Fast-Learning Adaptive-Subspace Self-Organizing Map: An Application to Saliency-Based Invariant Image Feature Construction

Huicheng Zheng, Grégoire Lefebvre, Christophe Laurent

Fast-Learning Adaptive-Subspace Self-Organizing Map: An Application to Saliency-Based Invariant Image Feature Construction

Huicheng Zheng, *Member, IEEE*, Grégoire Lefebvre and Christophe Laurent

***Abstract*— The Adaptive-Subspace Self-Organizing Map (ASSOM) is useful for invariant feature generation and visualization. However, the learning procedure of the ASSOM is slow. In this paper, two fast implementations of the ASSOM are proposed to boost ASSOM learning based on insightful discussions of the basis rotation operator of ASSOM. We reveal the objective function approximately maximized by the classical rotation operator. We then explore a sequence of two schemes to apply the proposed ASSOM implementations to saliency-based invariant feature construction for image classification. In the first scheme, a cumulative activity map computed from a single ASSOM is used as descriptor of the input image. In the second scheme, we use one ASSOM for each image category and a joint cumulative activity map is calculated as the descriptor. Both schemes are evaluated on a subset of the Corel photo base with 10 classes. The multi-ASSOM scheme is favored. It is also applied to adult image filtering and shows promising results.**

***Index Terms*— Adaptive-Subspace Self-Organizing Map, feature construction, adult image filtering, image classification.**

I. Introduction

THE Adaptive-Subspace Self-Organizing Map (ASSOM) proposed by Kohonen [1], [2] is basically a combination of the SOM [3] and the subspace method. By setting filters to correspond to basis vectors that span pattern subspaces, some transformation groups can be taken into account automatically. The ASSOM is an alternative to the standard Principal Component Analysis (PCA) method of feature generation. An earlier neural approach for PCA can be found in [4]. Equivalence between probabilistic PCA and a typical subspace method for Gaussian density estimation has been established [5]. The ASSOM can generate spatially ordered feature filters thanks to interactions among processing modules [2]. The input to an ASSOM array is typically an episode, i.e. a sequence of pattern vectors supposed to approximately span certain subspace. Typical examples of episodes include sequences of temporally consecutive speech signals or of image patches subject to transformations. By learning the episode as a whole, the ASSOM is able to capture the transformation coded therein. The simulation results in [1] and [2] have demonstrated that the ASSOM can induce ordered filter banks to account for translation, rotation and scaling. The relationship between the neurons in the ASSOM and their biological counterparts are reported in [2]. The ASSOM has been successfully applied to speech processing [6], texture segmentation [7], image retrieval [8] and image classification [8], [9]. A supervised ASSOM was proposed by Ruiz-del-Solar in [7].

The traditional learning procedure of the ASSOM involves computations related to a rotation operator matrix, which not only is memory demanding, but also has a computational load quadratic to the input dimension, i.e. the dimension of input vectors. Therefore, this algorithm in its original form is costly for practical applications, especially for image processing, where input patterns are often high dimensional. In order to reduce the learning time, the Adaptive Subspace Map (ASM) proposed by De Ridder *et al*. [8] drops topological ordering and performs a batch-mode updating of the subspaces with PCA. However, without topological ordering, it is no longer ASSOM. López-Rubio *et al*. [10] proposed the PCASOM by combining PCA with ASSOM, which runs about twice faster than the basic ASSOM under similar classification performance. López-Rubio *et al*. [11] proposed two new learning rules of the ASSOM based on a gradient-based approach or on the Levenberg-Marquardt method. The new rules converge faster than the traditional ASSOM. However, the gradient-based approach showed obvious oscillations in the learning curve. The Levenberg-Marquardt method, on the other hand, has a computational load cubic with the input dimension, which excludes its use for high-dimensional inputs. McGlinchey *et al*. [12] replaced the traditional basis vector updating formula with one proposed by Oja [13], where the computational load is only linear to the input dimension, but quadratic to the subspace dimension. The above-mentioned methods can beat the ASSOM realized in the traditional way in terms of learning speed, but not the fast implementations of the ASSOM to be discovered in this paper.

In this paper, we shall show that in the ASSOM learning, the increment of each basis vector is a scaling of the component vectors of the input episode, which leads to a computational load linear to both the input dimension and the subspace dimension. We reveal that the rotation operator proposed in the original ASSOM approximately maximizes another objective function. The related in-depth analysis leads to a batch-mode

Manuscript received June 20, 2006. This work was carried out during the tenure of a MUSCLE Internal fellowship (http://www.muscle-noe.org).

H. Zheng is with Media Processing and Communication Lab, Department of Electronics and Communication Engineering, Sun Yat-sen University, 510275 Guangzhou, China (email: zhenghch@mail.sysu.edu.cn)

G. Lefebvre and C. Laurent are with France Telecom R&D - TECH/IRIS/CIM, 4 Rue du Clos Courtel, 35512 Cesson Sévigné Cedex, France (email: {gregoire.lefebvre, christophe2.laurent}@orange-ftgroup.com)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

updating of the basis vectors, where the increment of each basis vector is a linear combination of component vectors in the episode. This modified operator further accelerates the learning procedure.

In this paper, we are also interested in applying the fast implementations of ASSOM to saliency-based invariant feature construction for image classification. Saliency-based approaches are driven by psychovisual works [14], [15], which have shown that the sensitivity of the human visual system (HVS) is not uniformly distributed across the image content. Such local methods can focus on different "concepts" in the image. Research has become very active on saliency-based approaches [16], [17], [18], [19], [20], [21]. Salient features can be represented by salient regions resulting from an image segmentation step [22], [23], by non-connected image zones resulting from the construction of saliency maps [24], by edges [25], or by special points [26], [27]. In this paper, we are mainly interested in salient points, which provide the most compact representation of the image content by limiting the correlation and redundancy [28]. The corner detector proposed by Harris and Stephens [26] is one of the most used and the most known point detectors. It is based on the computation of a local auto-correlation function at each pixel location. Eigenvalues of the Hessian matrix of this function are used as indicator of presence of corners. In [28], Laurent *et al*. proposed a wavelet-based salient point detector, which aims to address the following two observations: 1) contours are more perceptually important than other point related features such as corners; 2) salient points detected by a corner detector may be gathered in small image regions in the case of textured or noisy images, resulting in a very local image description. This wavelet-based detector reaches photometric invariance by incorporating a color invariance method proposed in [29].

Various descriptors have been studied for saliency-based approaches [30], [21]. Lowe proposed the Scale Invariant Feature Transform (SIFT) for transform of image data into scale-invariant local features [30]. The SIFT descriptor was used in [31] for image classification, where images are represented by "bags of keypoints" through clustering the SIFT descriptors and the Support Vector Machine (SVM) is finally implemented for classification. Ideally, salient features should be robust to geometric transformations, slight changes of viewpoint and variations of imaging conditions, which is hard for handcrafted feature extractors. The ASSOM is a suitable tool to deal with transformations by learning pattern subspaces. The previous applications of the ASSOM to invariant feature extraction [7], [8], [9] were not based on saliency approaches. In this paper, we explore a sequence of two schemes to apply the ASSOM to invariant feature construction for image classification under a saliency framework. In the first scheme, a single ASSOM is trained on all image patches extracted at the salient points from the training set by using the wavelet-based detector proposed in [28]. An image is represented by a cumulative activity map through projecting the patches extracted from the image on this ASSOM. In the second scheme, one ASSOM per class is trained and an image is represented by a joint cumulative activity map calculated from these ASSOMs. For both schemes, the SVM will then be implemented for classification. It may be interesting that in [31], feature invariance is obtained by using SIFT descriptors before feature clustering, whereas in this paper, feature invariance is achieved at the clustering stage, by using the ASSOM.

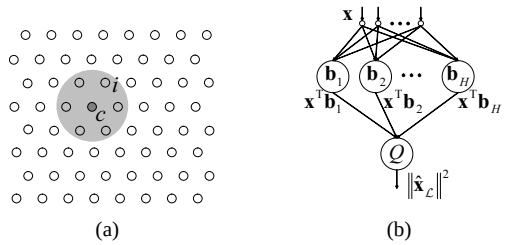

Fig. 1. (a) An ASSOM lattice with a hexagonal neighborhood. Each empty circle in the lattice represents a neural module as shown in (b). The gray region represents the neighborhood area of the winning module indexed by c at some learning step. (b) Neural architecture of a module in the ASSOM.

This paper is organized as follows. In Section II, we discuss the traditional ASSOM learning algorithm and present the alternative fast-learning implementations. The performance in terms of learning speed will be demonstrated by experiments. In Section III, we will apply the ASSOM to saliency-based invariant feature construction for image classification. The experimental results will be presented in Section IV. This paper will be concluded by Section V, which summarizes main points in the paper and give some perspectives.

II. Basis Learning Rules of the ASSOM

A. Basic ASSOM Learning

An ASSOM is composed of an array of modules with each one being realized by a two-layer neural network [2], as shown in Fig. 1. In the two dimensional case, a lattice with a rectangular or hexagonal neighborhood is often used as the layout of modules, as in the SOM. Fig. 1(a) shows an ASSOM lattice with a hexagonal neighborhood. The architecture of a neural module of ASSOM is illustrated in Fig. 1(b). Let $\mathcal{L}$ be a subspace spanned by H orthonormal basis vectors $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_H\}$ and $\mathbf{x}$ be an input vector. The neurons in the first layer compute the orthogonal projections $\mathbf{x}^{\mathrm{T}}\mathbf{b}_h$ of $\mathbf{x}$ on the individual basis vectors $\mathbf{b}_h$, for $h = 1, \ldots, H$. The only quadratic neuron of the second layer sums up the squared outputs of the first-layer neurons. The output of the module is then $\|\hat{\mathbf{x}}_{\mathcal{L}}\|^2$, with $\hat{\mathbf{x}}_{\mathcal{L}}$ being the orthogonal projection of $\mathbf{x}$ on $\mathcal{L}$. It can be regarded as a measure of the matching between the input vector $\mathbf{x}$ and the subspace $\mathcal{L}$. For an input episode $\mathbf{X} = \{\mathbf{x}(s), s \in S\}$, where S is the index set of vectors in the episode, Kohonen proposed to use the *energy* $\sum_{s\in S} \|\hat{\mathbf{x}}_{\mathcal{L}}(s)\|^2$ as the measure of matching between $\mathbf{X}$ and $\mathcal{L}$ [2].

Modules in the ASSOM compete with each other based on their output energies. Once the winner of the modules is determined, the winning module and its neighbors update their subspaces to better represent the input subspace. A neighborhood function $h_c^{(i)}$ is defined on the lattice of modules, where c indicates the index of the winning module and i the index of an arbitrary module. $h_c^{(i)}$ is a function of learning step and the area of neighborhood shrinks with the learning step.

The classical Kohonen's ASSOM learning algorithm can be summarized as follows. At the learning step t,

1) Input the episode $\mathbf{x}(s)$, $s \in S$. Locate the winning module indexed by $c = \arg\max_{i\in I} \sum_{s\in S} \|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|^2$, where I is the set of indices of modules in the ASSOM.
2) For each module i in the neighborhood of c, including c itself, and for each input vector $\mathbf{x}(s)$, update the basis vectors $\mathbf{b}_h^{(i)}$, according to the following procedure:
   a) Rotate each basis vector according to:
   $$\mathbf{b}_h^{(i)} = \mathbf{P}_c^{(i)}(s,t)\mathbf{b}_h^{'(i)} \ , \tag{1}$$
   where $\mathbf{b}_h^{(i)}$ is the new basis vector after rotation and $\mathbf{b}_h^{'(i)}$ the old one. $\mathbf{P}_c^{(i)}(s,t)$ is a rotation operator matrix defined by:
   $$\mathbf{P}_c^{(i)}(s,t) = \mathbf{I} + \lambda(t) h_c^{(i)}(t) \frac{\mathbf{x}(s)\mathbf{x}^{\mathrm{T}}(s)}{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|\|\mathbf{x}(s)\|} \ . \tag{2}$$
   $\lambda(t)$ is a learning-rate factor that diminishes with t. For the algorithm to converge, $\lambda(t)$ should satisfy $\sum_t \lambda(t) = \infty$ and $\sum_t \lambda^2(t) < \infty$ [32].
   b) Dissipate the components $b_{hj}^{(i)}$ of the basis vectors $\mathbf{b}_h^{(i)}$ to improve the stability of the results [2]: $\tilde{b_{hj}^{(i)}} = \mathrm{sgn}(b_{hj}^{(i)}) \max(0, |b_{hj}^{(i)}| - \varepsilon)$, where ε is a small positive value.
   c) Orthonormalize the basis vectors in module i.

A naive implementation of (1) and (2) requires a matrix multiplication which needs not only a large amount of memory, but also a computational load quadratic to the input dimension. It would be costly for practical applications.

B. Insight on the Basis Vector Rotation

In this section, we propose an alternative implementation of the basis updating rule in the ASSOM learning. In the first place we propose to reformulate (1) and (2). The term $\mathbf{b}_h^{'(i)}$ in (1) can be distributed to the right side of (2), leading to:

$$\mathbf{b}_h^{(i)} = \mathbf{b}_h^{'(i)} + \Delta\mathbf{b}_h^{(i)} \ , \tag{3}$$

where

$$\Delta\mathbf{b}_h^{(i)} = \lambda(t) h_c^{(i)}(t) \frac{\mathbf{x}(s)\mathbf{x}^{\mathrm{T}}(s)\mathbf{b}_h^{'(i)}}{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|\|\mathbf{x}(s)\|} \ . \tag{4}$$

$\mathbf{x}^{\mathrm{T}}(s)\mathbf{b}_h^{'(i)}$ is in fact a scalar value. The equation becomes:

$$\Delta\mathbf{b}_h^{(i)} = \alpha_{c,h}^{(i)}(s,t)\mathbf{x}(s) \ . \tag{5}$$

Here $\alpha_{c,h}^{(i)}(s,t)$ is a scalar value defined by:

$$\alpha_{c,h}^{(i)}(s,t) = \lambda(t) h_c^{(i)}(t) \frac{\mathbf{x}^{\mathrm{T}}(s)\mathbf{b}_h^{'(i)}}{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|\|\mathbf{x}(s)\|} \ . \tag{6}$$

This shows that the increment $\Delta\mathbf{b}_h^{(i)}$ is in fact a scaling of the component vector $\mathbf{x}(s)$, as illustrated in Fig. 2, which seems to have been ignored by many practitioners. Careful examination of (5) would reveal similarity of this formula with a recursive PCA suggested in [33]. The main difference here is that the gain of stochastic approximation is modulated by a neighborhood function dependent on module competition.

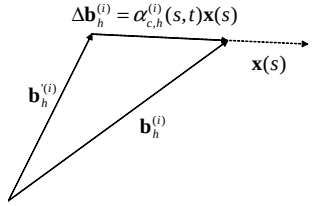

Fig. 2. An alternative view of the basis vector updating rule of ASSOM.

Note that in (6), $\mathbf{x}^{\mathrm{T}}(s)\mathbf{b}_h^{'(i)}$ is the projection of the component vector on the basis vectors represented by the neurons of the first layer, which we have already when computing the projection $\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|$ (cf. Fig. 1). If we calculate the scaling factor $\alpha_{c,h}^{(i)}(s,t)$ first, and then scale the component vector $\mathbf{x}(s)$ with this factor, the computations associated with the basis vector updating will be dramatically reduced. This implementation will be referred to as FL-ASSOM for fast-learning ASSOM. It is completely equivalent to the basic ASSOM in terms of generating topologically ordered invariant-feature filters.

Now we compare the computational loads of the basis vector updating in the basic ASSOM and in the FL-ASSOM. Let N be the input dimension. It is not hard to verify that a naive implementation of the updating rule defined by the matrix multiplications in (1) and (2) would need about $HN^2 + N^2$ scalar multiplications and about the same number of scalar additions. So the computational load is approximately $O(HN^2)$, i.e. quadratic to the input dimension and linear to the subspace dimension. The replacement proposed by McGlinchey *et al*. [12] leads to a computational load of $O(H^2N)$, i.e. linear to the input dimension but quadratic to the subspace dimension. Now with the proposed updating rule, the computations of $\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|$ and $\|\mathbf{x}(s)\|$ in (6) need about $HN + 2N$ multiplications, and $\alpha_{c,h}^{(i)}(s,t)\mathbf{x}(s)$ in (5) about HN multiplications. In all (5)and (6) need about $2HN + 2N$ multiplications. Similarly, the number of additions can be shown to be about $2HN + 2N$. So with (5)and (6), the computational load is approximately $O(HN)$, i.e. linear to both the input dimension and the subspace dimension. The methods proposed in [8], [10], [11], [12] can beat the classical implementation of the ASSOM, but not the implementation proposed here.

C. Discussion on the Operator $\mathbf{P}_c^{(i)}(s,t)$

The objective function of the ASSOM is defined as:

$$E = \int \sum_i h_c^{(i)} \sum_{s\in S} \frac{\|\tilde{\mathbf{x}}_{\mathcal{L}_i}(s)\|^2}{\|\mathbf{x}(s)\|^2} P(\mathbf{X}) \mathrm{d}\mathbf{X} \tag{7}$$

where $P(\mathbf{X})$ is the distribution function of the random episode $\mathbf{X}$. $\tilde{\mathbf{x}}_{\mathcal{L}_i}(s) = \mathbf{x}(s) - \hat{\mathbf{x}}_{\mathcal{L}_i}(s)$ is the residual of $\mathbf{x}(s)$ after projection on $\mathcal{L}_i$. In the following we review briefly the main steps in the derivation of $\mathbf{P}_c^{(i)}(s,t)$ from (7) and prove that $\mathbf{P}_c^{(i)}(s,t)$ approximately maximizes another objective function.

By using the Robbins-Monro stochastic approximation [32], a *sample objective function* has been aimed at:

$$E_{\mathrm{s}}(t) = \sum_i h_c^{(i)}(t) \sum_{s\in S} \frac{\|\tilde{\mathbf{x}}_{\mathcal{L}_i}(s)\|^2}{\|\mathbf{x}(s)\|^2} \tag{8}$$

Kohonen showed in [2] that

$$\frac{\partial E_{\rm s}}{\partial \mathbf{b}_h^{(i)}}(t) = -2h_c^{(i)}(t)\sum_{s\in S}\frac{\mathbf{x}(s)\mathbf{x}^{\rm T}(s)}{\|\mathbf{x}(s)\|^2}\mathbf{b}_h^{(i)} \ . \quad (9)$$

Moving $\mathbf{b}_h^{(i)}$ by a step length $\frac{1}{2}\lambda(t)$ in the negative direction of this gradient, since $\lambda(t)$ is small, we have:

$$\mathbf{b}_h^{(i)} = \left[\mathbf{I}+\lambda(t)h_c^{(i)}(t)\sum_{s\in S}\frac{\mathbf{x}(s)\mathbf{x}^{\rm T}(s)}{\|\mathbf{x}(s)\|^2}\right]\mathbf{b}_h^{'(i)} \quad (10)$$

$$\approx \prod_{s\in S}\left[\mathbf{I}+\lambda(t)h_c^{(i)}(t)\frac{\mathbf{x}(s)\mathbf{x}^{\rm T}(s)}{\|\mathbf{x}(s)\|^2}\right]\mathbf{b}_h^{'(i)} \ . \quad (11)$$

This approximation amounts to a successive rotation of the basis for each component vector in the episode.

For stability of the solution, Kohonen proposed to multiply the learning rate $\lambda(t)$ by $\frac{\|\mathbf{x}(s)\|}{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|}$ and obtained the following slightly modified rotation operator:

$$\mathbf{M}_c^{(i)}(t) = \prod_{s\in S}\left[\mathbf{I}+\lambda(t)h_c^{(i)}(t)\frac{\mathbf{x}(s)\mathbf{x}^{\rm T}(s)}{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|\|\mathbf{x}(s)\|}\right] \quad (12)$$

$$= \prod_{s\in S}\mathbf{P}_c^{(i)}(s,t) \ . \quad (13)$$

That is how the operator $\mathbf{P}_c^{(i)}(s,t)$ was developed in the classical ASSOM basis updating.

In fact, the operator $\mathbf{P}_c^{(i)}(s,t)$ approximately maximizes another objective function as we shall discover now:

$$E_{\rm m} = \int\sum_i h_c^{(i)}\sum_{s\in S}\frac{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|}{\|\mathbf{x}(s)\|}P(\mathbf{X}){\rm d}\mathbf{X} \ . \quad (14)$$

Proof: The sample function of $E_{\rm m}$ is:

$$E_{\rm ms}(t) = \sum_i h_c^{(i)}(t)\sum_{s\in S}\frac{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|}{\|\mathbf{x}(s)\|} \ . \quad (15)$$

Taking the gradient of $E_{\rm ms}(t)$, we have

$$\frac{\partial E_{\rm ms}}{\partial \mathbf{b}_h^{(i)}}(t) = h_c^{(i)}(t)\sum_{s\in S}\frac{\partial\left(\frac{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|}{\|\mathbf{x}(s)\|}\right)}{\partial \mathbf{b}_h^{(i)}} \ . \quad (16)$$

Moreover,

$$\frac{\partial\left(\frac{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|^2}{\|\mathbf{x}(s)\|^2}\right)}{\partial \mathbf{b}_h^{(i)}} = 2\frac{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|}{\|\mathbf{x}(s)\|}\frac{\partial\left(\frac{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|}{\|\mathbf{x}(s)\|}\right)}{\partial \mathbf{b}_h^{(i)}} \ . \quad (17)$$

So,

$$\frac{\partial E_{\rm ms}}{\partial \mathbf{b}_h^{(i)}}(t) = h_c^{(i)}(t)\sum_{s\in S}\frac{1}{2}\frac{\|\mathbf{x}(s)\|}{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|}\frac{\partial\left(\frac{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|^2}{\|\mathbf{x}(s)\|^2}\right)}{\partial \mathbf{b}_h^{(i)}} \ . \quad (18)$$

We have

$$h_c^{(i)}(t)\sum_{s\in S}\frac{\partial\left(\frac{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|^2}{\|\mathbf{x}(s)\|^2}\right)}{\partial \mathbf{b}_h^{(i)}} \quad (19)$$

$$= \frac{\partial\left(\sum_l h_c^{(l)}(t)\sum_{s\in S}1 - E_{\rm s}(t)\right)}{\partial \mathbf{b}_h^{(i)}} \quad (20)$$

$$= -\frac{\partial E_{\rm s}}{\partial \mathbf{b}_h^{(i)}}(t) = 2h_c^{(i)}(t)\sum_{s\in S}\frac{\mathbf{x}(s)\mathbf{x}^{\rm T}(s)}{\|\mathbf{x}(s)\|^2}\mathbf{b}_h^{(i)} \ . \quad (21)$$

So

$$\frac{\partial E_{\rm ms}}{\partial \mathbf{b}_h^{(i)}}(t) = h_c^{(i)}(t)\sum_{s\in S}\frac{\|\mathbf{x}(s)\|}{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|}\frac{\mathbf{x}(s)\mathbf{x}^{\rm T}(s)}{\|\mathbf{x}(s)\|^2}\mathbf{b}_h^{(i)} \quad (22)$$

$$= h_c^{(i)}(t)\sum_{s\in S}\frac{\mathbf{x}(s)\mathbf{x}^{\rm T}(s)}{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|\|\mathbf{x}(s)\|}\mathbf{b}_h^{(i)} \ . \quad (23)$$

Taking a step $\lambda(t)$ in the direction of this gradient, we get the rotation matrix

$$\mathbf{B}_c^{(i)}(t) = \mathbf{I}+\lambda(t)h_c^{(i)}(t)\sum_{s\in S}\frac{\mathbf{x}(s)\mathbf{x}^{\rm T}(s)}{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|\|\mathbf{x}(s)\|} \ . \quad (24)$$

When $\lambda(t)$ is small, it is equivalent to $\mathbf{M}_c^{(i)}(t)$ as in (12). ■

D. Further Boosting: Batch-mode Basis Vector Updating

Basis vector updating can be further boosted by working in a batch mode. We can avoid computing the value of $\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|$ in (6) by using the value computed previously during module competition. However, this could not be done inside the framework of FL-ASSOM since the subspaces are continuously changing in receiving each component vector of the episode. To save computation of $\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|$, the batch-mode rotation operator $\mathbf{B}_c^{(i)}(t)$ in (24) will be useful.

As was done for the FL-ASSOM, by distributing $\mathbf{b}_h^{'(i)}$ to terms in the operator $\mathbf{B}_c^{(i)}(t)$, the basis vector updating rule becomes:

$$\mathbf{b}_h^{(i)} = \mathbf{b}_h^{'(i)} + \Delta\mathbf{b}_h^{(i)} \ , \quad (25)$$

where

$$\Delta\mathbf{b}_h^{(i)} = \sum_{s\in S}\left(\alpha_{c,h}^{(i)}(s,t)\mathbf{x}(s)\right) \ . \quad (26)$$

The increment of each basis vector is thus a linear combination of component vectors in the episode. The difference between the updating rules (3) and (25) is that the former updates the basis vectors for each component vector one by one while the latter updates the basis vectors in a batch mode for the whole episode.

The scalar parameter $\alpha_{c,h}^{(i)}(s,t)$ has the same form as (6):

$$\alpha_{c,h}^{(i)}(s,t) = \lambda(t)h_c^{(i)}(t)\frac{\mathbf{x}^{\rm T}(s)\mathbf{b}_h^{'(i)}}{\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|\|\mathbf{x}(s)\|} \ . \quad (27)$$

The meaning of this equation is, however, a little different from that of (6). Here in (27) the basis vector updating is performed only after the whole episode has been received. Therefore, $\|\hat{\mathbf{x}}_{\mathcal{L}_i}(s)\|$ and $\mathbf{x}^{\rm T}(s)\mathbf{b}_h^{'(i)}$ can reuse the results previously calculated during module competition. What we need to do is only store the calculated values in registers and fetch them when needed. The computational load of (27) is thus trivial. Furthermore, the dissipation as well as orthonormalization of basis vectors can be performed only once for each episode without loosing accuracy since the basis vectors are not updated during the episode. The computational load can thus be further reduced. This method will be referred to as BFL-ASSOM for batch-mode fast-learning ASSOM.

Let us estimate the computational load of the BFL-ASSOM averaged on each component vector of the episode as we

did for the basic ASSOM and for the FL-ASSOM. As has been mentioned, the calculation of $\alpha_{c,h}^{(i)}(s,t)$ according to (27) needs only trivial computation. The majority of computation is in (26). Averaged on each vector in the episode, the computational load required by basis vector updating with the BFL-ASSOM is about HN multiplications and HN additions. Furthermore, since the dissipation and orthonormalization of basis vectors can be performed only once for each episode, the whole learning time can be further reduced.

E. Experiments

We first demonstrate by experiments that the BFL-ASSOM can generate topologically ordered invariant-feature filters as the basic ASSOM. The results of FL-ASSOM will be shown as the ground truth since the FL-ASSOM is mathematically equivalent to the basic ASSOM. Kohonen has shown that the ASSOM can generate basis vectors similar to Gabor filters for episodes subject to translation [1], [2]. We shall show that the BFL-ASSOM is able to generate similar filters.

The input episodes are constructed from a colored noise image, which is generated by filtering a white noise image with a second-order Butterworth filter. The cut-off frequency is set to 0.6 times the Nyquist frequency of the sampling lattice. Each episode is composed of 6 vectors, each of which is formed on a circular receptive field composed of 349 pixels. The vectors in the same episode have only random translation of no more than 5 pixels in both the horizontal and the vertical directions. The episodes are generated on random locations of the colored noise image. The mean value of components of each input vector is subtracted from each component of the vector. In order to symmetrize the filters with respect to the center of the receptive field, the input samples are weighted by a Gaussian function symmetrically placed at the center of the receptive field with a full width at half maximum (FWHM) that varies linearly with respect to the learning step t from 1 to 16 sampling lattice spacings. Each vector is normalized to a unit vector before being sent to the ASSOM.

The ASSOM array is composed of 9×10 modules aligned in a hexagonal lattice with two basis vectors at each module. The basis vectors of all the modules are initially randomized and orthonormalized. The radius of the circular neighborhood function $h_c^{(i)}(t)$ decreases linearly from 6.73 ($= 0.5 \times (9^2 + 10^2)^{1/2}$) to 0.9 ASSOM array spacings with t. The learning-rate factor $\lambda(t) = 0.1 \cdot T/(T+99t)$, where T is the total number of learning steps and set to $30,000$ for the current experiment.

As shown in Fig. 3(a), the translation-invariant filters generated by the BFL-ASSOM and those by the FL-ASSOM are similar. The difference is only the different organization of the filters due to random initialization of the two networks. For both networks, the formed filters are similar to Gabor filters of different frequencies and different orientations. Moreover, filters of similar frequencies and orientations are formed at nearby sites. For either network, filters corresponding to $\mathbf{b}_1$ and those corresponding to $\mathbf{b}_2$ have the same frequencies at the same locations but 90 degrees of phase difference, which confirms orthogonality of the corresponding basis vectors.

Fig. 3(b) shows how the average projection error e changes with the learning step t for either network. For each input

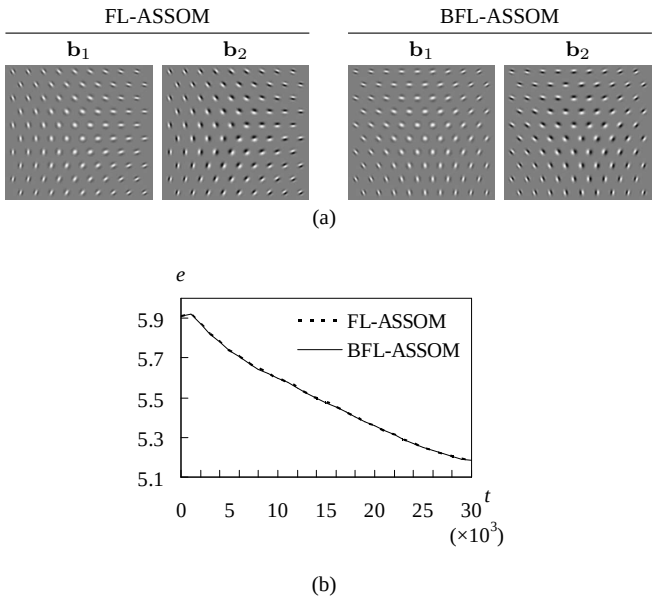

Fig. 3. (a) The Gabor-like filters generated by the BFL-ASSOM compared to those by the FL-ASSOM on episodes subject to translation. $\mathbf{b}_1$: First basis vectors. $\mathbf{b}_2$: Second basis vectors. (b) Changing of the projection error e with the learning step t for the FL-ASSOM and for the BFL-ASSOM.

episode $\mathbf{X} = \{\mathbf{x}(s), s \in S\}$, where $\mathbf{x}(s)$ are mean-subtracted and normalized, the projection error $e(\mathbf{X}) = \sum_{s \in S} \|\mathbf{x}(s) - \hat{\mathbf{x}}(s)\|^2$, where $\hat{\mathbf{x}}(s)$ is the projection of $\mathbf{x}(s)$ on the subspace of the winning module. e in the figure is the average of $e(\mathbf{X})$ over all the training episodes. Fig. 3(b) confirms that the difference between the learning curve of the FL-ASSOM and that of the BFL-ASSOM is practically negligible.

In the second experiment, we compare the computational loads of the basic ASSOM, the FL-ASSOM and the BFL-ASSOM with respect to the input dimension N and the subspace dimension H. We record the elapsed CPU seconds for each method. The number of iterations are fixed to $1,000$. Each episode is composed of 6 vectors, which are generated randomly according to a uniform probability distribution. The rectangular ASSOM array contains 10×10 modules.

The timing results obtained by using C++ implementations are summarized in Table I, where the means and standard deviations of the elapsed CPU times after 20 runs with different initializations are recorded. As was anticipated, the basis vector updating time of the basic ASSOM increases rapidly with the input dimension and is the bottleneck of the learning procedure. With the FL-ASSOM, the basis vector updating time is dramatically reduced, increasing slowly with the input dimension and with the subspace dimension. It is no longer a bottleneck of the learning procedure. However, learning time outside basis vector updating is not reduced. Now with the BFL-ASSOM, the basis vector updating time is further reduced. Moreover, learning time outside the basis vector updating is also reduced considerably compared to the other two networks. Thus, the whole learning time decreases from $1,956$ seconds with the basic ASSOM to 16.2 seconds with the BFL-ASSOM when $H = 4$ and $N = 400$.

The relationship between the basis vector updating time

TABLE I

TIMING RESULTS OF THE BASIC ASSOM, THE FL-ASSOM AND THE BFL-ASSOM WITH RESPECT TO THE INPUT DIMENSION N AND THE SUBSPACE DIMENSION H. LEFT SUB-TABLE: THE BASIS VECTOR UPDATING TIME (VU). μ REPRESENTS THE MEAN VALUE AFTER 20 RUNS. σ REPRESENTS THE CORRESPONDING SAMPLE STANDARD DEVIATION; RIGHT SUB-TABLE: THE WHOLE LEARNING TIME (WL), WHICH INCLUDES THE TIME FOR MODULE COMPETITION, BASIS VECTOR UPDATING, BASIS VECTOR DISSIPATION AND ORTHONORMALIZATION. ALL THE TIMES ARE GIVEN IN SECONDS

		H=2		H=3		H=4	
		μ_{VU}	σ_{VU}	μ_{VU}	σ_{VU}	μ_{VU}	σ_{VU}
ASSOM	N=100	78.0	0.54	96.8	0.73	120	4.7
	N=200	303	2.1	377	3.2	449	2.7
	N=300	682	3.8	846	7.9	1,003	7.6
	N=400	1,331	11	1,621	13	1,904	13
FL	N=100	0.942	0.083	1.26	0.097	1.60	0.10
	N=200	1.46	0.13	2.01	0.16	2.48	0.10
	N=300	2.10	0.12	2.80	0.20	3.60	0.18
	N=400	2.80	0.26	3.73	0.28	4.66	0.20
BFL	N=100	0.488	0.059	0.699	0.071	0.972	0.071
	N=200	0.569	0.056	0.817	0.093	1.12	0.14
	N=300	0.745	0.085	1.05	0.091	1.46	0.095
	N=400	0.930	0.10	1.46	0.10	1.84	0.12

		H=2		H=3		H=4	
		μ_{WL}	σ_{WL}	μ_{WL}	σ_{WL}	μ_{WL}	σ_{WL}
ASSOM	N=100	84.0	0.57	107	0.75	134	4.8
	N=200	314	2.2	395	3.2	475	3.0
	N=300	699	3.8	872	8.0	1,042	7.8
	N=400	1,354	11	1,658	13	1,956	13
FL	N=100	6.92	0.057	10.9	0.086	15.6	0.13
	N=200	12.5	0.14	19.9	0.19	28.4	0.26
	N=300	18.5	0.13	29.2	0.23	41.9	0.43
	N=400	24.3	0.27	38.7	0.32	55.1	0.37
BFL	N=100	2.50	0.044	3.62	0.032	4.82	0.037
	N=200	4.35	0.057	6.28	0.065	8.42	0.076
	N=300	6.34	0.064	9.19	0.071	12.3	0.085
	N=400	8.39	0.093	12.1	0.10	16.2	0.075

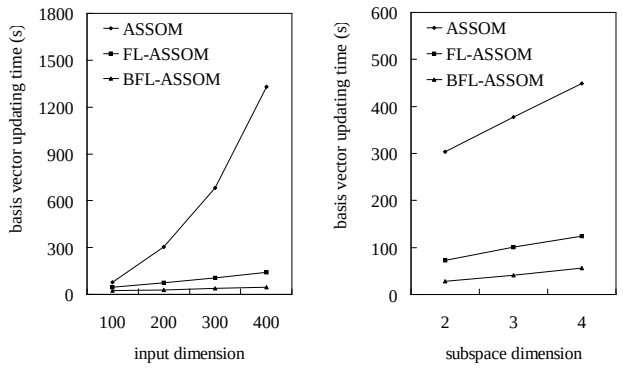

Fig. 4. The basis vector updating time with respect to (a) the input dimension at the subspace dimension $H = 2$ and (b) the subspace dimension at the input dimension $N = 200$. For sake of clarity, the updating time of the FL-ASSOM and that of the BFL-ASSOM are magnified by a factor of 50.

and the input dimension or the subspace dimension for the three implementations of ASSOM is visualized in Fig. 4. The basis vector updating time increases approximately linearly with respect to the input dimension for the FL-ASSOM and for the BFL-ASSOM, but apparently nonlinearly for the basic ASSOM. In all the cases, the updating time increases approximately linearly with respect to the subspace dimension.

III. SALIENCY-BASED INVARIANT FEATURE CONSTRUCTION FOR IMAGE CLASSIFICATION

In this section, we explore a sequence of two schemes where the ASSOM is applied to saliency-based invariant feature construction for image classification. The implemented ASSOM may be the FL-ASSOM or the BFL-ASSOM. We will compare their performance in Section IV.

A. Salient-Point Single ASSOM Scheme (SPSAS)

The first scheme to be explored is based on a single ASSOM as shown in Fig. 5. Salient points are first detected from the input image by using the wavelet-based detector proposed in [28]. Working with wavelets is justified by the consideration of the HVS for which multi-resolution, orientation and frequency analysis is of prime importance. In order to extract the salient points, a wavelet transform is firstly performed on the grayscale image. The obtained wavelet coefficients are represented by a zerotree structure [34]. This tree is then scanned at a first time from leaves to the root to compute the saliency value at each node. Afterwards, this tree is scanned for the second time from the root to leaves in order to determine the salient path from the root to raw salient points on the original image. By working with grayscale images, the points located on boundaries of highlights or shadows are apt to be detected as salient whereas they are only caused by illumination conditions. To remove such false salient points, a gradient image is calculated by using the color invariants proposed by Geusebroek *et al.* [29]. A threshold is then set to select the most salient points.

The ASSOM shall be trained to generate the appropriate feature filters based on the local regions (patches) around these salient points, which are supposed to carry essential information for image description and consequently for classification. The ASSOM can work on episodes composed of several component vectors, but construction of episodes is not necessary in a general situation if we want the ASSOM to learn subspaces. Although it is possible to construct episodes by artificially introducing transformations such as translation, rotation or scaling as in [8], this process could generate artificial variants which might not really exist in the test set. So we prefer to use raw image patches to train the ASSOM.

Let p_k, $k \in \{1, 2, \ldots, K\}$ be K salient points detected from the image $\mathcal{I}$, which amount to K patches $\mathbf{x}_k$. These patches are fed into a single ASSOM with $|I|$ modules, which was previously trained in an unsupervised way on all patches extracted from all categories of images in the training set. For each patch $\mathbf{x}_k$, the module i ($i \in I$) generates an energy $\|\hat{\mathbf{x}}_{k\mathcal{L}_i}\|^2$, with $\hat{\mathbf{x}}_{k\mathcal{L}_i}$ being the orthogonal projection of $\mathbf{x}_k$ on the subspace $\mathcal{L}_i$ of the module i. Energies generated by all the modules construct an activity map, which is a vector

$$\mathbf{a}_k = \left[\|\hat{\mathbf{x}}_{k\mathcal{L}_1}\|^2 \ \cdots \ \|\hat{\mathbf{x}}_{k\mathcal{L}_i}\|^2 \ \cdots \ \|\hat{\mathbf{x}}_{k\mathcal{L}_{|I|}}\|^2\right]^{\mathrm{T}} . \qquad (28)$$

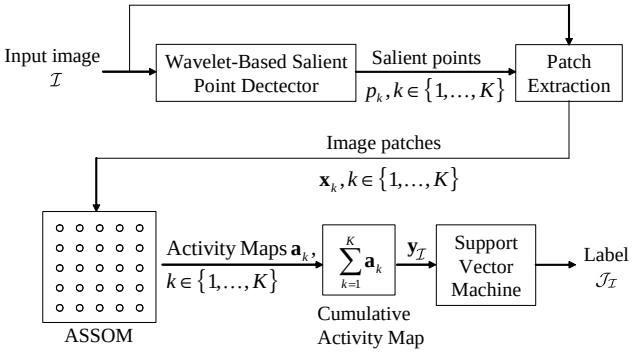

Fig. 5. SPSAS image classification architecture.

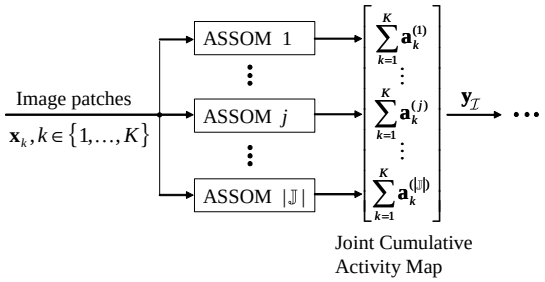

Fig. 6. Construction of the joint cumulative activity map in the SPMAS.

We remark that this activity map construction corresponds to a soft (or fuzzy) clustering process in the sense that, in every module of the ASSOM, the input patch has a membership defined by the energy. It has been shown that a fuzzy clustering process has advantages over a hard one [35]. Activity maps of all patches extracted from $\mathcal{I}$ are then accumulated to form a cumulative activity map $\mathbf{y}_\mathcal{I}$ characterizing the image $\mathcal{I}$, which is defined by:

$$\mathbf{y}_\mathcal{I} = \sum_{k=1}^{K} \mathbf{a}_k \quad . \tag{29}$$

This feature vector is then classified by an SVM previously trained on feature vectors of the training images. The implementation of the SVM used in this paper is part of a publicly available machine learning tool collection WEKA [36]. Here we use a Gaussian kernel $G(\mathbf{u},\mathbf{v}) = \exp(-\alpha\|\mathbf{u}-\mathbf{v}\|^2)$, where $\alpha = 0.02$ in our experiments.

B. Salient-Point Multi-ASSOM Scheme (SPMAS)

The SPSAS does not make use of the fact that we have the label information for images in the training set. It might lead to map modules that mix up different categories of features and confuse the (SVM) classifier. A better strategy would be to use a specific ASSOM for each category. This idea was explored in [9] for recognition of handwritten digits and produced promising results. But the size of images in their case is very small (25×20 pixels), permitting a direct ASSOM learning. In our case, the images have much larger sizes and cannot be directly dealt with under their framework.

The SPMAS replaces the single ASSOM in the SPSAS with an array of ASSOMs, each one being trained for one category of image patches. Separate ASSOMs for different categories permit the system to learn the individual feature sets more precisely than a single ASSOM for all categories. Let $\mathbb{J}$ be the set of image labels (categories), $\mathbf{c}_j = \sum_{k=1}^{K} \mathbf{a}_k^{(j)}$, $j \in \mathbb{J}$ be the cumulative activity map generated by the j-th ASSOM. The new feature vector $\mathbf{y}_\mathcal{I}$ is a joint cumulative activity map formed by combining the $|\mathbb{J}|$ cumulative activity maps:

$$\mathbf{y}_\mathcal{I} = \left[\mathbf{c}_1^{\mathrm{T}} \ \cdots \ \mathbf{c}_j^{\mathrm{T}} \ \cdots \ \mathbf{c}_{|\mathbb{J}|}^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{30}$$

as illustrated in Fig. 6. The feature vector $\mathbf{y}_\mathcal{I}$ is sent to the SVM for classification, as in the SPSAS.

A similar single-SOM versus multi-SOM framework was proposed in [37] for face recognition. Each face image is firstly partitioned into non-overlapping sub-blocks corresponding to local feature vectors. A single SOM is trained for all classes of feature vectors or a separate SOM is trained for each class. Face images are then represented by sets of SOM weight vectors. A soft k-nearest neighbor ensemble method is proposed to identify unlabeled images. The single-SOM scheme and the multi-SOM scheme showed similar performance in their experiments. A block-to-block comparison is used in their research for face identification, which assumes good calibration between training faces and testing faces. It is not suitable for general-purpose image classification problems, where good calibration is not guaranteed. Also, representation of the feature vectors by the weight vectors may be noisy, a smoothed representation such as the activity maps may be more appropriate.

IV. Experimental Results

A. Multi-Category Classification

In the first experiment, we evaluate our system in terms of multi-category image classification on the SIMPLIcity database [1], which is part of the well known Corel database and has been used to test the SIMPLIcity content based image retrieval system in [23]. The database consists of ten categories including African people and villages (Afr), beaches (Bea), buildings (Bui), buses (Bus), dinosaurs (Din), elephants (Ele), flowers (Flo), food (Foo), horses (Hor), mountains and glaciers (Mou), each containing 100 images of 384×256 pixels. Some representative examples from each category are presented in Fig. 7. The images in each category are divided into two equal parts: 50 for training and the other 50 for testing.

We focus on RGB color features in the experiment. The image patches are circles with 597 pixels (about a radius of 14), which amount to $1,791 = 597 \times 3$ dimensional vectors. The mean value of components of each input vector is subtracted from each component of the vector. The training steps are empirically set to $80,000$ for the ASSOM in the SPSAS and $30,000$ for each ASSOM in the SPMAS, simply because there are more training patches available for the single ASSOM in the SPSAS than for each ASSOM in the SPMAS. We use a rectangular ASSOM lattice. The choice between

[1] http://wang.ist.psu.edu/∼jwang/test1.tar

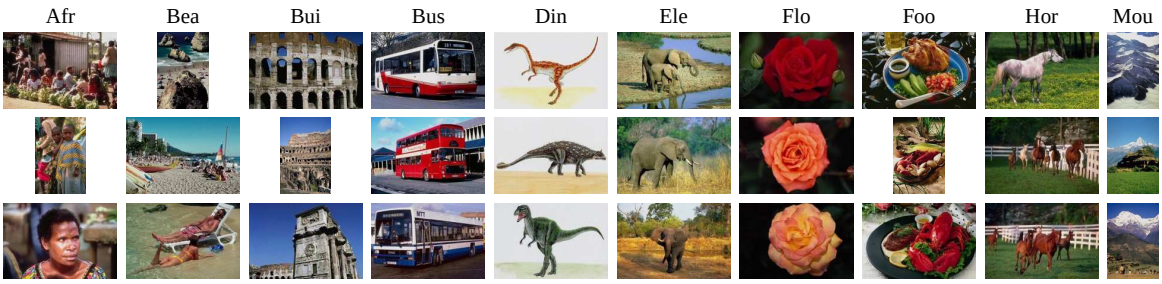

Fig. 7. Example images from the SIMPLIcity database.

TABLE II
COMPARISON OF CLASSFICATION ACCURACIES BETWEEN THE SPSAS AND THE SPMAS, AVERAGED ON THE 10 CATEGORIES

ASSOM size	SPSAS		SPMAS	
	Training	Test	Training	Test
5×5	45.8%	42%	87%	70%
10×10	71.2%	58%	100%	76.2%
15×15	88.4%	56.8%	100%	77%

TABLE III
COMPARISON OF CLASSIFICATION ACCURACIES BETWEEN THE BFL-ASSOM AND THE FL-ASSOM, AVERAGED ON THE 10 CATEGORIES

ASSOM size	FL-ASSOM		BFL-ASSOM	
	Training	Test	Training	Test
5×5	87%	70%	88.2%	73.8%
10×10	100%	76.2%	100%	78.2%
15×15	100%	77%	100%	78%

rectangular or hexagonal lattice does not seem to be critical according to our experiments.

1) SPSAS Versus SPMAS: We first compare the SPMAS to the SPSAS based on the FL-ASSOM. The experimental results are summarized in Table II. Both schemes suffered from the overfitting problem due to lack of data. The SPSAS shows less overfitting than the SPMAS when the size of the ASSOM is small. But overfitting of the SPSAS inflates more quickly than that of the SPMAS when the size of the ASSOM increases. In this sense, the SPSAS is more sensitive to the number of parameters than the SPMAS. The SPMAS shows better classification accuracies than the SPSAS on both the training set and the test set across different configurations, which confirms advantages of using separate ASSOMs to learn features of different categories. On the test set, the SPMAS has an improvement of accuracy of $18.2\% - 28\%$ over the SPSAS. We will stick to the SPMAS in the following experiments.

2) BFL-ASSOM Versus FL-ASSOM: Table III summarizes the classification accuracies of the SPMAS with the BFL-ASSOM and with the FL-ASSOM of various sizes. The number of neurons in the first layer of each ASSOM module is fixed to 2. This table shows that the performance of the BFL-ASSOM is a little better than the FL-ASSOM. The reason could be that the BFL-ASSOM is a more accurate learning process deduced from the corresponding objective function than the FL-ASSOM. Thus the local features could be better structured with the BFL-ASSOM than with the FL-ASSOM. The improvement of the classification accuracy is $1\% - 3.8\%$ on the test set if we replace the FL-ASSOM with the BFL-ASSOM. Taking the learning speed into account, the BFL-ASSOM seems more attractive than the FL-ASSOM. According to Table III the performance of the BFL-ASSOM-based SPMAS is nearly optimal when the ASSOMs are of the size 10×10. We will stick to the 10×10 BFL-ASSOM-based SPMAS in the following experiments.

3) SPMAS Feature Filter Visualization: Fig. 8 shows the feature filters generated from the 10 categories of training images by using the 10×10 BFL-ASSOM-based SPMAS. Two neurons are implemented in the first layer of each module. In this way, each ASSOM learned two 10×10 lattices of basis vectors. In order to show the correspondence between the components of the basis vectors and the R, G, B components of the input patches, each three subsequent components of the basis vectors are grouped back to form an "RGB" pixel. The components are normalized to the range $[0, 255]$ with the value 128 corresponding to a component 0. "Pixels"of the basis vectors are organized to form the same shape as the input patches, i.e. a circle.

Each basis vector can be rotated to the opposite direction without altering the spanned subspace. Thus we could rotate some of the basis vectors in order to get maps where neighboring basis vectors appear more similarly. For example, the orangish basis vectors could be turned to bluish in the $\mathbf{b}_1$ lattice of the elephant category although we did not do that. This would not change the performance of the ASSOM and the consequent classifier.

As we mentioned previously in Section II-B, the ASSOM subspace learning process is similar to a recursive PCA. In the basis vector orthonormalization process, the first basis vectors are only normalized, and the second ones are orthogonalized with respect to the first ones. Thus the first basis vectors are likely to capture the first principal components of the input feature subspaces while the second ones are likely to capture the second principal components. There are some observable characteristics of features corresponding to the various categories of images. For flowers, the first basis vectors show a distinct red tone because most of the flowers, at least in the SIMPLIcity database, have a red tone. For buildings, both basis vectors do not show distinct colors. This is also the case for the dinosaurs since the dinosaur pictures are artificial

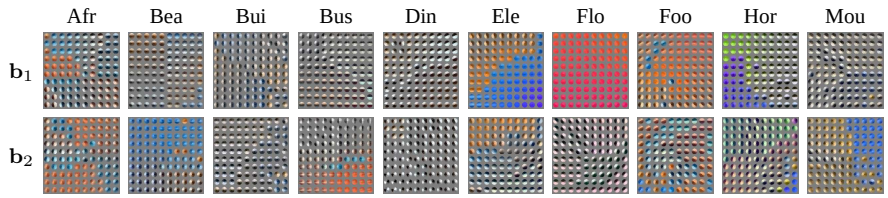

Fig. 8. Feature filters for the 10 categories of the SIMPLIcity training set. $\mathbf{b}_1$: The first basis vectors. $\mathbf{b}_2$: The second basis vectors.

TABLE IV
CLASSIFICATION ACCURACIES OF THE 10×10 BFL-ASSOM-BASED SPMAS WITH DIFFERENT SUBSPACE DIMENSIONS. THE RESULTS ARE AVERAGED ON THE 10 CATEGORIES

	$H=2$	$H=4$	$H=6$	$H=8$	$H=10$
Training	100%	100%	100%	100%	100%
Test	78.2%	78.6%	82.4%	84.6%	86%

TABLE V
CONFUSION MATRIX OF THE SPMAS. THE BOLDFACED FIGURES CORRESPOND TO CORRECTLY CLASSIFIED TEST IMAGES

True Classes ↓	Afr	Bea	Bui	Bus	Din	Ele	Flo	Foo	Hor	Mou
Afr	**43**	2	0	0	0	2	0	2	1	0
Bea	1	**36**	2	0	1	4	1	0	2	3
Bui	1	5	**39**	1	0	2	0	1	0	1
Bus	0	2	2	**46**	0	0	0	0	0	0
Din	0	0	0	0	**50**	0	0	0	0	0
Ele	3	1	3	0	1	**40**	0	0	2	0
Flo	1	0	0	0	1	0	**46**	2	0	0
Foo	2	1	2	0	0	0	1	**44**	0	0
Hor	0	0	0	0	0	0	1	0	**49**	0
Mou	1	8	0	0	1	1	2	0	0	**37**

models without distinct colors.

In Fig. 9, we show three exemplary images and their joint cumulative activity maps, where gray levels of the modules in the ASSOMs are proportional to the energies of the respective modules. For the first two images, the ASSOMs corresponding to the correct image categories are more activated than other ASSOMs. However, this is not always so obvious. For example, the third image strongly activated both the "Bui" ASSOM and the "Bus" ASSOM. In fact, many bus images in the SIMPLIcity database have buildings in the backgrounds. It is not surprising to find out that the "Bus" ASSOM has learned the concept of some building parts. That is why we choose to keep the activity maps of different ASSOMs and use the SVM to make the decision.

4) Effects of the Subspace Dimension: The classification results with the subspace dimension $H = 2, 4, 6, 8, 10$ are summarized in Table IV. From $H = 2$ to $H = 10$, the average accuracy on the test set is improved for 7.8%, which suggests that higher subspace dimensions capture more precisely the variances of input patterns. The price to pay is that a higher subspace dimension involves a heavier computational load. Even though the complexity of the system seems to increase with higher subspace dimensions, the "curse of dimension" does not seem to appear. The reason could be that the length of the feature vector $\mathbf{y}_{\mathcal{I}}$ is constant with respect to the subspace dimension. Also, the SVM is less prone to overfitting than some other methods since it can limit the complexity of the model by the number of support vectors [38]. Fig. 10 shows some examples of the results from the test set. We also performed a 5 times 5-fold cross validation with $H = 10$, which shows that the classification accuracy can reach a mean value of 85.5% with a standard deviation of 2.6%.

According to [23] and [39], the best classification accuracy ever met across various features is 84.1% on the SIMPLIcity data set, which is worse than what we obtained, i.e. 86% on the test set or a mean value 85.5% with the 5 times 5-fold cross validation. Considering difficulty of the classification problem, the performance of our system is promising. It should be emphasized that the good results are obtained solely with raw image patch learning by ASSOM without calculating any descriptors beforehand. It shows that the system can learn patterns directly from the input signal itself.

5) SPMAS Confusion Matrix: The confusion matrix of the SPMAS is shown in Table V. Most of the test images are correctly classified. The best classified category is dinosaur, where all the test images are correctly recognized. The worst classified is the beach category, where only 36 (72%) images are correctly recognized. 4 (8%) images in the beach category are classified as elephant. Examination of the beach category shows that most images in this category have yellow sands and blue sky as backgrounds. Blue sky can often be found in the elephant category while the color of sand is often similar to that of soil in the elephant category. 8 (16%) mountains and glaciers images are classified as beach. This is because mountains can often be found in the beach category and they often share the blue sky as the background. In fact, an object could appear in a range of contexts and a salient point could be located at the border of two different but adjacent objects. When this happens, the context plays an important role. So higher level features, such as global features or semantics-based features, should be helpful in further improving the classification accuracy. Also for some objects, such as sand in beach images and soil in elephant images, colors may not be enough for discriminating them whereas other descriptors such as texture should be more useful.

B. Adult Image Filtering

In the second experiment, we apply the SPMAS to adult image filtering. There are respectively 733 adult and 733 benign training images, 377 adult and 467 benign test images. Each BFL-ASSOM was trained with $200,000$ iterations. The

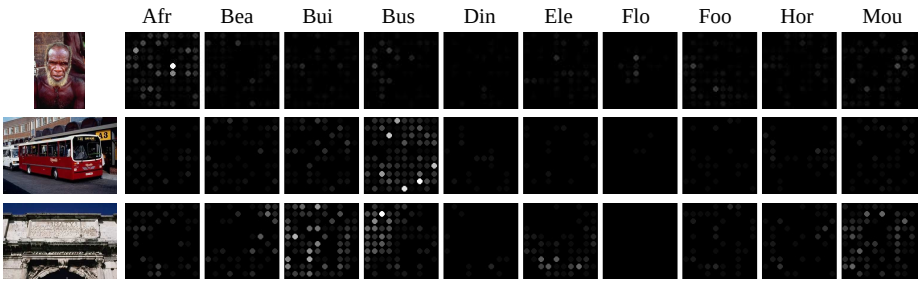

Fig. 9. Three images and their joint cumulative activity maps.

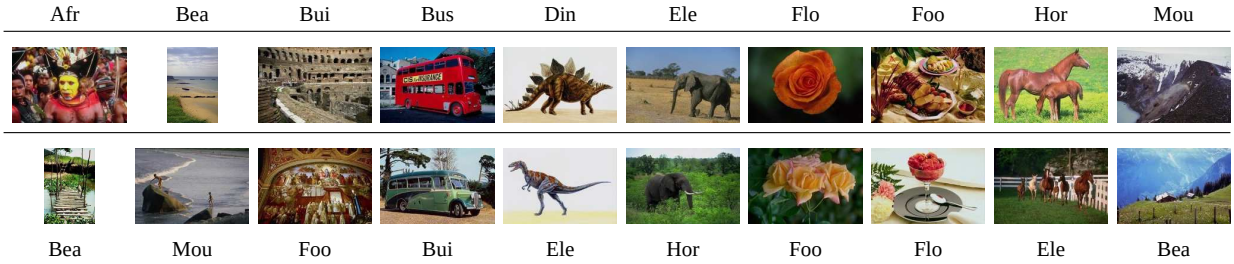

Fig. 10. Examples of the classification results. Horizontal lines are separation marks of different rows. The correct labels are given in the top row. The second row presents examples of correctly classified images. The bottom row shows the incorrectly classified images with the assigned (wrong) labels.

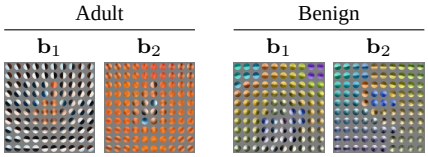

Fig. 11. BFL-ASSOMs generated for adult images and benign images.

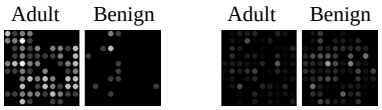

Fig. 12. Joint cumulative activity maps of (a) an adult image and (b) a benign image.

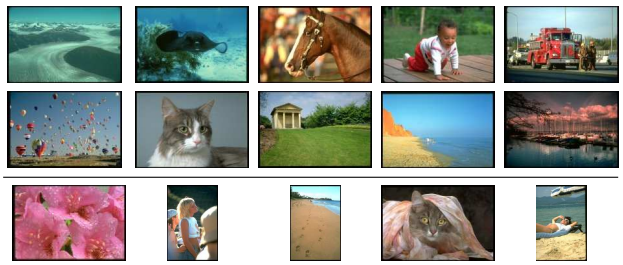

Fig. 13. Classification results of some benign test images. Over the line: Correctly classified examples. Under the line: Incorrectly classified examples.

subspace dimension is 2 and the topology of the ASSOMs is a 10×10 rectangular lattice. The image patches at salient points are circles of 597 RGB pixels. The trained BFL-ASSOMs are shown in Fig. 11, where $\mathbf{b}_1$ of the adult images exhibits evident orientations, while $\mathbf{b}_2$ embodies an obvious yellow tone. In Fig. 12, we show the exemplary joint cumulative activity maps of an adult image and a benign image. For the adult image, the "adult" ASSOM is obviously more activated than the "benign" ASSOM. However, for the benign image, the difference is not that obvious. This is related to the difficulty in learning "the rest of the world", which is practically impossible to be sufficiently sampled. Also, maybe we should wonder what the "benign" ASSOM has really learned from "the rest of the world". It has probably just learned a flat distribution, where nothing is specially interesting.

We use the true positive (TP) rate and the false positive (FP) rate to describe the performance of adult image filtering. The TP rate is the proportion of correctly blocked adult images and the FP rate is the proportion of incorrectly blocked benign images. The SPMAS shows a TP rate of 89.1% with an FP rate of 13.9% on the training set. The F1 measure is 0.878 on the adult class and 0.874 on the benign class. On the test set, the TP rate is 90.2% and the FP rate is 13.1%. The F1 measure is 0.874 on the adult class and 0.892 on the benign class. Fig. 13 shows some classified examples from the benign test subset. The correctly classified images cover a wide range of scenes, including people or objects with skin-like colors. The incorrectly classified images include people with exposed skin or non-human objects with large areas of skin-like colors. To deal with such false alarms, higher-level analysis of the scenes might be necessary, such as detection of humans and context analysis. The receiver operating characteristics (ROC) curve of our system is shown in Fig. 14. The area under the curve (AUC) is a high value 0.958.

We compared the SPMAS with some other adult image filtering systems based on skin detection. Jones and Rehg [40] proposed one of the best skin detectors in the literature with the Compaq database. They built an adult image detection system based on their skin detector. Five features are calculated from

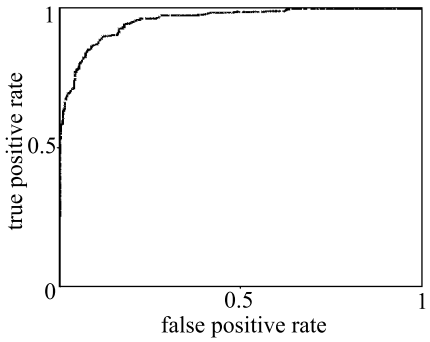

Fig. 14. ROC curve of the SPMAS for adult image filtering.

the skin detection output and two additional features correspond to the dimensions of the input image. A neural network is implemented for classification. The adult image filtering system proposed in [41] is based on a MRF (Markov random fields) skin detector. Nine features are extracted from the skin detection output, including the shape information of skin regions. A multi-layer perceptron (MLP) is then implemented for classification. These systems, as well as the proposed SPMAS, were evaluated on different databases. Despite this, the comparison still makes sense since all the databases are comprehensive and contain thousands of images randomly downloaded from Internet, which are likely to follow similar distributions. Jones and Rehg's system has a TP rate of 88.9% under an FP rate of 13.1%. The system proposed in [41] has a TP rate of about 87.1% under an FP rate of 13.7%. Apparently the SPMAS is competitive to these skin-detection-based adult image filtering systems.

In fact for the skin-detection-based systems, although detected skin provides evidence of adult content, the false "skin" or missed real skin may confuse the adult image detector as well. The SPMAS, however, does not depend on such a skin preprocessor which might be a source of false detection itself. It learns features directly from raw image patches. In this way, the SPMAS also saves considerable manual labor of labeling skin pixels for training of the skin detector.

V. Conclusions and Perspectives

The ASSOM is useful for dimension reduction, invariant-feature generation and visualization. Our study reveals that the increment of each basis vector in the ASSOM learning is a scaling of the component vectors of the input episode, which leads to a fast implementation of the ASSOM, i.e. the FL-ASSOM. With the FL-ASSOM, the computational load of basis updating is linear to the input dimension, which was quadratic with a naive implementation of the traditional ASSOM. We discovered the objective function approximately maximized by the traditional ASSOM and further proposed a batch-mode fast implementation of the ASSOM, i.e. the BFL-ASSOM, where the increment of each basis vector is a linear combination of the component vectors in the input episode. Computational load can be further saved with the BFL-ASSOM. The algorithms previously proposed in the literature could be faster than the traditional implementation of the ASSOM, but not than the fast implementations proposed here. The acceleration of ASSOM learning is especially meaningful for images, which are usually associated with high dimensions.

Experimental results revealed superiority of the SPMAS to the SPSAS in saliency-based invariant feature construction for image classification. In the SPMAS, one ASSOM is trained for each image category to make use of the fact that image labels are known for the training set. The feature vector of the input image is built by combining cumulative activity maps of different ASSOMs. The SPMAS showed promising performance on a 10-category image classification problem and on adult image filtering. Compared to other skin-detection-based adult image filtering systems, one important advantage of the SPMAS is that the manual labor related to preparation of the skin training set can be saved.

There could be several other ways to improve the ASSOM or the subsequent SPMAS. For example, a non-uniformly distributed input signal will lead to ASSOM modules with subspaces too close (when these modules are in the dense zones of the input space) or too far away from each other (when they are in the sparse zones of the input space). The similar problem is often encountered in the SOM. Hierarchical SOMs have been proposed to mitigate this problem. Such map structure permits multi-resolutional representation of the input signal space and fast locating of the winner as demonstrated by Liu *et al*. [42]. A similar strategy could be used for the ASSOM. In this paper, the ASSOMs are only trained on the RGB color feature. We can as well train different ASSOMs on different features, e.g. color and texture, and then combine these ASSOMs. This could be realized in a boosting framework (AdaBoost for example).

Acknowledgment

The authors are grateful to the anonymous reviewers for their valuable comments and suggestions, which have helped to improve the quality of this paper.

References

[1] T. Kohonen, "The Adaptive-Subspace SOM (ASSOM) and its use for the implementation of invariant feature detection," in *Proc. Int. Conf. on Artificial Neural Networks*, F. Fogelman-Soulié and P. Gallinari, Eds., vol. 1, Paris, 1995, pp. 3–10.

[2] T. Kohonen, S. Kaski, and H. Lappalainen, "Self-organized formation of various invariant-feature filters in the Adaptive-Subspace SOM," *Neural Computation*, vol. 9, no. 6, pp. 1321–1344, 1997.

[3] T. Kohonen, *Self-Organizing Maps*, 3rd ed. Springer-Verlag Berlin Heidelberg New York, 2001.

[4] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, vol. 5, no. 6, pp. 927–935, 1992.

[5] C. Wang and W. Wang, "Links between PPCA and subspace methods for complete Gaussian density estimation," *IEEE Trans. Neural Networks*, vol. 17, no. 3, pp. 789–792, 2006.

[6] H. Hase, H. Matsuyama, H. Tokutaka, and S. Kishida, "Speech signal processing using Adaptive Subspace SOM (ASSOM)," The Inst. of Electronics, Information and Communication Engineers, Tottori University, Koyama, Japan, Tech. Rep. NC95-140, 1996.

[7] J. Ruiz-del-Solar, "TEXSOM: Texture segmentation using Self-Organizing Maps," *Neurocomputing*, vol. 21, no. 1-3, pp. 7–18, 1998.

[8] D. de Ridder, O. Lemmers, R. P. W. Duin, and J. Kittler, "The Adaptive Subspace Map for image description and image database retrieval," in *Proc. Joint IAPR Int. Workshops SSPR and SPR*, 2000, pp. 94–103.

[9] B. Zhang, M. Fu, H. Yan, and M. Jabri, "Handwritten digit recognition by Adaptive-Subspace Self-Organizing Map (ASSOM)," *IEEE Trans. Neural Networks*, vol. 10, no. 4, pp. 939–945, 1999.

[10] E. López-Rubio, J. Muñoz-Pérez, and J. A. Gómez-Ruiz, "A Principal Components Analysis Self-Organizing Map," *Neural Networks*, vol. 17, no. 2, pp. 261–270, 2004.

[11] E. López-Rubio, J. Muñoz-Pérez, J. A. Gómez-Ruiz, and E. Domínguez-Merino, "New learning rules for the ASSOM network," *Neural Comput & Applic*, vol. 12, no. 2, pp. 109–118, 2003.

[12] S. Mcglinchey and C. Fyfe, "Fast formation of invariant feature maps," in *European Signal Processing Conference*, Island of Rhodes, Greece, 1998.

[13] E. Oja, "Neural networks, principal components, and subspaces," *Int. J. of Neural Systems*, vol. 1, no. 1, pp. 61–68, 1989.

[14] I. Gordon, *Theories of Visual Perception*, 2nd ed. Wiley, 1997.

[15] D. Marr, *Vision*. W.H. Freeman and Company, 1982.

[16] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 5, pp. 530–535, 1997.

[17] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. on Computer Vision*, Corfu, Greece, 1999, pp. 1150–1157.

[18] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. of Computer Vision*, vol. 37, no. 2, pp. 151–172, 2000.

[19] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 12, pp. 1349–1380, 2000.

[20] N. Sebe and M. S. Lew, "Salient points for content-based retrieval," in *Proc. British Machine Vision Conf.*, Manchester, UK, 2001, pp. 401–410.

[21] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Madison, USA, 2003, pp. 257–263.

[22] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 8, pp. 1026–1038, 2002.

[23] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-sensitive integrated matching for picture libraries," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 9, pp. 947–963, 2001.

[24] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.

[25] T. Gevers and A. W. M. Smeulders, "Pictoseek: combining color and shape invariant features for image retrieval," *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 102–119, 2000.

[26] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.*, Manchester, UK, 1988, pp. 147–151.

[27] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. 7th European Conf. on Computer Vision*, Copenhagen, Denmark, 2002, pp. 128–142.

[28] C. Laurent, N. Laurent, M. Maurizot, and T. Dorval, "In-depth analysis and evaluation of saliency-based color image indexing methods using wavelet salient features," *Multimedia Tools and Applications*, vol. 31, no. 1, pp. 73–94, 2006.

[29] J.-M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts, "Color invariance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 12, pp. 1338–1350, 2001.

[30] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[31] G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual categorization with bags of keypoints," in *Proc. 8th European Conf. on Computer Vision*, Prague, Czech Republic, May 2004, pp. 327–334.

[32] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[33] E. Oja, *Subspace Methods of Pattern Recognition*. Letchworth, UK: Research Studies Press, 1983.

[34] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[35] J. Lu, X. Yuan, and T. Yahagi, "A method of face recognition based on fuzzy c-means clustering and associated sub-NNs," *IEEE Trans. Neural Networks*, vol. 18, no. 1, pp. 150–160, 2007.

[36] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. Morgan Kaufmann, 2005.

[37] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, "Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft kNN ensemble," *IEEE Trans. Neural Networks*, vol. 16, no. 4, pp. 875–886, 2005.

[38] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley&Sons, 2001.

[39] T. Deselaers, D. Keysers, and H. Ney, "Features for image retrieval: a quantitative comparison," in *DAGM'04: 26th Pattern Recognition Symposium*, ser. Lecture Notes in Computer Science, vol. 3175, Tübingen, Germany, 2004, pp. 228–236.

[40] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *Int. J. of Computer Vision*, vol. 46, no. 1, pp. 81–96, 2002.

[41] H. Zheng, M. Daoudi, and B. Jedynak, "Blocking adult images based on statistical skin detection," *Electronic Letters on Computer Vision and Image Analysis*, vol. 4, no. 2, pp. 1–14, 2004.

[42] D. Liu, X. Xiong, B. DasGupta, and H. Zhang, "Motif discoveries in unaligned molecular sequences using self-organizing neural networks," *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 919–928, 2006.

Huicheng Zheng (M'07) was born in 1974. He received the B.Sc. (Eng.) degree in electronics and information systems and the M.Sc. (Eng.) degree in communications and information systems from Sun Yat-sen University, Guangzhou, China, in 1996 and 1999, respectively, and the Ph.D. degree in computer science from University of Lille 1, France, in 2004.

From 2005 to 2006, he was a Research Fellow with European Research Consortium for Informatics and Mathematics, during which period the host institutions where he stayed were France Telecom R&D, Rennes, France, and Trinity College Dublin, Dublin, Ireland. He served as a Visiting Scholar at Center for Imaging Science, The Johns Hopkins University, Baltimore, in 2004. Since 2007, he has been a Lecturer in Department of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou, China. His current research interests include computer vision, neural networks, machine learning, and other related areas.

Grégoire Lefebvre was born in 1977. He received the M.E. degree in applied mathematics from National Institute of Applied Sciences (INSA), Rouen, France, in 2000, and the Ph.D. degree in cognitive science from University of Bordeaux, France, in 2007.

Since 2007, he has been working as a Researcher at France Telecom R&D - Orange Labs, Rennes, France, where his research interests include multimedia indexing, face detection and recognition, and neural networks.

Christophe Laurent was born in 1971. He received the Ph.D. degree in 1998 from the University of Bordeaux (France), where he worked on parallel processing in the field of computer vision.

From 1998 to 2001, he was a Research Engineer in Thomson Multimedia R&D, where he worked on security of information technology. During this period, he led several projects related to e-commerce, network security and protection of multimedia contents. In 2001, he joined France Telecom R&D, where his research interests were image indexing and particularly salient point-based image representation, string-based representation, image classification and face processing. From 2005, he has been working as a Technical Architect at the billing IT division of France Telecom, where he is in charge of the evolution of the settlement technical architecture to embed new technologies. He has published more than 30 papers and is the author of 20 patents.