Classification of images based on Hidden Markov Models

Marc Mouret[a,b] Christine Solnon[a,b] Christian Wolf[a,c]
[a]Université de Lyon, CNRS
[b]Université Lyon 1, LIRIS, UMR5205, F-69622, France
[c]INSA-Lyon, LIRIS, UMR5205, F-69621, France
marcmouret@gmail.com, {christine.solnon,christian.wolf}@liris.cnrs.fr*

Abstract

We propose to use hidden Markov models (HMMs) to classify images. Images are modeled by extracting symbols corresponding to 3×3 binary neighborhoods of interest points, and by ordering these symbols by decreasing saliency order, thus obtaining strings of symbols. HMMs are learned from sets of strings modeling classes of images. The method has been tested on the SIMPLIcity database and shows an improvement over competing approaches based on interest points. We also evaluate these approaches for classifying thumbnail images, i.e.*, low resolution images.*

1. Introduction

Image classification, indexing and retrieval traditionally resort to features of different types, namely color [8], texture [24] and shape [1, 16] extracted globally or locally on interest points [13], regions or contour segments [6]. While object detection algorithms have become quite sophisticated integrating accurate shape models [9, 23, 7], the large intra-class variation in image classification problems prevents precise and detailed shape modeling. Not surprisingly, the most successful methods are based on bags of visual words [12, 18], *i.e.* representations which build a vocabulary of visual words and which do *not* take into account the spatial distribution of the features in the image. Introducing spatial dependencies has been attempted several times, however, the task is difficult: the dependencies need to be strong enough to improve classification, but not too strong in order to allow for the large intra-class variance of the dataset.

Hidden Markov models (HMMs) have already been used for object detection, for instance in [21]. However, mostly the very accurate spatial modeling of the object's appearance does not allow to generalize across large shape variations, which is necessary for image classification. Their application in image classification is mostly restricted to regularizing segmentation algorithms, where the classification itself is handled by a different method [11], or in restricted domains as in document image classification, where blocks can be represented as nodes of a hidden Markov tree [4].

The work closest to our method might be [14], which classifies images using a part based conditional random field whose tree shape is calculated with a minimum spanning tree on the SIFT keypoint positions, and similar work presented in [5].

Our solution is based on the representation of images as strings of symbols: symbols correspond to 3×3 binary neighborhoods of interest points; they are ordered by decreasing saliency order. By integrating saliency order but ignoring spatial distribution of interest points, we enrich the discriminative power while keeping its variance. A hidden Markov model captures differences between classes and intends to model the large variations between image classes.

The paper is outlined as follows. Section 2 motivates the ordering of visual keywords into strings. Section 3 introduces the Markov model. Section 4 experimentally evaluates our approach on classical, *i.e.* rather high resolution images, and compares it with other approaches based on interest points. Section 5 gives experimental results on thumbnail images, *i.e.* rather low resolution images. Section 6 finally concludes.

2. From bags of visual words to strings of salient points

Bags of visual words have been first introduced in [12, 18] and different variants have been proposed in other works. The basic idea can be described as follows: the image is considered as a kind of visual document and treated as such, *i.e.* as a collection of "words", which allows to apply many successful algorithms of the text retrieval community. The

*The authors acknowledge an ANR grant BLANC 07-1_184534: this work was done in the context of project SATTIC.

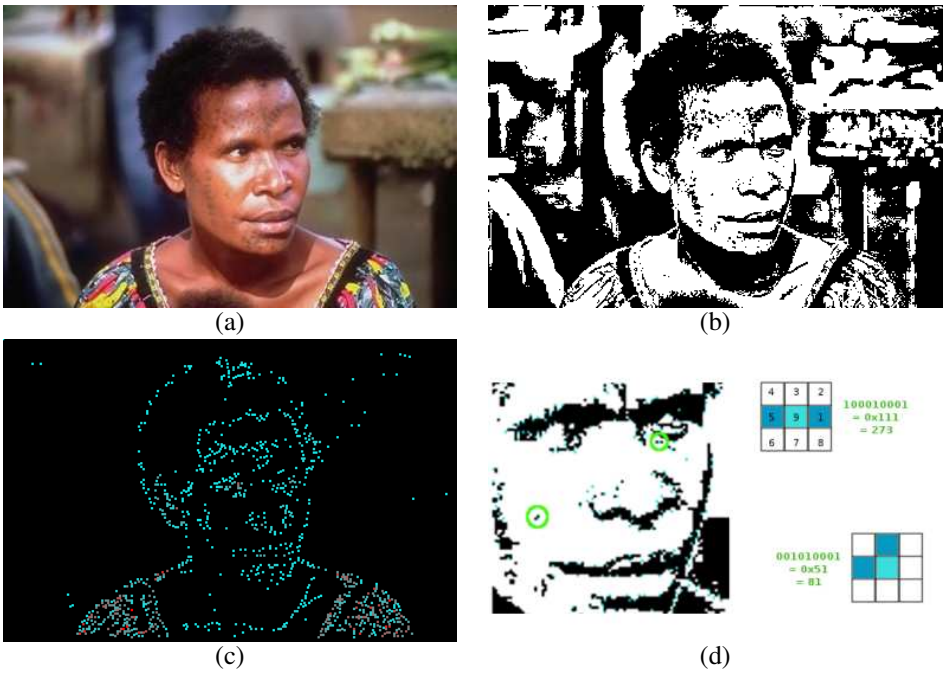

Figure 1. Symbol extraction: (a) input image (b) binarized image (c) extracted points (d) extracted symbols.

vocabulary of visual words can be obtained by different means, for instance by clustering local features extracted from a database of training images. On the test image, the same features are extracted and for each feature the nearest codeword is found, giving rise to a set of words.

In spite of its simplicity, the approach is one of the most successful ones in the image classification community. However, a major drawback remains, namely the lack of interaction between the local features.

Ros et al. have proposed to model images by strings of discrete symbols [17]. Starting from a binarized image (see Fig. 1b), they extract interest points with the detector proposed by Bres and Jolion [2]. The location of the key points corresponds to high contrast at several levels of resolution, extracted from a contrast pyramid. This contrast measure is called saliency in the rest of this paper[1]. Fig. 1c shows an example of detected locations.

Each of these points is associated with a symbol which corresponds to its local 3×3 binary neighborhood so that a number between 0 and $2^9-1 = 511$ is assigned to each symbol (see Fig. 1d). Finally, these symbols are ordered into a string by decreasing contrast energy.

The main advantage of this method is also its main drawback: the ordering of the symbols by saliency introduces interactions between the features, which are not necessarily representative at a local level. For this reason, classical string edit distances as the one of Levenstein [10] are not relevant and give very disappointing classification results.

In [19] we have proposed and compared histogram-based distances for such strings of symbols. The first distance, called d_H, does not consider the order of symbols in strings: it is defined as the sum, for every symbol $s \in [0; 511]$, of the absolute difference of the number of occurrences of s in the two strings. The second distance, called d_{H_ω} takes into account the order the symbols appear in strings by associating a weight ω_k with every position k in strings. This weight is defined by $\omega_k = l - k$, where l is the length of the string, so that symbols at the beginning of the string have higher weights than those at the end of the string. This weight is integrated in d_{H_ω} by considering a weighted sum instead of the number of occurrence of each symbol s. Experiments showed us that d_{H_ω} significantly improves classification rates with respect to d_H, thus bringing to the fore the interest of integrating saliency order. Indeed, saliency order avoids fixing the class models too much in terms of geometry and shape, allowing the discriminance-invariance trade-off to slightly favor invariance —needed in image classification problems as opposed to object detection problems.

3. Modeling images with HMMs

In order to exploit saliency ordering when classifying images, we propose to statistically model the symbol changes indirectly by considering them as observations associated to a chain of unknown (hidden) states, which can be described as a HMM [15]. In other words, the ordering of the symbols

[1] Saliency is a term not rigorously defined. It is commonly associated to high and hopefully "relevant" local changes of the image content.

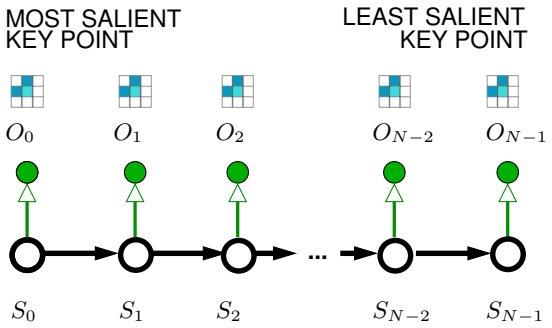

Figure 2. The dependency graph of the HMM. Empty nodes are hidden, shaded nodes are observed.

(the observations) is only relevant (and modeled) indirectly through the ordering of unknown hidden states, whose semantic meaning does not need to be known. The hidden states are learned during the training phase.

Figure 2 shows the dependency graph of the model for an image with N interest points: each node corresponds to a variable, where observed variables are denoted O_i and shown as shaded circles whereas hidden (state) variables are denoted S_i and depicted as empty circles. In the following and as usual, uppercase letters denote random variables or sets of random variables and lower case letters denote realizations of values of random variables or of sets of random values. In particular, $p(O_i{=}o_i)$ will be abbreviated as $p(o_i)$ when it is convenient. The whole set of observations will be denoted as O and the whole set of hidden state variables will be denoted as S.

Each hidden state variable S_i is assigned to an observation O_i and the i^{th} pair of hidden and observed variables (S_i, O_i) corresponds to the i^{th} interest point in the image with respect to decreasing saliency order. Therefore the hidden state variables S_i and the observations O_i are indexed by the same index i defined through the ordering of the interest points by saliency. Each observation O_i may take values in $[0, 511]$ —the symbol values described in section 2— and each hidden variable S_i may take state values in $[0, T{-}1]$ where T —the size of the state space— is a user-defined parameter.

The joint probability distribution $p(s, o)$ of the whole set of variables $\{s, o\}$ is parameterized by three parameter sets: the transition matrix A containing the transition probabilities $a_{kl} = p(S_i = k|S_{i-1} = l)$, the observation emission matrix B containing the symbol emission probabilities $b_{kt} = p(O_i = t|S_i = k)$ and the initial state probabilities $\pi_k = p(S_0 = k)$. The probability factorizes as follows:

$$p(s, o) = \pi_{s_0} \prod_{i=1}^{N-1} a_{s_i s_{i-1}} \prod_{i=0}^{N-1} b_{s_i o_i} \qquad (1)$$

In our approach, each image class is characterized by its own behavior, therefore each class is described by its own model. The difference between two classes is thus of statistical nature: not the interaction between the observations themselves is described, but the interaction between a set of yet to learn hidden states and the production of the observations from these hidden states. Note that, as the hidden states are a number of discrete values only, where the state space size T is a user-defined parameter, their meaning is defined through their interactions, *i.e.* the values of the transition matrix $A = a_{kl}$. The HMM learning algorithm tries to find the optimal description in terms of hidden states and state transition probabilities producing the observed symbols. The meaning of the hidden states themselves can stay unknown to the algorithm, as in similar work [5, 14].

The parameters for each class c —two matrices and a vector — can be denoted as a parameter vector $\theta^c = [\pi^c_k \; a^c_{kl} \; b^c_{kt}]^T$. The problem is thus twofold:

- in the training phase, learn the parameter vector θ^c for each class c;
- in the test phase, determine for each image modeled by a string of symbols o, the most probable class c, *i.e.*,

$$\hat{c} = \arg\max_c \; p(o|\theta^c)$$

We use the forward algorithm in order to determine the probability $p(o|\theta^c)$ as well as the classical Baum-Welch variant of the Expectation-Maximization algorithm in order to learn the parameter vectors [15]. From the multiple variants learning the parameters from multiple observation sequences [3] we chose averaging over the parameters learned from single observation sequences.

Our general training method is supervised, since the class label for each image is considered as known and used during the training phase. However, for each image we resort to the training algorithm Baum-Welsh which is considered as unsupervised, since the values of the different hidden states of the training image are not known. Indeed, as mentioned above, during the training phase we try to find the best state behavior for each class, *i.e.* the parameter vector θ^c which best explains the observed symbols o_i extracted from the training images. This makes it possible to calculate the optimal state sequence $s_i, i = 1..N$ for each training image using the Viterbi algorithm [20, 15]. It could be interesting to study these sequences in order to find out what kind of "hidden behavior" has been found by the training algorithms. However, the state sequences are not needed for classification.

4. Experimental results

We have performed experiments on the SIMPLIcity database [22] which contains 1000 images extracted from

	500	*1000*	*2000*	*4000*
HMM(1)	63.1	63.2	62.7	62.8
HMM(2)	63.5	64.4	**68.1**	67.3
HMM(5)	63.4	64.9	67.1	70.0
HMM(10)	**64.5**	65.2	67.1	70.2
HMM(20)	62.6	65.0	66.6	70.1
HMM(50)	58.4	63.9	67.2	**70.6**
HMM(100)	51.8	60.4	66.1	70.3
KNN(d_H)	63.2	64.3	64.0	62.8
KNN(d_{H_ω})	63.0	66.3	67.6	66.2
GM(d_H)	63.4	65.9	58.0	50.8
GM(d_{H_ω})	61.6	**66.4**	65.9	60.8

Table 1. Average classification rates when considering the 500, 1000, 2000 and 4000 most salient points. HMM(T) denotes HMMs with T states. KNN(d) denotes the k nearest neighbor approach with respect to distance d. GM(d) denotes the generalized median strings with respect to distance d.

the well known commercial COREL database[2]. The database contains ten clusters representing semantic generalized meaningful categories, *i.e.*, "African people, "beaches", "buildings", "buses", "dinosaurs", "elephants", "flowers", "food", "horses" and "mountains". Needless to say that the categories are extremly heterogenous in terms of signal contents, as illustrated in Fig. 3. There are 100 images per cluster.

Each image contains 384×256 pixels and is modeled by a string of symbols extracted as explained in Section 2. The original strings have 4000 symbols[3]. To study the influence of the length of the strings, we report experimental results obtained when limiting the number of symbols to different lengths ranging between 500 and 4000.

Table 1 compares classification rates obtained with three different approaches.

- HMM(T) is the approach based on hidden Markov models described in section 3 where T is the parameter which determines the number of states.
- KNN(d) is the k nearest neighbors approach: to classify an image, we compute the distance between this image and every image of a database, the classes of the k closest images determine the class. We set k to 5, which gives the best average results.
- GM(d) gives results obtained when characterizing each image class by the generalized median, *i.e.*, the string which minimizes its distance to every string in the class, and classifying images with respect to the class of its closest generalized median.

Both for KNN and GM, we report results obtained with the two histogram-based distances d_H and d_{H_ω} introduced in [19] and recalled in Section 2. The training set and the test set were kept seperated by following a strict "leave-one-out" principle: the string which is classified is removed from its class before computing the HMM parameters or the generalized median of this class. So each string is tested only on parameters to whose learning it did not contribute.

Let us first compare results obtained with different values of the T parameter which determines the size of the state space of HMMs. Indeed, when $T = 1$, HMMs only learn the probability $b_{1t} = p(O_i = t|S_i = 1)$ of observing symbol t (with $t \in [0; 511]$) as all hidden states are always equal to 0. In this case, the symbol order is not taken into account at all. Increasing T allows to integrate information brought by the ordering by saliency. Table 1 shows that for shorter strings, HMMs often obtain better results with smaller numbers of hidden states whereas for strings of 4000 symbols, the best results are obtained with larger numbers of hidden states.

Note that linearly increasing the size of the state space T will quadratically increase the number of parameters in the matrix A, which causes the well-known associated problems: very quickly the requirements in terms of training data will be too high in order to reliably learn the parameters.

The result also shows us that the 500 first symbols are more or less all equally important so that the information related to the ordering should not be too much emphasized. However, classification rates are significantly improved when integrating information provided by other symbols, after the 500 first ones, as classification rates are improved when considering longer strings, provided that the order of symbols is actually taken into account.

The interest of integrating saliency ordering is also brought to the fore when comparing results obtained with d_H and d_{H_ω}, both for KNN and GM. Indeed, for strings of 500 symbols, d_H obtains better results than d_{H_ω} whereas, for strings of more than 500 symbols, d_{H_ω} is significantly better than d_H.

The fact that HMM obtains better results than KNN and GM (except for strings of length 1000) shows that saliency order is better exploited with HMMs. Hence, HMMs are able to classify correctly more than 70% of the 1000 images of the simplicity database. Table 2 gives the corresponding confusion matrix. The best results are obtained on dinosaurs (100%), buses (92%) and flowers (88%), which contain rather specific images, whereas the worst results are

[2]The SIMPLIcity database can be downloaded on the James Z. Wang web site at http://wang.ist.psu.edu/jwang/test1.tar.

[3]Images with 384×256 often have less than 4000 salient points so that it is not reasonable to extract more than 4000 points. When there are less than 4000 salient points, strings are padded with a new extra symbol.

Figure 3. Some examples from the SIMPLIcity image database used in the experiments.

	A	B	C	D	E	F	G	H	I	J
A	**66**	1	1	1	2	5	19	3	1	1
B	0	**92**	1	0	1	0	2	1	3	0
C	2	0	**78**	0	15	0	1	0	4	0
D	0	0	0	**100**	0	0	0	0	0	0
E	2	1	11	1	**64**	0	11	2	8	0
F	9	0	0	0	1	**88**	1	0	0	1
G	3	3	7	0	11	1	**63**	3	2	7
H	3	13	2	0	7	0	7	**64**	2	2
I	5	9	6	2	14	0	14	13	**34**	3
J	11	0	0	1	9	4	14	1	3	**57**

Table 2. Confusion matrix: line X column Y gives the number of images of class X that have been classified in class Y by HMM(50) when using 4000 points, where A=African people, B=buses, C=horses, D=dinosaurs, E=elephants, F=flowers, G=mountains, H=buildings, I=beaches, J=food.

	500	*1000*
HMM(1)	**61.6**	60.1
HMM(2)	60.7	60.6
HMM(5)	61.0	**60.9**
HMM(10)	59.7	60.3
HMM(20)	57.7	59.1
HMM(50)	54.8	56.7
HMM(100)	47.9	53.5
KNN(d_H)	58.3	55.7
KNN(d_{H_ω})	57.9	59.3
GM(d_H)	58.4	52.0
GM(d_{H_ω})	60.4	57.2

Table 3. Average classification rates for thumbnail images respectively modeled by 500 and 1000 symbols.

obtained on beaches (34%) and food (57%), which contain very different images.

Let us point out that this result is obtained *without using any color information* as images are binarized before extracting interest points, and we only consider the binary black and white 3×3 mask around each pixel.

Finally, let us emphasize that HMM, KNN and GM have different time complexities. Indeed, the complexity of classifying a new image with KNN depends on the number of images in the database as one has to compute its distance to all images in the database. This is not the case for HMM and GM for which one only has to compute a distance or measure a probability with respect to only one representative by class (a generalized median or a learned HMM).

5 Classification of thumbnail images

In this section, we evaluate our approach and compare it with those of [19] on thumbnail images, *i.e.*, low resolution images such as those returned by search engines on the Web. Indeed, approaches designed for high resolution images, such as methods based on segmentation and/or local features with large support, can hardly be used to process those thumbnail images. Our goal here is to evaluate relevancy of approaches based on interest points, including our new HMM-based approach, for classifying these low resolution images.

To this aim, we have reduced all images of the SIMPLIcity database from their original size of 384×256 pixels to 128×86 pixels. For these thumbnail images, we have only extracted the 1000 first salient points.

Table 3 shows us that, for these thumbnail images, it is not really relevant to integrate saliency ordering. Indeed,

better results are obtained with smaller number of states for HMM, and with d_H for KNN and GM. However, we are still able to correctly classify 61.6% of the thumbnail images with HMM(1).

6 Conclusion

We have proposed to use hidden Markov models to classify images modeled by strings of interest points ordered with respect to saliency. Experimental results have shown that this approach allows one to improve classification rates by better integrating saliency ordering. Experimental results have also shown that approaches based on interest points may be used to classify thumbnail images, that have rather low resolution. However, in this case, saliency ordering is not really relevant.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–521, 2002.

[2] S. Bres and J.-M. Jolion. Detection of interest points for image indexation. In L. N. i. C. S. Springer Verlag, editor, *Proceedings of the third international conference on Visual Information and Information Systems*, volume 1614, pages 427–434, 1999.

[3] R. Davis, B. Lovell, and T. Caelli. Improved estimation of hidden markov model parameters from multiple observation sequences. In *Proceedings of the International Conference on Pattern Recognition*, volume 2, pages 168–171, 2002.

[4] M. Diligenti, P. Frasconi, and M. Gori. Hidden tree markov models for document image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):519–523, 2003.

[5] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In IEEE, editor, *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 264–271, 2003.

[6] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):36–51, 2008.

[7] D. Hoiem, C. Rother, and J. Winn. 3d layoutcrf for multi-view object class recognition and segmentation. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[8] J. Huang, S. Kumar, and R. Zabih. Automatic hierarchical color image classification. *EURASIP Journal on Applied Signal Processing*, 1:151 – 159, 2003.

[9] A. Kapoor and J. Winn. Located hidden random fields: Learning discriminative parts for object detection. In *European Conference on Computer Vision (ECCV)*, 2006.

[10] A. Levenstein. Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phy. Dohl.*, 10:707–710, 1966.

[11] J. Li, A. Najmi, and R. Gray. Image classification by a two-dimensional hidden markov model. *IEEE Transactions on Signal Processing*, 48(2):517–533, 2000.

[12] J.-H. Lim and J. Jin. Home photo indexing using learned visual keywords. In *ACM International Conference Proceeding Series - Selected papers from the 2002 Pan-Sydney workshop on Visualisation*, pages 69–74, 2003.

[13] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, 1999.

[14] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1097–1104. MIT Press, 2005.

[15] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[16] J. Revaud, G. Lavoué, and A. Baskurt. Improving zernike moments comparison for optimal similarity and rotation angle retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):627–636, 2009.

[17] J. Ros, C. Laurent, J.-M. Jolion, and I. Simand. Comparing string representations and distances in a natural images classification task. In *GbRPR*, volume 3434 of *Lecture Notes in Computer Science*, pages 72–81. Springer, 2005.

[18] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the International conference on computer vision (ICCV)*, volume 2, pages 1470–1477, 2003.

[19] C. Solnon and J.-M. Jolion. Generalized vs set median strings for histogram-based distances: algorithms and classification results in the image domain. In *5th IAPR-TC-15 workshop on Graph-based Representations in Pattern Recognition*, number 4538 in LNCS, pages 404–414 (poster). Springer, 2007.

[20] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, IT-13:260–269, 1967.

[21] J. Wang, V. Athitsos, S. Sclaroff, and M. Bethke. Detecting objects of variable shape structure with hidden state shape models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):477–492, 2008.

[22] J. Wang, J. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(9):947–963, 2001.

[23] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. volume 1, pages 37–44, 2006.

[24] C. Wolf, J. Jolion, W. Kropatsch, and H. Bischof. Content Based Image Retrieval using Interest Points and Texture Features. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 234–237, 2000.