$

The coefficients a, b, c from (6), can be selected to properly model the human's eye perception model of color. For this purpose, an adequate choice of these weights can be $a = 0.54, b = 0.32, c = 0.14$.

Using this transformation, different colors can give the same grayscale intensity, so color information is lost. To overcome this obstacle, (4) can be applied on each of the RGB color bands of the image to obtain unichrome features [1]. With this approach, we obtain a set of energies from each spectral band, so the information extracted from textures, grows by a factor of three. Another disadvantage of this technique is that color information is not correlated because it is simply concatenated.

3 Band ratioing

3.1 Introduction

Band ratioing is a enhancement technique mainly used in the field of remote sensing. It is usually applied to process LANDSAT TM images[3] to enhance details such as vegetation, grass, soil, etc. It is defined as follows:

[3] http://rst.gsfc.nasa.gov/Sect1/Sect1_15.html

$$I(x,y) = \frac{B_1(x,y)}{B_2(x,y)} \quad . \tag{7}$$

where $B_1(x,y)$ and $B_2(x,y)$ are two different spectral bands of the color image. Its computation is extremely easy, but the bands involved must be processed to avoid the case when $B_2(x,y) = 0$. To accomplish this, we only have to increase every pixel from both bands by 1. In theory, ratios will be in the interval (0,256], but in practice most values will be rather small. For this reason, it is advisable to use logarithm compression to enhance small ratios over larger ones, so (7) can be rewritten as follows.

$$I'(x,y) = \log\left(\frac{B_1(x,y)}{B_2(x,y)}\right) \quad . \tag{8}$$

It can be easily seen that this technique tends to enhance what is different in two spectral bands, and as it will be seen in the following section, its output is suitable for feature extraction.

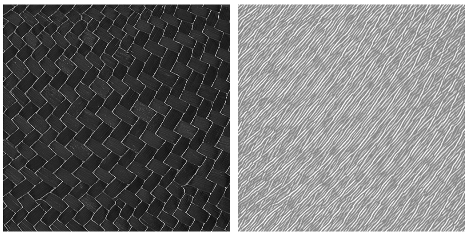

Fig. 2. Ratio R/B and filtered ratio of Fabric0

3.2 Gabor filtering of rationed color textures

In the previous section, we saw that Band Ratioing enhances what is different in two color bands. If a pixel contains a grayscale value ($R = G = B$), its ratio will be 1, but if at least two color components are not equal, the band ratio will encode the color information in a single value. This is very interesting for feature extraction from color textures, since we can directly use any grayscale feature extraction method available. In this paper, we have selected Gabor filtering method, since it is extensively used for texture classification and segmentation by many researchers due to its good performance.

To apply Gabor filtering on a rationed image, we can combine (8) and (4) to get the following expression:

$$G'_{mn}(x,y) = \log\left(\frac{B_1(x,y)}{B_2(x,y)}\right) * f_{mn} \quad . \tag{9}$$

Note that (9) directly convolves the band ratios with the Gabor filter, so it is not necessary to scale the ratios to fit in a byte value.

In Fig. 2 we can see a band ratio of a color texture taken from the VisTex database[4] and the result of filtering the ratio using a Gabor filter with parameters $m = 1, n = 45°$.

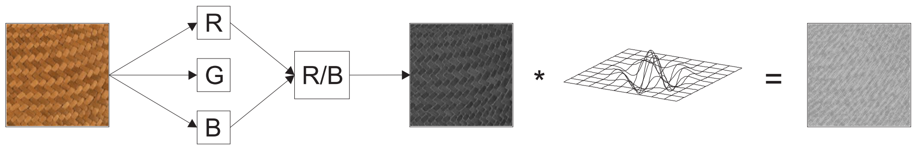

Fig. 3. Gabor filtering of rationed images

4 GA-P algorithms

GA-P technique [8] is an hybrid between genetic algorithms and genetic programming, that was first used in symbolic regression problems. Individuals in GA-P have two parts: a tree based representation and a set of numerical parameters. Different from canonical GP, the terminal nodes of the tree are not used to store numbers but linguistic identifiers that are pointers to a chain of numerical constants (see Fig. 4.) The behavior of the GA-P algorithm is mainly due to the crossover operator. Both parts of the individual can be selected and crossed. We have employed GA-P algorithms in the identification and control of complex dynamical processes, and in classification problems [9]. In this work, we have modified the standard GA-P technique in the following way: we keep two GA parts for each GP tree. The first of the GA sections is used to store real constants that are used in the calculation of the GP expression. The second is a vector of real numbers containing as many parameters as features are used to describe a texture. We use this second GA component to keep information about the computational cost of calculating each feature. In the experiments performed in this work, this cost is uniform, so all the components of this second part are set to one. The GP part of an individual represents an arithmetic expression over the GA constants and the pre-calculated features that gives a real number as result. We use this expression, combined with a adequate wrapper function, to classify the texture samples.

5 Experiments

For testing the performance of the band ratioing technique combined with Gabor filtering, we have used the texture database defined in [2] which is composed of

[4] http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html

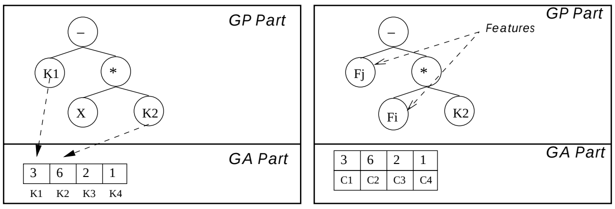

Fig. 4. Representation of a generic individual in GA-P techniques. GA-P individuals consist of two parts: a tree and set of numerical constants. Left:classical GA-P. Right: our implementation, including computational costs.

30 color textures taken from the Vistex database (Fig. 5). As in [3] we divide each 512x512 image into 64 disjunct images of 64x64 pixels each, which give us a total of 1920 texture samples. For each texture class, we randomly select a 80% of the samples for training, and the rest for testing purposes.

We have performed a number of experiments, applying a different preprocessing algorithm on each experiment, as showed in Table 1. In experiment *A*, grayscale features are computed by converting color images to gray using (6). In experiment *B*, we extract unichrome features from each color band of the images. Experiments *C*-*E* are performed by filtering different band ratios, and finally, in experiment *F* we extract unichrome features from two different band ratios, so the number of features increases by a factor of two.

Table 1. Experiments performed

Experiment	Preprocessing algorithm	Feature #
A	Grayscale	12
B	RGB	12×3
C	R/G	12
D	R/B	12
E	G/B	12
F	R/G + R/B	12+12

6 Results

For evaluation of the classification performance in each experiment, we have chosen to apply two classification schemes: a Knn classifier (supervised) and a GAP-based one (unsupervised).

Fig. 5. 30 color textures: Bark0, Bark4, Bark6, Bark8, Bark9, Brick1, Brick4, Brick5, Fabric0, Fabric4, Fabric7, Fabric9, Fabric11, Fabric13, Fabric16,Fabric17, Fabric18, Food0, Food2, Food5, Food8, Grass1, Sand0, Stone4, Tile1, Tile3, Tile7, Water6, Wood1, Wood2 (left/right,top/bottom)

6.1 Supervised classification results

For testing the performance of the feature sets showed in Table 1, using a supervised scheme, we have selected a Knn classifier, taking $K = 5$. This will allow us to directly compare our results with those obtained in [3]. As it can be seen in Table 2, experiment F surpasses the RGB features while maintaining low feature dimensionality. Although they are not showed here, we are even able to improve the performance of the *Complex Gabor Features* found in [3] with a lower number of features.

Table 2. Classification rates using a Knn classifier.

Experiment	Hits
A	90.71%
B	91.90%
C	87.38%
D	88.10%
E	83.57%
F	93.81%

6.2 Classifying textures with GA-P induced functions

Experiments. Our first goal was to obtain a GA-P function defined as $F : R^{24} \to R$ that could distinguish each one of the 30 textures. The set of parameters of the experiment is summarized in Table 3.

Table 3. Algorithm parameters

Objective	Classify 1500 samples
Function set	+, -, *, DIV, IFLTE, SIN, COS
Terminal set	F_i i: 1:36
Fitness cases	1500 texture samples
Raw fitness	Correctly classified
Classification error	1500 - previous value
Wrapper	+1:-1
Parameters	M= 10000 G=100
Mutation probability	0.05
Success predicate	1500 hits

We have reached a maximum classification success of 40% over the 1500 training samples, which is significantly poorer than the results obtained with other classification methods like Knn, so we tried to obtain 30 different GA-P-induced functions that allow us to distinguish each class from the remaining 29. Initially, we restrict our study to the experiments named A and F, in order to analyze the advantages of the use of *Band Ratioing* for extracting texture features over other techniques. The parameter set of this experiment was the same as the one described in Table 3.

This was a hard goal to reach. We obtained between 10 and 15 functions that solve the problem acceptably (90% success rate). But there was a significative difference between the results obtained on experiment A and F (grayscale features vs. band ratioing). Table 4 shows the best classification error (training and test) for each one of the 30 different classes. Each experiment was run up to 50 times to demonstrate that some classes were not separable by means of a GA-P system like this. Using band ratioing features we obtained directly 19 functions able to distinguish acceptably a class from the remaining 29. Gray level based data left to 9 functions. Table 5 shows the average number of acceptable classifiers obtained using data from experiments A and F.

Feature selection. Although the results obtained with our GA-P system are not good for classification purposes, there is some important knowledge we can extract from the GA-P expressions than classify correctly any set of samples. For example, the expression

Table 4. Misclassification values on experiment A and F. 50 runs

Function	Training (A)	Test(A)	Training(F)	Test (F)
F_1	32%	–	0%	3%
F_2	98%	–	62%	–
F_3	96%	–	40%	–
F_4	4%	11%	0%	3%
F_5	0%	5%	0%	4%
F_6	0%	6%	0%	7%
F_7	62%	–	24%	–
F_8	46%	–	6%	11%
F_9	4%	7%	0%	6%
F_{10}	96%	–	0%	1%
F_{11}	6%	7%	0%	4%
F_{12}	6%	11%	0%	9%
F_{13}	100%	–	18%	–
F_{14}	98%	–	4%	6%
F_{15}	100%	–	0%	12%
F_{16}	100%	–	0%	2%
F_{17}	20%	–	14%	–
F_{18}	38%	–	0%	3%
F_{19}	4%	9%	8%	11%
F_{20}	66%	–	66%	–
F_{21}	26%	–	52%	–
F_{22}	74%	–	10%	13%
F_{23}	2%	5%	10%	16%
F_{24}	44%	–	32%	–
F_{25}	2%	6%	0%	8%
F_{26}	24%	–	0%	5%
F_{27}	70%	–	4%	9%
F_{28}	92%	–	20%	–
F_{29}	98%	–	84%	–
F_{30}	100%	–	88%	–

Table 5. Average number of acceptable classifiers obtained from gray level data (A) and from band ratioing data (F)

Experiment	Functions
A	8.11
F	14.31

```
CLASS: 16
#Best individual:
((0.2718 × ((H + ((X - 0.2718) - ((W - O) + (D × (J -
- 0.635))))) - J)) - S) + (((((D - J) - U) × ((((D × (K - K)) ×
× ((J × R) + H)) - W) + ((0.3685 - J) + 0.4012))) × ((0.2503 +
+ 0.2503) + 0.3685)) - ((N + L) - H))

Missclassification error:0%
```

tell us that we can distinguish a sample of this class using only twelve of the twenty four features. Our purpose is to breed standard classification methods with this information in order to enhance their results.

7 Conclusions

In this paper, a new method for extracting color texture features have been presented. The results obtained using this new approach shows that using band ratios, we can get useful color information from textures while keeping a low number of features.

References

[1] Jain, A., Healy, G.: A multiscale representation including opponent color features for texture recognition. IEEE Transactions on Image Processing **7(1)** (1998) 124–128

[2] de Wouver, G.V.: Wavelets for Multiscale Texture Analysis. PhD thesis, University of Antwerp (Belgium) (1998)

[3] Palm, C., Keysers, D., Lehmann, T., Spitzer, K.: Gabor filtering of complex hue/saturation images for color texture classification. In: Proceedings of 5th Joint Conference on Information Science (JCIS2000), Atlantic City, USA. Volume 2. (2000) 45–49

[4] Haralick, R., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics **3(6)** (1973) 610–621

[5] Conners, R., McMillin, C.: Identifying and locating surface defects in wood: Part of an automated lumber processing system. IEEE Transactions on Pattern Analysis and Machine Intelligence **5(6)** (1983) 573–584

[6] Bovik, A., Clark, M.: Multichannel texture analysis using localized spatial filters. IEEE Transactions on Pattern Analysis and Machine Inteligence **12(1)** (1990) 55–73

[7] Kruizinga, P., Petkov, N., Grigorescu, S.: Comparison of texture features based on gabor filters. In: Proceedings of the 10th International Conference on Image Analysis and Processing, Venice, Italy. (1999) 142–147

[8] Howard, L.M., D'Angelo, D.J.: The GA–P: A genetic algorithm and genetic programming hybrid. IEEE Expert **10** (1995) 11–15

[9] García, S., González, F., Sánchez, L.: Evolving fuzzy rule based classifiers with GAP: A grammatical approach. In Poli, R., Nordin, P., Langdon, W.B., Fogarty, T.C., eds.: Genetic Programming, Proceedings of EuroGP'99. Volume 1598 of LNCS., Goteborg, Sweden, Springer-Verlag (1999) 203–210