

Comparing String Representations and Distances in a Natural Images Classification Task

Julien Ros¹, Christophe Laurent¹, Jean-Michel Jolion², and Isabelle Simand²

¹ France Telecom R&D - TECH/IRIS,
4, rue du Clos Courtel, 35512 Cesson Sévigné Cedex - France
{julien.ros, christophe2.laurent}@francetelecom.com

² LIRIS, FRE CNRS 3672 INSA,
Bât. J. Verne, INSA Lyon,
69621 Villeurbanne cedex - France
{jean-michel.jolion, isabelle.simand}@liris.cnrs.fr

Abstract. This paper shows how strings can be used in a natural images classification task. We propose to build an attributed string from a set of regions of interest detected thanks to an interest point detector. These salient zones are characterized by local signatures describing singularities and they are linked by using graph seriation algorithms and perceptual methods. Once each image is represented by a string of signatures, we propose to use string-based edit distances and an ordered histograms-based distance in order to perform the classification task. Experiments have shown that whereas seriation algorithms give approximately the same results, the ordered histogram based distance is more efficient for the considered application.

1 Introduction

Nowadays, digital images are more and more present in the cyberworld. Indeed, peer to peer sharing, digital camera and Internet network provide an access to a lot of images for most people. As image databases grow exponentially, people need to have powerful solutions to manage them. Image classification is one such solution allowing to group images into semantically meaningful categories. It can thus be helpful for daily tasks such as browsing, annoting, indexing, etc. In this paper, we are interested in classification methods using low level image features. Such image clustering approaches perform first a feature extraction step in order to reduce the amount of data and to extract relevant and discriminating measures used during the classification step. This extraction phase results in a feature vector (also called *signature*) describing the image content.

Classically, image recognition approaches extract the image signature by considering the image content as a whole. Signatures can describe color by using e.g. classical histograms [16] or even texture [11] by using e.g. Gabor filter banks. However, during the last decade, it has been shown that better classification

rates can be obtained by computing signatures around only a limited number of pixels called *interest points*. Recognition is then performed thanks to registering algorithms. In this case, the problem is clearly the lack of ordering between the interest points because the image can no longer be considered as a vector, increasing thus the complexity of the classification step. In this paper, we propose to define such an order thanks to two main approaches : a spectral graph seriation approach and a saliency-based approach. The image is then described by a string of local signatures, each one characterizing singularities in the region of interest thanks to a foveal wavelet descriptor [13]. Finally, the last classification step can be performed by a distance between strings. We have tested some of them and present the classification results.

The paper is organized as follows. Section 2 presents the salient points detector used in order to define the string nodes. Section 3 describes the different methods that can be envisaged to generate a string from a set of salient points. These strings are then compared thanks to some distances presented in section 4. Experiments comparing these approaches are presented in section 5 and finally, section 6 concludes this paper.

2 Interest Points Detection

The use of interest points for image retrieval was proposed in [3, 14] and was motivated by the definition of special points which capture only the relevant information of the signal. Consequently, assigning a local signature to each region of interest centered on an interest point could be more discriminative of the image content than computing a global signature. Nevertheless, finding interest points is quite difficult because it requires the definition of what is perceptually relevant in a signal.

Many approaches have been proposed in the literature to detect interest points. In [5], an algorithm using the local auto-correlation of the image localizes them on corners. Although this detector is very often used [14], it has the drawback of positioning the points on textured regions omitting other regions which can be critical for the classification. Moreover, there is no perceptual justification about the importance of corners. In [2], the authors propose a detector that locates interest points in high contrast area. Finally, observing that multi-resolution, orientation and frequency analysis are of prime importance for the Human Visual System, some wavelet-based detectors have been proposed in [7, 9] that locate points on sharp region boundaries.

The detector proposed in [7] is used in our system and proceeds as follows:

- a discrete wavelet transform [10] is firstly performed on the image I up to a resolution level 2^r ($r \leq -1$);
- the obtained wavelet coefficients are zerotree represented [15] resulting in a hierarchical data structure (tree) of wavelet coefficients;
- this tree is traversed a first time from leaves to the root node by computing at each resolution 2^j ($j \leq -1$) a saliency map $S_{2^j}^I$ reflecting the perceptual

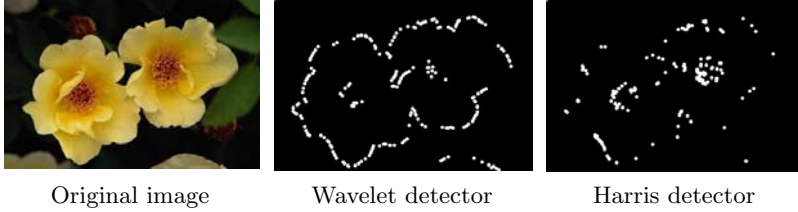


Fig. 1. Interest points detection

relevance of the wavelet coefficients present in the level 2^j . The saliency value $S_{2^j}^I(x, y)$ at the location (x, y) is defined by:

$$\begin{cases} S_{2^{-1}}^I(x, y) = \alpha_{-1} \left(\frac{1}{3} \sum_{s=1}^3 \frac{|w_{2^{-1}}^s(x, y)|}{|Max(D_{2^{-1}}^s)|} \right) \\ S_{2^j}^I(x, y) = \frac{1}{2} \left(\alpha_j \left(\frac{1}{3} \sum_{s=1}^3 \frac{|w_{2^j}^s(x, y)|}{|Max(D_{2^j}^s)|} \right) \right. \\ \left. + \frac{1}{4} \sum_{u=0}^1 \sum_{v=0}^1 S_{2^{j+1}}^I(2x + u, 2y + v) \right) \end{cases} \quad (1)$$

where $w_{2^j}^s(x, y)$ stands for the wavelet coefficient of the subband $D_{2^j}^s$ located at (x, y) , $Max(D_{2^j}^s)$ ($s = 1, 2, 3$) denotes the maximum wavelet coefficient value over the detail subband $D_{2^j}^s$, and α_k (with $k \in [r, -1]$ and $0 \leq \alpha_k \leq 1$) is a weighting factor balancing the importance of saliency values with respect to the resolution level;

- from the saliency maps previously computed, the tree is traversed a second time from the root to the leaves in order to choose, at each tree level, the most salient wavelet coefficients.

The final result of these different steps is the construction of a saliency map S^I with the same resolution as I and that reflect the perceptual importance of the pixels. Indeed, the higher is $S^I(x, y)$, the more the pixel (x, y) is perceptually important. If N interest points are needed, then the N pixels with highest coefficients $S^I(x, y)$ in the saliency map are chosen.

As it can be seen on Figure 1, this interest point detector locates points on sharp region boundaries. The points are also more spread than the classical Harris corner detector [5] in the case of textured images.

3 Strings Construction

Interest points are usually mixed with registration techniques [14] for assessing similarity between images. In these approaches, each point is considered independently of each other and the dependencies or correlation that may exist between them are not used. However, it is well known that human eyes are able to classify an image from a set of focus of attention and saccadic eye movements. We

propose thus to link the detected interest points in order to construct a string composed of local signatures which describe each region of interest centered on an interest point. Therefore, images comparison can be performed by a string comparison. Several techniques to construct such strings can be considered and we propose to compare some of them.

3.1 Graph Seriation

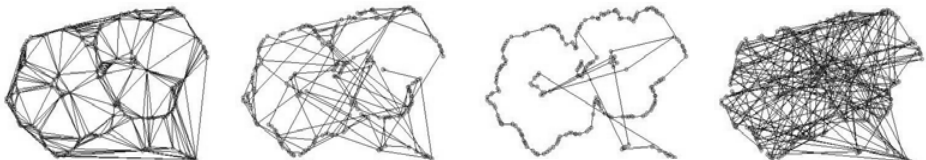
From a set of interest points, an attributed graph can be generated thanks to a Delaunay triangulation (see Figure 2(a)) in which each node of the graph is described by a local signature describing the image content in the neighborhood of the node. This graph can then be transformed into a string by a graph seriation approach [1, 4]. Two kinds of graph seriation are studied in this paper.

Spectral Graph Seriation. In the spectral graph seriation approach, the string is constructed by only considering the adjacency matrix of the graph and implicitly its structure. In [4], the authors propose an algorithm which performs the graph seriation by using the eigenvector ϕ^* corresponding to the leading eigenvalue of the adjacency matrix. Nodes are then ordered in the decreasing order of their magnitude in the leading eigenvector components (see Figure 2(b)).

Similarity Spectral Graph Seriation. In the case of attributed graphs, it seems relevant to consider nodal values and thus the similarity between them. The idea was first proposed in [1] where the seriation is performed thanks to the Fiedler vector of the Laplacian matrix. In the following, we propose an alternative of it.

As in [4], we begin from the node associated with the largest component of ϕ^* . Next, we search through the set of the nearest neighbors, the node which is the most similar to the previous one in the sense of a L_2 distance between the foveal wavelets signatures [13] associated to the nodes being compared. This step is repeated until all nodes have been visited.

This method permits to generate strings following edges (see Figure 2(c)). Indeed, a foveal signature characterizes orientation and regularity of an edge, thus two signatures are similar if they belong to the same edge. Note that this method is similar to [1] where the Laplacian matrix is replaced by a similarity



(a) Delaunay graph (b) Spectral seriation [4] (c) Similarity seriation (d) Perceptual seriation

Fig. 2. String construction from a spatial distribution of interest points. The original image is shown on Figure 1

matrix $A = [a_{ij}]$ where $a_{ij} = d(s_i, s_j)$ with s_i being the local signature associated to the node x_i and d the distance between local signatures.

3.2 Perceptual Seriation

The second method presented in this paper consists in building a string by considering only the saliency values of the detected interest points (see equation 1). In this case, interest points are ordered in the decreasing order of the saliency value magnitude (see Figure 2(d)).

4 Strings Comparison

Once strings are constructed by one of the methods exposed in section 3, the last step consists in matching them to assess similarity between images.

4.1 Edit Distance and its Variants

Strings comparison can be first performed by using distances proposed in the field of automatic spelling correction or texts comparison. Such distances are based on the work of Levenstein presented in [8]. If we denote by Σ a finite alphabet and by $X = (x_1x_2\dots x_n)$ and $Y = (y_1y_2\dots y_m)$ two finite strings whose elements are in Σ then the string edit distance $D(X, Y)$ is the minimum cost needed to transform X into Y using elementary edit operations. These edit operations are of three kinds:

- $(x_i \rightarrow y_j)$ is the substitution of the symbol x_i by y_j ;
- $(x_i \rightarrow \epsilon)$ denotes the suppression of the symbol x_i ;
- $(\epsilon \rightarrow y_j)$ denotes the insertion of the symbol y_j .

If a cost function γ is assigned to each of these edit operations, then the string edit distance can be efficiently computed in $O(mn)$ thanks to a dynamic programming algorithm [18] based on the following recursive property:

$$D(i, j) = \min \begin{cases} D(i-1, j-1) + \gamma(x_i, y_j) \\ D(i-1, j) + \gamma(x_i, \epsilon) \\ D(i, j-1) + \gamma(\epsilon, y_j) \end{cases} \quad (2)$$

where $D(i, j)$ is the edit distance between the sub-strings $(x_1\dots x_i)$ and $(y_1\dots y_j)$. Nevertheless, in [12], the authors show that the classical string edit distance lacks some normalization because it does not consider the length of the strings to be compared. For example, if X and Y are two strings of length 2, they can have the same edit distance as two strings of length 50. However, it seems that in the second case, they are more similar. Consequently, the normalization of the edit distance by the length of the edit path was proposed. It leads to the following definition:

$$d(X, Y) = \min_P \left(\frac{W(P)}{L(P)} \right) \quad (3)$$

where:

- P is an edit path from X to Y , $W(P)$ is the cost of this edit path;
- $L(P)$ is the length of the edit path.

The computation of the normalized edit distance is performed thanks to a fractional programming algorithm [17] with the same complexity than the edit distance algorithm.

Finally, a major improvement of the classical string edit distance is proposed in [19] where the authors propose to use the neighborhood of each symbol in the string to compute the distance. This work is based on the Markov field theory because it is shown that the classical string edit distance algorithm can be seen as a zero order Markov edit distance.

In the following, we use edit operations costs defined in [13]. The substitution cost $\gamma(x_i, y_j)$ is the L_2 distance between the two foveal signatures x_i and y_j and the insertion and deletion costs $\gamma(\epsilon, y_j)$ and $\gamma(x_i, \epsilon)$ are defined by the L_2 distance between the foveal signature considered (i.e. y_j or x_i) and the null signature (i.e. the signature filled with 0) corresponding to the signature of an homogeneous region.

4.2 Ordered Histograms-Based Distance

Histograms are known to be very powerful in the case of content-based image retrieval [16]. They permit to capture the essential statistics present in the images and the comparison of them is less expensive than string-based edit distance algorithms. Consequently, coupling these two approaches for comparing strings of local signatures can be of interest. In [6], the authors propose a distance which considers the order and the distribution of symbols present in a string. Nevertheless, this distance must be adapted in the case of strings of signatures whose values are continuous in a k -dimensional space. For this purpose, we propose to compute k distances defined in [6], each one for a component in a signature. Then, we sum them to get the final distance between the two strings. Furthermore, as underlined in [7], each image can be represented by a different number of salient points depending on the complexity of the image content. Consequently, the two strings to be compared can have different lengths breaking thus the triangular inequality of the distance proposed in [6]. This modification leads to obtain a dissimilarity measure.

The algorithm proceeds as follows for two attributed strings X and Y :

```

1:  $d \leftarrow 0$ 
2: for  $j : 1$  to  $k$  do
3:    $H_1^j \leftarrow 0; H_2^j \leftarrow 0; W \leftarrow 0$ 
4:   for  $i : 1$  to  $\min(m, n)$  do
5:      $H_1^j(x_i[j]) \leftarrow H_1^j(x_i[j]) + (\min(m, n) - i + 1) * c(x_i[j], y_i[j])$ 
6:      $H_2^j(y_i[j]) \leftarrow H_2^j(y_i[j]) + (\min(m, n) - i + 1) * c(x_i[j], y_i[j])$ 
7:      $W \leftarrow W + (\min(m, n) - i + 1) * c(x_i[j], y_i[j])$ 
8:   end for

```

```

9:    $d \leftarrow \sum_i |H_1^j(i) - H_2^j(i)|/W + d$ 
10: end for
11: return  $d$ 

```

where $c(x_i[j], y_i[j])$ is the substitution cost between j^{th} elements of the two signatures x_i and y_i and H_1 and H_2 are respectively the histograms associated to X and Y . This algorithm is computationally less expensive than a string edit distance because it computes the distance in $O(\min(m, n)k)$. However, it makes the strong assumption that the two strings to be compared are perfectly aligned.

5 Experiments

In order to compare the different approaches, a supervised image classification system based on the k-nearest neighbors algorithm has been developed. We use a training images database which is divided into six clusters (Alps, football, cars, ships, flowers, space) (see Figure 3), each cluster being composed of 15 natural images. To test our system, we picked 90 different images that are proposed to the system for classification (see Figure 4). Regarding the parameters used during the experiments, the L_2 distance has been used to compute $c(x_i[j], y_i[j])$ in the ordered histograms-based distance algorithm (see section 4.2) and the histograms H_1^j and H_2^j are composed of 100 bins.

We have firstly compared the classification results obtained by each distance presented in section 4 for each seriation algorithm separately. Figure 5 presents the classification rates obtained by each method using different number of interest points and the computing times needed to compare two images using the different distances on a Pentium IV 3GHZ. It is clear that ordered histograms distance outperforms all other distances proving that the strong assumption discussed before is not so hard. Moreover, ordered histograms distance is computationally more efficient and it is essential in the case of large images databases. However, the main reason of these better classification rates is that our insertion and deletion costs for the edit distances are not well adapted and so they degrade

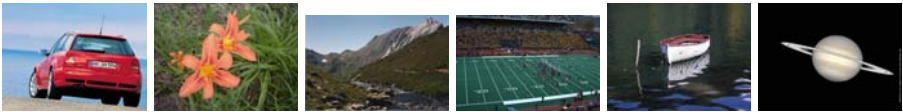


Fig. 3. Some image samples present in the training database



Fig. 4. Some image samples present in the test database

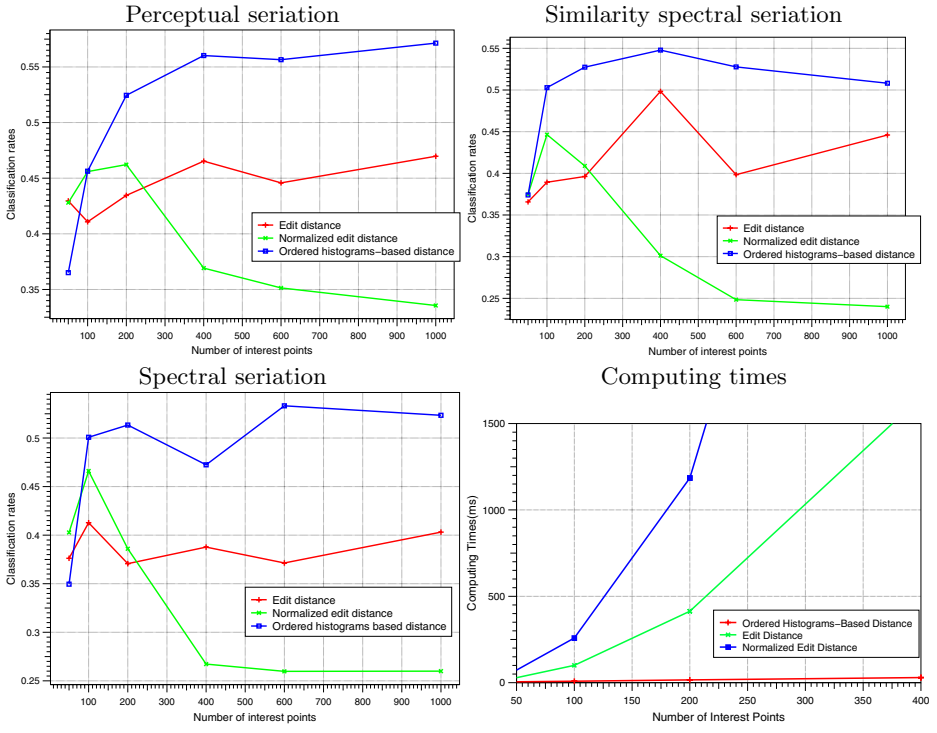


Fig. 5. Comparison of distances for the three seriation approaches

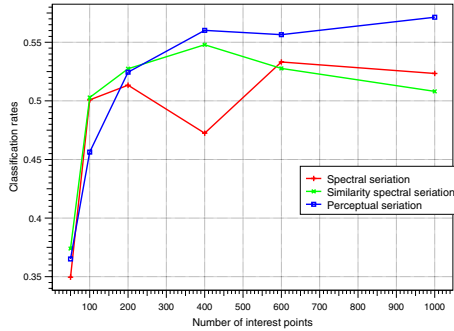


Fig. 6. Comparison of seriation methods

results. Thus, it could be interesting to take into account the probability density function of the foveal signatures in a cluster in order to define these edit costs.

If we compare the three seriation approaches using only the ordered histograms based distance (see Figure 6), we can see that the perceptual seriation gives the best result when more than 200 interest points are used. Nevertheless, classification rates are not so different. Finally, if we consider computing times,

it seems not relevant to use graph seriation approaches because they do not improve classification rates compared to a perceptual approach and they are more complex.

Finally, it is important to note that when color is not able to discriminate a cluster, our approach is well suited. In the dataset we used, it appears to be the case for the cars and ships clusters which obtain respectively 41% and 38% of good classification rates with a global histogram approach whereas our method permits to obtain 85% and 54% for 400 interest points.

6 Conclusion and Perspectives

In this paper, we have presented a comparison of seriation techniques and string distances that can be used in the case of a natural images classification task using foveal signatures. It has been shown that histograms-based distance gives better classification rates than the others edit distances presented. Nevertheless, it could be interesting to implement a training algorithm as in [6] in order to learn the edit costs which are very difficult to establish and on which depends critically the classification rates.

It has also been shown that seriation approaches give approximately the same classification rates indicating that string order is not very important in the recognition task. However, it could be interesting to implement the same approach in order to perform small substring matching and it is quite sure that we are also interested in other seriation approaches.

References

1. Atkins J.E., Boman E.G., and Hendrickson B. A Spectral Algorithm for Seriation and the Consecutive Ones Problem. *SIAM Journal on Computing*, 28(1):297–310, 1998.
2. Bres S. and Jolion J.M. Detection of Interest Points for Image Indexation. In *3^d Int. Conf. on Visual Information Systems*, pages 427–434, 1999.
3. Gouet V. and Boujeema N. Object-based queries using color points of interest. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 30–36, December 2001.
4. Hancock E.R. and Vento M. Graph Matching Using Spectral Seriation and String Edit Distance. In *4th IAPR Int. Workshop (GbRPR 2003)*, volume 2726 of *Lecture Notes in Computer Science*. Springer, 2003.
5. Harris C. and Stephens M. A Combined Corner and Edge Detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.
6. Jolion J.M. and Simand I. Representation d’Images par des Chaines de Symboles: Application à l’Indexation d’Images. In *COmpression et REprésentation des Signaux Audiovisuels (french workshop)*, 2004.
7. Laurent C., Laurent N., and Visani M. Color Image Retrieval Based on Wavelet Salient Features Detection. In *3^d Int. Workshop on Content-Based Multimedia Indexing*, pages 327–334, 2003.
8. Levenstein A. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Phys. Dokl.*, 10:707–710, 1966.

9. Loupiau E., Sebe N., Bres S., and Jolion J.M. Wavelet-based Salient Points for Image Retrieval. In *IEEE Int. Conf. on Image Processing*, pages 518–521, 2000.
10. Mallat S. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.
11. Manjunath B.S. and Ma W.Y. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.
12. Marzal A. and Vidal E. Computation of normalized edit distance and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):926–932, 1993.
13. Ros J. and Laurent C. Natural Image Classification Using Foveal Strings. In *The International Workshop on Multidisciplinary Image, Video, and Audio Retrieval and Mining*, October 2004.
14. Schmid C. and Mohr R. Local Grayvalue Invariants for Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.
15. Shapiro J.M. Embedded Image Coding Using Zerotrees of Wavelet Coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, 1993.
16. Swain M.J. and Ballard D.H. Color Indexing. *International Journal on Computer Vision*, 7(1):11–38, 1991.
17. Vidal E., Marzal A., and Aibar P. Fast Computation of Normalized Edit Distances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):899–902, September 1995.
18. Wagner R.A. and Fischer M.J. The String-to-String Correction Problem. *Journal of the Association for Computing Machinery*, 21(1):168–173, 1974.
19. Wei J. Markov Edit Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):311–321, March 2004.