COMPARATIVE STUDY OF TEXTURE FEATURE FOR ROTATION INVARIANT RECOGNITION

A. Hafiane, C. Rosenberger, and H. Laurent

Laboratoire Vision Robotique ENSI de Bourges-Université d'Orléans
88, Boulvard Lahitolle 18020 Bourges cedex

ABSTRACT

Human visual system easily and rapidly recognizes a scene or image under different affine transformations, which is not the true for the machine. Rotation is more complex than translation and engenders more difficulties in analysis. This paper address evaluation and comparison of texture descriptors, particularly Local Relational String, under rotation effects. Many methods are invariant for geometric transformation, but this is not sufficient to handle the classification problem. We show in this study, when training samples represent a large range of rotated textures, methods with high discriminative properties leads to a very good classification rate despite their no invariance for rotation.

1. INTRODUCTION

Texture analysis is an important problem since it conditions the quality of segmentation and interpretation in lots of applications such as in the textile industry or for satellite imaging.

There exist many approaches for texture description and analysis [20], which can be divided into four categories: statistical methods, structural methods, model based-methods and transform-base methods. Texture characterisation and similarity measures represent the main issues in texture classification [14]. There is no obvious common definition of textures it can be defined as spatial arrangement of textons, distribution of patterns or specific spatial frequencies. This implies no common features that can be defined for textures. For example, the popular Grey Level Cooccurrence Matrix [11] and derivatives [7][8] define the joint probability that a given grey level occur in specific distance in the image. The geometrical methods study the spatial distribution of textures primitives and are often used for periodic structures [25][23][24][1]. Texture is also considered as model such as Markov Random Fields (MRF) where the texture is defined as realisation of random field with spatial context [3][6]. Other techniques suppose that the human visual system transform a retinal image into spatial frequency representation [2]. Gabor filters [21][4] handle this property thereby using a bank of filters. Recent works combine structural and statistical properties [17][13][10] showing a high classification performances. The texture appearance is considered as a result of dominant neighbourhood properties, and empirical distribution of intrinsic primitives. For this hybrid approach, geometrical or shape properties and spatial distribution patterns become very important.

In this communication, we propose a comparative study of texture features. The aim is to study their ability to recognize multi-oriented textures by following a rigorous protocol. We verify the performances of Local Relational String [10] under different conditions, which are not proposed in the original paper. We test also the fusion of all descriptors and the influence of the feature reduction in the classification by PCA (principal component analysis) procedure.

In the next section, we detail computation of the texture features. Section 3 is devoted to the learning and recognition method we use to quantify the reliability of the previous features. Section 4 presents the protocol and the chosen database. Experiments and results are presented in section 5. We conclude in section 6.

2. TEXTURE ATTRIBUTES ANALYSIS

We have chosen some of the most efficient and general (computed without any *a priori* knowledge) texture attributes [5]:

- attributes from the cooccurrence matrix [11] denoted COOC, which is computed with a distance of one pixel and four orientations : 0 , $\pi/4$, $\pi/2$, $3\pi/4$. We select the following measures: mean, variance, contrast, energy, entropy, homogeneity, correlation, inertia, inverse moment, shade, prominence.
- attributes from local histograms (2 parameters) [15] denoted HIST: Two attributes are computed from local histograms of an image of size $N \times N$.
  1. Mean modulus:
  $$MOD = \frac{\sum_{i=1}^{NH} ||H_i||}{NH}$$
  where $||H_i|| = \sum_{j=1}^{NG} (NG - \frac{n^2}{NG}) log(r_i)$, r_i is the number of pixels in the window having a grey level i and NH is the number of histograms computed in the image.
  2. Mean phase :
  $$PHASE == \frac{\sum_{i=1}^{NH} phase(H_i)}{NH}$$
  where $phase(H_i) = \frac{2\pi j}{NG}$ with $j = arg\ max_{i=1,NG}(r_i)$ NG is the number of grey levels in the original image.
- attributes from local extrema (4 parameters) [9] denoted EXT:
  This method permits to evaluate the relative frequency of luminance gradient. It consists in counting the number of local extrema in direction θ on a distance d. The algorithm has two steps:
  – The first step corresponds to image smoothing in order to suppress luminance peaks having a low variation. We define I_k the grey level of the pixel k, $\tilde{I}_k$ the smoothed value and S the threshold used for the detection of extrema. The smoothed image is computed as follows:

$$\tilde{I} = \begin{cases} if\ \tilde{I}_k \leq I_{k+1} - S/2 \\ then\ \tilde{I}_{k+1} = I_{k+1} - S/2 \\ if\ I_{k+1} - S/2 < \tilde{I}_k \leq I_{k+1} + S/2 \\ then\ \tilde{I}_{k+1} = I_{k+1} \\ if\ I_{k+1} + S/2 < \tilde{I}_k \\ then\ \tilde{I}_{k+1} = I_{k+1} + S/2 \end{cases}$$

 - The second step computes the mean number of extrema given different directions θ (in this case $\theta = k.\frac{pi}{4},\ k = 0,3$), a distance d and a threshold S.

- attributes from the curvilinear integral (4 parameters) [9] denoted CURV: The computation of the curvilinear integral is realized by following a line L_i:

$$\int_{L_i} \sqrt{\lambda_1 dx^2 + \lambda_2^2 dy^2 + dI^2_{(x,y)}}$$

where $\lambda_1 = \frac{NLIG}{NG}, \lambda_2 = \frac{NCOL}{NG}$, $NLIG$, $NCOL$ are the numbers of lines and columns in the original image.
We compute this descriptor in 4 directions ($\theta = k.\frac{\pi}{4},\ k = 0,3$) for a given distance d.

- normalized autocovariance function (32 parameters)[16]: We present a method to compute the 1D normalized autocovariance of a texture from its 2D normalized autocovariance function. The resulting function is called F and is computed as:

$$F(r) = \frac{1}{\pi r} \sum_{(i,j)\in\hat{C}_r} F\tilde{A}C(i,j) \quad \forall r > 0$$

where $\hat{C}_r = \{(i,j)/i > 0,\ \sqrt{i^2+j^2} = r\}$ is the set of points on the semi-circle of radius r. Let be $m = \sqrt{i^2+j^2}$, $F\tilde{A}C$ is defined as:

$$F\tilde{A}C(i,j) = \begin{cases} FAC(i,j) & if\ m \in \mathbb{N} \\ \frac{\sum_{k=1}^{v} d(s^{(k)},m)FAC(s^{(k)})}{\sum_{k=1}^{v} d(s^{(k)},m)} & otherwise \end{cases}$$

The first term is the 2D normalized autocovariance function defined as :

$$FAC(i,j) = \frac{1/N_\Delta \sum_{s=(k,l)\in R_\Delta} I(s)I(k+i,l+j)}{1/N \sum_{k,l} I^2(k,l)}$$

where N_Δ and R_Δ are respectively the number of points and the region on which the product $I(s)I(k+i,l+j)$ is computed. N is the total number of pixels, I is the luminance function (its mean is equal to zero), i and j are the horizontal and vertical displacement.

- attributes from Gabor filters described by the following function:

$$G_{\lambda,\theta,\varphi}(x,y) = e^{-\frac{1}{2}(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2})} \cos(2\pi\frac{x'}{\lambda} + \varphi)$$

$$x' = x\cos(\theta) + y\sin(\theta)$$
$$y' = -x\sin(\theta) + y\cos(\theta)$$

where σ is the standard deviation. θ express the orientation of the filter. The parameter $\frac{1}{\lambda}$ determines the spatial frequency. φ represents the phase. We have chosen four angles (θ : 0, $\pi/4$, $\pi/2$, $3\pi/4$) and three frequencies (λ : 2, 4, 8 pixels). We kept φ=0. The image is convolved with a bank of filters with different values of λ and θ. The energy function is computed for each parameter from the convolved image which yields 12 features. The filter size is 17×17 pixels.

- attributes from Local Relational String (LRS) [10] Let be g_0 a central pixel and $\Omega = \{g_1, g_2, g_3, g_4\}$ the set of neighbours. The relative relation $\mathscr{X}$ between two pixels are defined over the set $\mathscr{S}$

$$\mathscr{S} = \{(g_0, g_i) \in \mathscr{I} \ \mid\ \exists r_i \in \mathscr{R}, r_i = \mathscr{X}(g_0, g_i)\}\ g_i \in \Omega$$

where $\mathscr{I}$ is a set of pixels and $\mathscr{R}$ represents a set of linguistic variables (equal to , less than, greater than).

$$\mathscr{R} = \{<, >, =\}$$

LRS is defined as an ordered symbolic string.

$$LRS : r_1 r_2 r_3 r_4$$

Histogram is computed over the transformed image to describe the texture. LRS handle also the scale problem by extending the relations to the far neighbours yielding a histgram for each neighbourhood. We selected 3 neighbourhood sets with distance 3 pixels from the central pixel.

3. TRAINING AND RECOGNITION METHOD

Suppose we have a training set $\{\mathbf{x}_i, \mathbf{y}_i\}$ where $\mathbf{x}_i$ is the invariant descriptors vector described in the previous section. $\mathbf{x}_i$ is composed of values corresponding to each descriptor computed on the face and $\mathbf{y}_i$ the individual. For problems with two classes, with the classes $y_i \in \{-1, 1\}$, a support vector machine [22][18] implements the following algorithm. First, the training points $\{\mathbf{x}_i\}$ are projected into a space $\mathscr{H}$ (of possibly infinite dimension) by means of a function $\Phi(\cdot)$. The second step consists in finding a decision hyperplane in this space. The criterion for optimality is defined shortly afterwards. Note that for the same training set, different transformations $\Phi(\cdot)$ may lead to different decision functions.

A transformation is achieved in an implicit manner using a kernel $K(\cdot,\cdot)$ and consequently the decision function can be defined as :

$$f(\mathbf{x}) = \langle w, \Phi(\mathbf{x})\rangle + b = \sum_{i=1}^{\ell} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

with $\alpha_i^* \in R$. The values w and b are the parameters defining the linear decision hyperplane. We use in the proposed system a radial basis function as kernel function :

$$K(\mathbf{u}, \mathbf{v}) = e^{-\gamma * \|\mathbf{u}-\mathbf{v}\|^2}$$

In SVMs, the optimality criterion to maximize is the margin, that is to say, the distance between the hyperplane and the nearest point $\Phi(\mathbf{x}_i)$ of the training set. The α_i^* which optimise this criterion are obtained by solving the following problem :

$$\begin{cases} \max_{\alpha_i} \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i K(\mathbf{x}_i, \mathbf{x}_j y_j) \\ \text{with constraints,} \\ 0 \leq \alpha_i \leq C\,, \\ \sum_{i=1}^{\ell} \alpha_i y_i = 0\,. \end{cases}$$

where C is a penalization coefficient for data points located in or beyond the margin and provides a compromise between their numbers and the width of the margin (for this study $C = 1$). Originally, SVMs have essentially been developed for the two classes problems. However, several approaches can be used for extending SVMs to multiclass problems. The method we use in this communication is called *one against one*. Instead of learning N decision functions, each class is discriminated here from another one.

4. TEXTURE DATABASE

The performance measure does not depend only on classifier, but also the experimental platforms including reference algorithms and test images. The subjective nature of many experimental evaluations have engendered critic about the validity of tests [11]. To overcome this problem some research teams have built test frameworks available for public [19] considered as benchmark for textures experiments and evaluations. Among those frameworks Meastex[1] provides a significant evaluation system. MeasTex software is an open source containing image database, quantitative measurement framework for texture analysis algorithms, and implementation of major texture classification paradigms. Nevertheless, the images base of this benchmark is poor in term of quality and quantity comparing to the new ones. The Outex [2] [12] textures base has recently became regarded as the best available one. It presents many problems related to textures analysis and recognition. Outex is organised as train/test set within several categories each one address a particular textures issues. For instance illumination, resolution, geometric transformation (by acquisition), etc. Outex use a predetermined train/test samples. In our work we pick randomly the samples from each class and testing the impact of the number of training set over the performances. As described previously we use SVM to measure the classification rate.

5. EXPERIMENTAL RESULTS

We perform different tests for comparison of the selected features.

5.1 Test1

Here experiments are conducted over Outex_TC_00000 category. For each class, there are 20 monochrome images (128×128) with incandescent constant illumination and a spatial resolution 100dpi. Figure 2 shows the classification rate versus training rate. We can notice at 10% training images, Gabor filters and LRS present a good classification performances greater than 80 %. The evolution of curves shows that LRS tend to be more efficient than the other methods followed by Gabor filters. Local extrema method has a stable variation, stay around 70%. The Cooccurrence Matrix start at

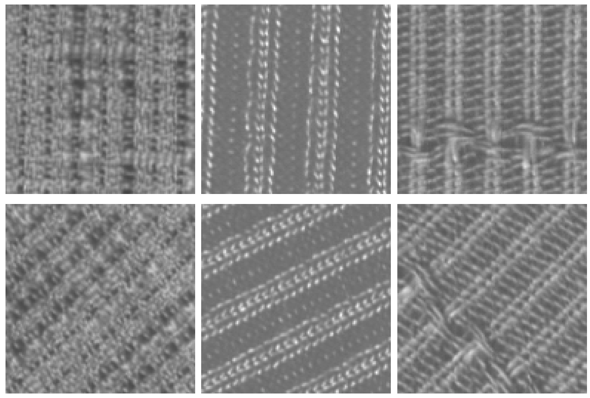

Figure 1: Samples from Outex_TC_00010 category

60% to finish around 70%, it needs more trainings than local extrema to reach an this result. The autocovariance technique does better than local histograms and curvilinear integral, but its classification rate is far from the first four methods.

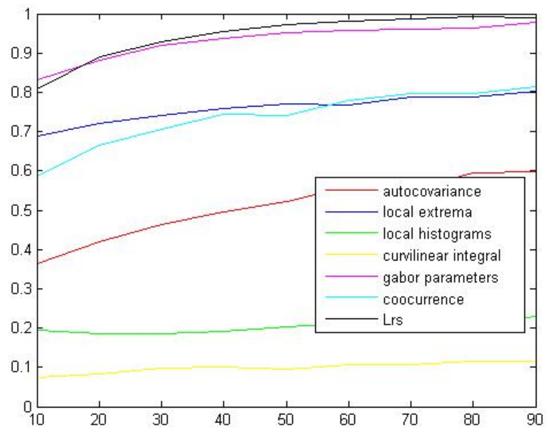

Figure 2: Recognition results on the first database or different percentages of images used for the learning step

5.2 Test2

We have tested algorithms under different orientations. Indeed, the Outex_TC_00010 category offers a set of a rotated host textures with angles: $5^o, 10^o, 15^o, 30^o, 45^o, 60^o, 75^o, 90^o$. Each rotation is performed for 24 classes yielding 8×20 images per class plus the original textures, therefore each class contain 180 images. As the previous experiences, samples are chosen randomly from each class. For instance, 10% of training set gives 18 images with different rotations of each class. In figure 3, we notice the improvement in performances, this is due to the number of the training set. For example, at 10% in Outex_TC_00000 yield 2 images per class. Further, more methods such as Gabor filters and cooccurrence matrix take into account the rotation effect. LRS has not this particularity, but its high discriminative properties help to reach good performances. Examining the evolution of curves, at 20% of training set there are a rapid improve-

[1]http://www.cssip.elec.uq.edu.au/guy/meastex/meastex.html
[2]http://www.outex.oulu.fi

ment in the best three methods. LRS yields 99,99% with 40% of the training database. No significant changes for the other methods, curvilinear integral and local histograms show the poorest results.

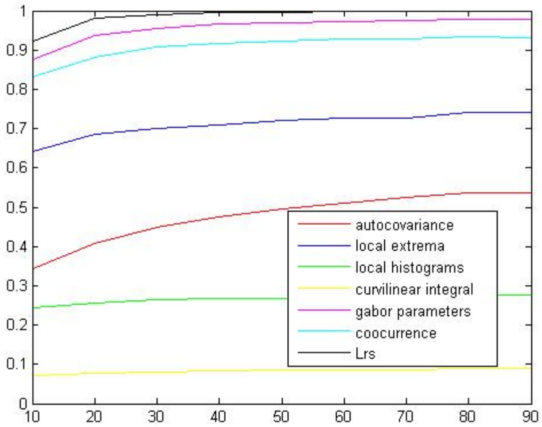

Figure 3: Recognition results on the second database or different percentages of images used for the learning step

5.3 Test3

The aim here is to test the contribution of fusion of features. All features described before are concatenated to build one feature vector. Figure 4 represents 3 results obtained from Outex_TC_00010 category. The best method (LRS) is compared with fusion technique and fusion combined with principal component analysis (PCA). As shown in the figure 4, the fusion procedure is not able to improve the performances. The different natures of descriptors decrease the performance even using PCA which yield worse result.

6. CONCLUSION

In this paper we presented the evaluation performance of several methods for texture classification. The rotation problem is addressed here thereby using 8 different angles with step of 15^o, from 0^o up to 90. LRS, Gabor filters and cooccurrence matrix showed the best performances among the 7 compared descriptors. the fusion of features do not bring out any improvement in performances, even using pca to reduce redundancy. This study shows interesting results for the tested descriptors. It can be used as benchmark to compare other methods of texture classification or recognition algorithms.

REFERENCES

[1] D. Blostein and N. Ahuja. Shape from texture: Integrating texture-element extraction and surface estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1233–1251, 1989.

[2] F. W. Campbell and J. G. Robson. Application of fourier analysis to the visibility of gratings. *Journal Physiol.*, 197:551–566, 1968.

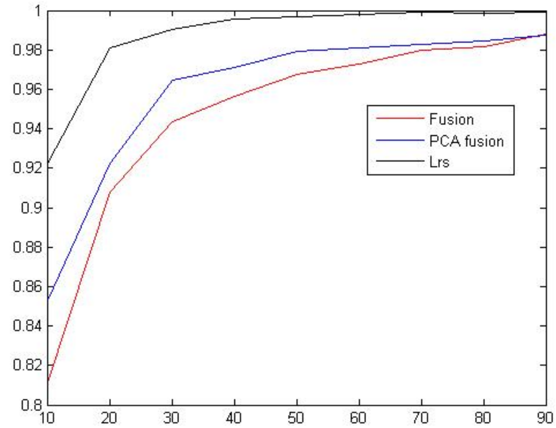

Figure 4: Recognition results on the second database: benefit of the fusion and variable selection

[3] R. Chellappa and S. Chatterjee. Classification of textures using gaussian markov random fields. *IEEE Trans. Acoustics Speech Signal Process*, 33:959–963, August 1985.

[4] M. Clark and A. C. Bovik. Texture segmentation using gabor modulation/demodulation. *Pattern Recognition Letters*, 6(4):261–267, 1987.

[5] J. P. Cocquerez and S. Philipp. *Analyse d'images : filtrage et segmentation*. Masson, 1995.

[6] M.L. Comer and E.J. Delp. Segmentation of textured images using a multiresolution gaussian autoregressive model. *IEEE Transactions on Image Processing*, 8(3):408–420, 1999.

[7] L. S. Davis, M. Clearman, and J. K. Aggarwal. An empirical evaluation of generalized cooccurrence matrices. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 3(2):214–221, 1981.

[8] C. C. Gotlieb and H. E. Kreyszig. Texture descriptors based on co-occurrence matrices. *Comput. Vision Graph. Image Process.*, 51(1):70–86, 1990.

[9] V. Haese-Coat and J. Ronsin. évaluation et comparaison de méthodes de segmentation en imagerie spot. Technical report, INSA RENNES, 1992.

[10] A. Hafiane and B. Zavidovique. Local relational string for textures classification. In *IEEE ICIP*, Atlanta, USA, 2006.

[11] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Trans. Sys. Man. Cybern*, 3(6):610–621, 1973.

[12] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinene. Outex - a new framework for empirical evaluation of texture analysis algorithms. In *Proc. 16th Intl. Conf. Pattern Recognition 2002*, 2002.

[13] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern*

Anal. Mach. Intell., 24(7):971–987, 2002.

[14] T. Randen and J. H. Husoy. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310, 1999.

[15] J. Ronsin, D. Barba, and S. Raboisson. Comparaison between cooccurrence matrices, local histogramm and curvilinear integration for texture characterization. In *SPIE*, volume 596, pages 98–111, 1985.

[16] C. Rosenberger, K. Chehdi, C. Cariou, and J. M. Ogier. Texture analysis of an image by using a rotation invariant model. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, PHOENIX, 1999.

[17] J.A. Rushing, H.S. Ranganath, T.H. Hinke, and S.J. Graves. Using association rules as texture features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):845–858, 2001.

[18] B. Scholkopf and A. Smola. Leaning with kernels. *MIT Press*, 2001.

[19] G. Smith and I. Burns. Measuring texture classification algorithms. *Pattern Recognition Letters*, 18(14):1495–1501, 1997.

[20] M. Tuceryan and A. K. Jain. Texture analysis. *Handbook of pattern recognition & computer vision*, pages 235–276, 1993.

[21] M. R. Turner. Texture discrimination by gabor functions. *Biological Cybernetics*, 55:71–82, 1986.

[22] V. Vapnik. Statistical learning theory. *Wiley*, 1998.

[23] F.M. Vilnrotter and R. Nevatia. Structural texture analysis applications. In *DARPA82*, pages 243–252, 1982.

[24] H. Voorhees and T. Poggio. Detecting textons and texture boundaries in natural images. In *In Proceedings of the First International Conference on Computer Vision*, pages 250–258, 1987.

[25] S. W. Zucker. Toward a model of texture. *Computer Graphics and Image Processing*, 5:190–202, 1976.