

Reconnaissance invariante par rotation de textures par des chaînes de relations locales

Adel HAFIANE¹, Christophe ROSENBERGER¹, Bertrand ZAVIDOVIQUE²

¹Laboratoire Vision et Robotique
ENSI de Bourges - Université d'Orléans, 88 boulevard Lahitolle, 18020 Bourges Cedex France

²Institut d'Electronique Fondamentale
Batiment 220, Université de Paris-sud, 91405 Orsay Cedex France
adel.hafiane@ensi-bourges.fr, christophe.rosenberger@ensi-bourges.fr,
bertrand.zavidovique@ief.u-psud.fr

Résumé – Les structures locales constituent un élément essentiel dans la description de textures. L'extraction d'une information locale pertinente permet d'augmenter les performances de reconnaissance de texture. Les transformations géométriques affectent en général les structures locales ce qui rend les techniques basées sur ce type d'information vulnérable. Dans cet article, nous nous intéressons à ce problème et plus particulièrement celui de la rotation. Nous avons proposé récemment une méthode efficace de caractérisation de textures qui a prouvé une bonne efficacité dans la classification de textures. En revanche, cette méthode n'est pas invariante à la rotation. Le but de ce travail est de remédier à ce problème en utilisant les techniques d'apprentissage supervisé. La méthode du Séparateur à Vaste Marge a été employée à cette fin. Les expériences effectuées ont montré que l'apprentissage permet de réduire les erreurs de reconnaissance et par conséquent augmenter les performances du système.

Abstract – The local structures provide a useful information for texture description. Retrieving relevant local information allow to improve performances for texture recognition. Geometrical transformations have an impact in local structures which became a problem for techniques based on such type of information. In this paper, we are interesting by the rotation problem. We have proposed recently an efficient method for texture characterization which showed good performances for textures classification. However, this method is not rotation invariant. The aim of this work is to overcome this problem by using some supervised learning techniques. Support Vector Machine is used for that purpose. Experiences showed that the learning method reduced errors of recognition, therefore it improves performances.

1 Introduction

L'analyse de textures a été abordée d'une manière intensive ces dernières années en vue d'obtenir un système de vision intelligent comparable à celui de l'être humain. Les capacités de ce dernier à discerner les textures restent très supérieures à celles d'une machine. Cependant, plusieurs méthodes permettent de caractériser automatiquement des textures et de résoudre des problèmes liés à ce phénomène tels que l'extraction ou la reconnaissance d'objets, la reconstruction 3-D ou la perception de la déformation des surfaces. Les performances de la description d'une texture sont alors souvent jugées via celles de la perception, elle-même fondée sur la classification de textures et la segmentation. Une caractéristique efficace présente des propriétés discriminantes élevées. Il n'existe pas une définition précise de la texture au-delà du fait qu'elle restitue l'état de surface, mais certaines propriétés en ont été à l'usage vérifiées pertinentes. On retiendra deux d'entre elles [4] :

1. répétition régulière ou aléatoire d'une (famille de) primitive(s) de base.
2. distribution spatiale des niveaux de gris, incluant donc des notions fréquentielles.

Des travaux récents [6, 7] ont montré l'intérêt des propriétés locales dans la description de la texture. Ces méthodes procèdent généralement en deux étapes : extraction des propriétés locales et mesure de leur répartition dans l'image. Dans cet article, nous étudions un nouveau opérateur basé sur une représentation symbolique entre un pixel et son voisinage [3]. Nous montrons son intérêt pour la reconnaissance invariante de textures sur une base significative d'images et nous comparons cette approche avec d'autres descripteurs de la littérature.

Cet article est organisé comme suit : La deuxième section décrit le calcul des caractéristiques. La troisième section présente les propriétés du descripteur CRL. La section quatre décrit la technique d'apprentissage. Les expériences et les résultats sont présentés dans la cinquième section. La sixième section présente nos conclusions sur cette étude.

2 Caractérisation

La méthode développée repose sur les propriétés locales dans le voisinage d'un pixel. Notre but est d'exhiber des caractéristiques qui ne sont pas explicites dans l'image originale. En d'autres termes l'objectif est de trouver une transformation dans laquelle on extrait des primitives indiquant un certain aspect des structures locales d'une ma-

nière abstraite. Ceci est réalisé par le passage d'une représentation de l'image de valeurs numériques à une représentation symbolique où chaque symbole est défini par un code.

Soit un voisinage de 4-connexité $\Omega = \{g_1, g_2, g_3, g_4\}$ comme le montre la figure 1 (a). Pour extraire des relations liant g_0 à ses voisins spatiaux, on définit une fonction \mathcal{Z} pour deux pixels sur un ensemble \mathcal{S} comme :

$$\mathcal{S} = \{(g_0, g_i) \in \mathcal{I} \mid \exists r_i \in \mathcal{R}, r_i = \mathcal{Z}(g_0, g_i)\}; g_i \in \Omega \quad (1)$$

où \mathcal{I} représente l'image et \mathcal{R} correspond à un ensemble de variables linguistiques (égal, inférieur, supérieur) :

$$\mathcal{R} = \{<, >, =\} \quad (2)$$

On définit la relation entre un pixel et ses voisins comme une chaîne symbolique ordonnée appelée *Chaîne de Relations Locales* (CRL) :

$$CRL : r_1 r_2 r_3 r_4 \quad (3)$$

L'exemple de la figure 1 (b) produit la chaîne suivante :

$$>>=<$$

Cette opération est appliquée sur toute l'image produisant un codage particulier dans chaque point. Afin de caractériser la texture, on mesure la distribution des CRL. Pour cela on utilise un histogramme. En effet, nous avons quatre comparaisons par trois relations donc le nombre maximal de cases est $3^4 = 81$.

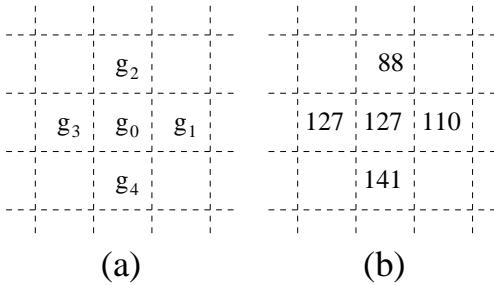


FIG. 1 – Voisinage à 4-connexités

Cependant, une petite taille de voisinage n'est pas suffisante pour capter une information complète. De plus, les variations d'échelle peuvent affecter les structures locales de la texture. Il est donc nécessaire d'étendre le traitement à des voisins spatialement lointains. Impliquer plus de points dans la structure élémentaire basée sur 4-connexité augmente la complexité des calculs. On opte pour des CRL séparées pour chaque ensemble de points. Soit $\Omega^d = \{g_1^d, g_2^d, g_3^d, g_4^d\}$ un ensemble de points à une distance d . La formule générale de CRL est donnée par :

$$r_i^d = \mathcal{Z}(g_0, g_i^d) \quad ; \quad i = 1, 2, 3, 4. \quad (4)$$

Etant donné Ω^d , on définit la texture comme une distribution de CRL :

$$H_{CRL} = (H_{CRL}^1, H_{CRL}^2, \dots, H_{CRL}^d) \quad (5)$$

La figure 3 montre la structure de voisinage que nous utilisons. CRL est appliquée sur un voisinage de 4 pixels

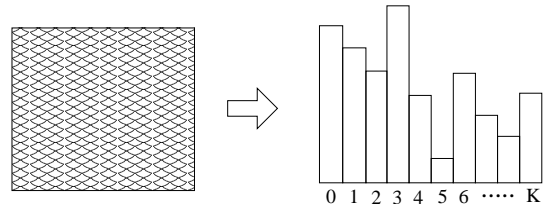


FIG. 2 – De la texture vers l'histogramme des CRL

selon la verticale et l'horizontale (figure 3 (a)) ou selon les deux diagonales (figure 3 (b)). Les points de même couleur dans la figure 3 constituent un voisinage. L'opérateur CRL produit une image de motifs pour chaque voisinage. En l'occurrence, on obtient 6 histogrammes pour cette structure. Ces histogrammes sont concaténés ensemble, la distribution finale est de taille de 6×81 éléments. La figure 2 illustre le principe général de cette description. Il est possible d'étendre la taille de voisinage à des pixels plus lointains. En augmentant la distance d , les performances s'améliorent jusqu'à une valeur limite de d . Il est évident que le calcul devient plus gourmand avec l'augmentation de la taille de voisinage. En étudiant cette particularité, nous avons remarqué que la distance $d = 3$ constitue un bon compromis entre la complexité du calcul et l'efficacité du descripteur.

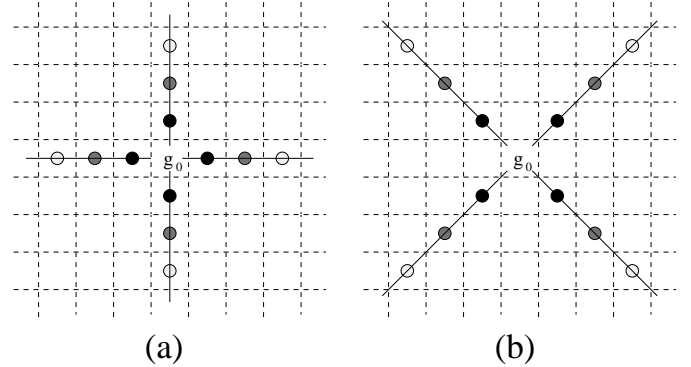


FIG. 3 – Voisinage sur une distance de 3 pixels

3 Propriétés

Parmi les propriétés les plus importantes de CRL est invariant par rapport aux changements monotones et uniformes d'éclairage. Supposant qu'une partie de l'image à subir un changement d'éclairage. Supposant que les pixels reçoivent un gain \mathcal{A} en intensité lumineuse :

$$g'_i = \mathcal{A} \cdot g_i \quad (6)$$

On peut vérifier que :

$$\mathcal{Z}(g_i, g_j) \equiv \mathcal{Z}(g'_i, g'_j) \quad (7)$$

Les relations relatives restent inchangées et donc CRL ne varie pas avec ce changement d'éclairage. Cette propriété est importante car elle permet d'éliminer une composante souvent source d'erreur pour la reconnaissance de textures. En effet, comme la texture est indépendante de

la couleur ou de l'éclairage, une structure robuste permet d'augmenter les propriétés discriminatoires d'un descripteur.

La deuxième propriété importante est la transformation géométrique particulièrement la rotation. Cette dernière pose en général plus de problème par rapport aux autres transformations. La CRL proposé ne prend pas en compte cette particularité parce que une rotation locale des pixels voisins change l'ordre dans la chaîne et par conséquent on obtient des motifs différents. Pour remédier à cela nous avons choisi d'apprendre au système à reconnaître les rotations, en utilisant des exemples de rotation. Pour cela nous avons employé un Séparateur à Vaste Marge (SVM)¹.

4 Apprentissage supervisé

L'apprentissage supervisé ici est un problème multiclassés avec un nombre de classes dépendant de la quantité d'indices de qualité. Un problème multiclassés est résolu par une polychotomie basée sur une approche *un-contre-un* [5]. L'algorithme d'apprentissage est ici Séparateur à Vaste Marge à n classes. Ainsi, nous recherchons un hyperplan dans un espace \mathcal{H} défini comme :

$$f(x) = \sum_{i=1}^{\ell} \alpha_i^* y_i K(x_i, x) + b \quad (8)$$

qui maximise la marge entre l'hyperplan et les données projetées x_i dans \mathcal{H} . Alors, les α_i^* sont les solutions du problème d'optimisation suivant :

$$\max_{\alpha_i} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (K(x_i, x_j) + \frac{1}{C} \delta_{i,j}) \quad (9)$$

avec $\sum_i \alpha_i y_i = 0, \quad 0 \leq \alpha_i$

où K est le noyau associé à \mathcal{H} , $\delta_{i,j}$ est le symbole de Kronecker et C un paramètre de compromis entre la largeur de la marge et le nombre d'exemples situés au delà de la marge.

5 Résultats expérimentaux

Nous avons utilisé des textures de la base Outex². Elle contient une large variété de textures organisées en plusieurs catégories selon des critères liés aux problèmes à résoudre. On y trouve par exemple des textures prises sous différentes conditions d'illumination pour tester l'invariance aux changements d'éclairage, des prises de vue avec rotation ou translation en vue de l'étude d'invariance aux changements géométriques.

Deux sous-bases contenant 24 classes de textures ont été utilisées. La première est de la catégorie Outex_TC_00000. La figure 4 illustre des échantillons de cette catégorie. Chaque classe comprend 20 images monochromes de taille 128×128 pixels avec une illumination incandescente constante et une résolution de 100dpi.

La deuxième sous base Outex_TC_00010 comporte des images de la catégorie Outex_TC_00000 ayant subi diffé-

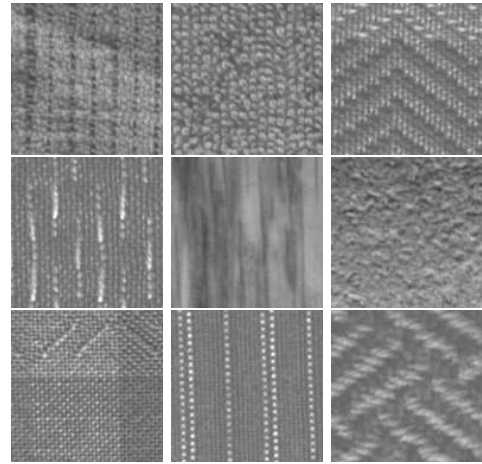


FIG. 4 – Quelques textures de la base Outex.

rentes rotations selon plusieurs angles. Chaque classe comporte 20 images avec les rotations suivantes : $0^\circ, 5^\circ, 10^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ$. Il en résulte 180 images par classe. La figure 5 présente quelques exemples de texture de la deuxième base. Cela augmente le nombre d'échantillons et d'exemples à tester par rapport à la catégorie Outex_TC_00000. La taille, la résolution et les conditions d'éclairage restent inchangées par rapport à la catégorie précédente.

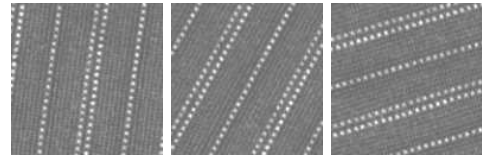


FIG. 5 – Échantillons d'une texture avec différentes rotations

La comparaison est effectuée pour différents descripteurs de texture de la littérature :

- paramètres de la matrice de cooccurrence [4] : 13 paramètres,
- extrema locaux [8] : 2 paramètres,
- intégrale curviligne [2] : 4 paramètres,
- histogrammes locaux [2] : 4 paramètres,
- autocovariance normalisée [9] : 32 paramètres,
- paramètres de Gabor [1] : 12 paramètres.

Afin de tester l'efficacité des descripteurs proposés, nous avons fait varier le nombre des échantillons utilisés lors de l'apprentissage par le SVM. Nous avons réalisé 20 essais pour le tirage aléatoire de la base d'apprentissage et de test afin de refléter l'efficacité d'un descripteur.

Les résultats obtenus pour la catégorie Outex_TC_00000 sont présentés dans la figure 6. On constate que CRL donne les meilleurs taux suivie par les filtres de Gabor puis les extrema locaux ainsi que GLCM. Les taux de reconnaissance des autres méthodes sont faibles.

Sur la base Outex_TC_00010 (problème de rotation) les résultats sont illustrés par la figure 7. En prenant 36 échantillons par texture sur les 180 disponibles pour la phase d'apprentissage, l'on obtient un taux de reconnaissance de 98% des autres textures (apparaissant à différentes orientations lors de l'acquisition).

¹<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

²<http://www.outex.oulu.fi>

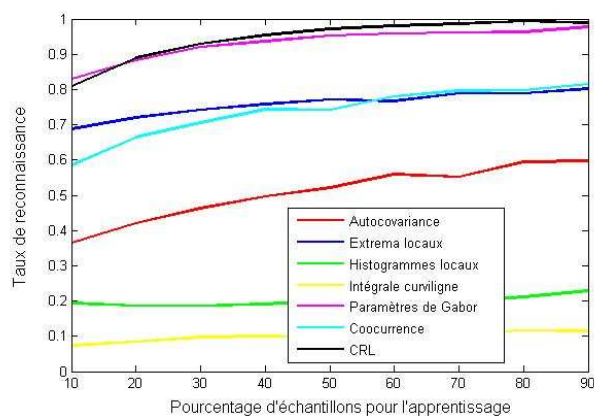


FIG. 6 – Résultats de reconnaissance sans rotation

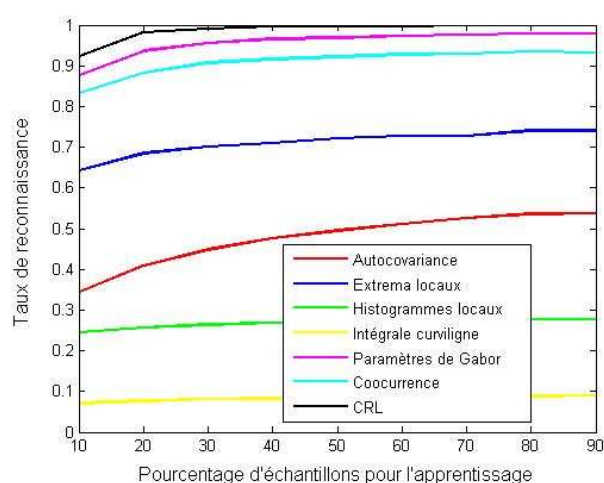


FIG. 7 – Résultats de reconnaissance avec rotation

À présent, on souhaite étudier les performances des descripteurs en les fusionnant ensemble. Pour cela, nous avons concaténé dans un seul vecteur les 7 descripteurs présentés précédemment. La figure 8 montre les résultats obtenus pour la catégorie Outex_TC_00010. On remarque une dégradation des performances pour la fusion comparative au meilleur résultat sans fusion. Une légère amélioration est apportée par l'analyse en composantes principales sans atteindre un taux de reconnaissance comparable à celui de CRL.

6 Conclusion

Cette étude a mis en évidence l'efficacité relative de ses descripteurs, ce qui est une information primordiale en vue d'une application (indexation, segmentation, suivi,...). Les tests effectués sur la base Outex ont montré que 20% d'échantillons de chaque classe permet d'atteindre des taux de reconnaissance élevés. Le problème de rotation a été abordé par un apprentissage supervisé. Les résultats obtenus par la technique de SVM sont satisfaisants compte tenu du taux de classification. Nous avons constaté également que la fusion de plusieurs descripteurs de différentes natures ne mène pas forcément à des meilleures performances.

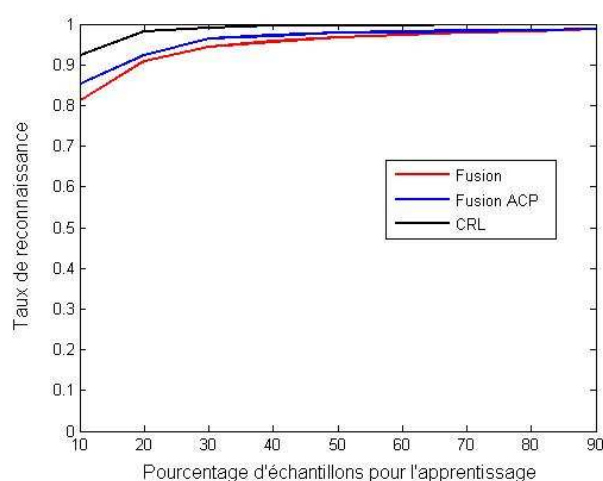


FIG. 8 – Résultats de reconnaissance par fusion des descripteurs.

Références

- [1] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Journal of Biological Cybernetics*, 61 :102–113, 1989.
- [2] V. Haese-Coat and J. Ronsin. évaluation et comparaison de méthodes de segmentation en imagerie spot. Technical report, INSA RENNES, 1992.
- [3] A. Hafiane and B. Zavidovique. Local relational string for textures classification. In *IEEE International Conference on Image Processing*, pages 2157–2160, October 2006.
- [4] R.M. Haralick, K. Shanmungan, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6) :610–621, November 1973.
- [5] C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13 :415–425, 2002.
- [6] 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.
- [7] T. Ojala, K. Valkealahti, E. Oja, and M. Pietikäinen. Texture discrimination with multidimensional distributions of signed gray-level differences. *Pattern Recognition*, 34(3) :727–739, 2001.
- [8] J. Ronsin, D. Barba, and S. Raboisson. Comparaison between cooccurrence matrices, local histogram and curvilinear integration for texture characterization. In *SPIE*, volume 596, pages 98–111, 1985.
- [9] 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.