Neurocomputing 150 (2015) 106–115

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Adaptive feature descriptor selection based on a multi-table reinforcement learning strategy

Monica Piñol [a,*], Angel D. Sappa [a], Ricardo Toledo [a,b]

[a] Computer Vision Center, Universitat Autònoma de Barcelona, 08193-Bellaterra, Barcelona, Spain
[b] Computer Science Dept., Universitat Autònoma de Barcelona, 08193-Bellaterra, Barcelona, Spain

ARTICLE INFO

Article history:
Received 21 October 2013
Received in revised form
28 January 2014
Accepted 22 March 2014
Available online 14 October 2014

Keywords:
Reinforcement learning
Q-learning
Bag of features
Descriptors

ABSTRACT

This paper presents and evaluates a framework to improve the performance of visual object classification methods, which are based on the usage of image feature descriptors as inputs. The goal of the proposed framework is to learn the best descriptor for each image in a given database. This goal is reached by means of a reinforcement learning process using the minimum information. The visual classification system used to demonstrate the proposed framework is based on a bag of features scheme, and the reinforcement learning technique is implemented through the Q-learning approach. The behavior of the reinforcement learning with different state definitions is evaluated. Additionally, a method that combines all these states is formulated in order to select the optimal state. Finally, the chosen actions are obtained from the best set of image descriptors in the literature: PHOW, SIFT, C-SIFT, SURF and Spin. Experimental results using two public databases (ETH and COIL) are provided showing both the validity of the proposed approach and comparisons with state of the art. In all the cases the best results are obtained with the proposed approach.

1. Introduction

In the computer vision domain the visual object classification (VOC) has attracted the attention of researchers over the last two decades (e.g., [1–4]). Generally, VOC is based on the representation of the given scene in a space of features, which were extracted and then described by means of some feature descriptors. These feature descriptions are then used as discriminative elements to characterize the given objects. They are computed using information of *interest points* together with their neighborhood; such interest points are pixels with special characteristics (e.g., [5,6]). Hence, given an image, the feature descriptors characterize the objects at a higher abstraction level, where classical learning techniques can be used in order to recognize the target object. More elaborated techniques, such as Bag of Features (BoF), are becoming nowadays popular for visual object recognition (e.g., [7,3,8,9]). The BoF consists of four steps as detailed below:

1. Extract the features from the images of the training set using a given detector and a given descriptor.
2. Build a dictionary of visual words using the features extracted before.
3. Construct a histogram, using (1) and (2), for each image in the training set. Hence, the histogram bins represent the number of times a visual word is in the image.
4. Train a classification algorithm using the histogram obtained before.

The BoF architecture is flexible, so that there are different combinations that can be used to implement the four steps presented above. The final performance of the BoF depends on the correct algorithm selection.

The current work is focused on the first step of the BoF; in particular, the goal is to learn the best algorithm to describe the interest points. From our experience, the performance of the BOF is strongly influenced by the image feature descriptor, so we state that identifying the best image descriptor for each image will improve the classification rate. A naive approach to solve this problem could be the concatenation of all the possible descriptors. However, this solution is not always feasible since on the one hand it could take a large amount of resources (e.g., memory, CPU time) and on the other hand this would introduce noise to the solution [10]. The challenge of the problem and the importance of finding the right solution have been recently addressed. An approach to select the best descriptor for each image is presented in [10,11].

* Corresponding author.
E-mail addresses: mpinyol@cvc.uab.es (M. Piñol), asappa@cvc.uab.es (A.D. Sappa), ricard@cvc.uab.es (R. Toledo).

http://dx.doi.org/10.1016/j.neucom.2014.03.080
0925-2312/

In [10] a method for selecting the best descriptor for every image in the database is proposed. In order to select the best descriptor, several attributes of the image (e.g., colorfulness, roughness, shininess, etc.) are taken into account. Although interesting results are presented, their main drawback is the use of a supervised learning scheme where the authors select the descriptors with a subjective criterion. On the contrary, in [11] a method that learns the best descriptor for each image using a Reinforcement Learning (RL) scheme is presented. The RL is a simple learning method based on a trial and error strategy. This work presents two improvements from [11].

1. We propose to use several state definitions.
2. A multi-table scheme is introduced in order to exploit the best state definition for each image.

In summary, this work proposes a novel method to learn the best descriptor from a given set. In order to improve the performance, multiple state definitions are used. This scheme works with a BoF approach, and in concrete, the implementation uses a kd-tree in the second step and a support vector machine (SVM) in the fourth step. The reminder of the paper is organized as follows. Section 2 presents the state of the art. Section 3 summarizes the RL technique. Then, Section 4 presents in detail the proposed method. Experimental results and comparisons are provided in section 5. Section 6 gives the conclusions and future work.

2. State of the art

Reinforcement learning is a learning technique widely used in the robotics community; recently, some work involving RL have been proposed in the computer vision field. For instance, in image segmentation, the RL technique is used to select the appropriate threshold (e.g., [12,13]). In [14] the authors propose a RL based approach to tackle the face recognition problem. The authors present a method to learn the set of dominant features for each image. An approach that joins an active learning with RL is presented in [15]; in this case it learns the exploration and exploitation trade-off during the sampling process. Finally, there are also some works in visual object recognition using RL (e.g., [16–19]). In [20], the authors propose a bottom-up/top-down recognition system in order to learn a similar model than the human learning. In concrete, the method joins the RL and a first order logic technique to recognize objects. In [17] a method that learns the features of the image in order to recognize objects is presented . A RL based approach is used for selecting the classification algorithm, which is later on embedded in the fourth step of a BoF scheme [19].

In this work, we propose the use of RL to learn the best feature descriptor for VOC with a BoFs approach. The novelty of the work, with respect to the former one [11], is the multi-table framework that enables the use of different states

3. Reinforcement learning

The reinforcement learning, as mentioned before, is a trial-and-error learning process [21] where the agent does not have a prior knowledge about which is the correct action to take. RL can be used as a technique to solve a Markov decision process (MDP) problem, in which the *agent* learns how to take an *action* in a given *environment* in order to maximize the expected *reward*. These concepts are incorporated to the tuple of MDP $\langle S, A, \delta, \tau\rangle$ where:

- S is a set of environment states. In this work the states are characteristics of the image.
- A is a set of actions. This work uses five image feature descriptors as actions: Spin, SIFT, SURF, C-SIFT and PHOW.δ
- is a transition function, $\delta : S \times A \rightarrow S$. τ
- is a reward/punishment function, $\tau : S \times A \rightarrow \Re$.

Using the definitions presented above, the RL method works as follows: the agent interacts with the environment to select an action (a_h). The action is selected to maximize the expected reward (r_t) based on $\tau(s_z, a_h)$. Applying the action (a_h) to the state (s_z), the environment gives a new state (s_{z+1}) according to the δ function, and a reward/punishment (r_t) according to the τ function. An illustration summarizing this process is depicted in Fig. 1.

The RL can be implemented using dynamic programming, Monte Carlo method, and temporal difference learning. In this work, a temporal difference based method has been used because it does not require a model and it is fully incremental [21]. More specifically, we decided to use the Q-learning algorithm [22]. In Q-learning, the agent learns the action policy $\pi : S \longrightarrow A$, where the policy π maps the states and the actions to maximize the expected reward (formulated as $E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots]$). The value function ($V^\pi$) over the states is generated using the policy π and the states (Eq. (1)) and π^* is the optimal policy that the agent must learn during the training process (see Eq. (2)):

$$V^\pi(s) \equiv E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots] \equiv E\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i}\right], \tag{1}$$

$$\pi^* \equiv \arg\max_\pi V^\pi(s), \quad \forall(s). \tag{2}$$

The best value function V^{π^*} is learnt with the RL strategy using the δ and τ functions explained before. However, since none of them (δ and τ) are known a priori the Q-learning proposes an algorithm to also learn the optimal policy (π^*). Specifically, the Q-learning defines an evaluation function (Q) as Eq. (3) that is used to find the best action. In this case, the π^* can be defined as in the following equation:

$$Q(s,a) = E[r(s,a) + \gamma V^{\pi^*}\delta(s,a)], \tag{3}$$

$$\pi^*(s) = \text{argmax}_a E[r(s,a) + \gamma V^{\pi^*}\delta(s,a)] = \text{argmax}_a Q(s,a). \tag{4}$$

The δ and τ can be deterministic or nondeterministic. δ is deterministic when, given a state (s_z) and applying an action (a_h) a new state (s_{z+1}) is returned; if we cannot guarantee to reach always the same new state (s_{z+1}) the δ function is nondeterministic. The same happens with the τ function, for a given state (s_z) and action (a_h) it could return different rewards (r_t). When the δ and/or τ functions are nondeterministic the environment is nondeterministic. In the case of nondeterministic Q-learning, the formulation of the Q evaluation is defined as follows:

$$Q_n(s_z, a_h) \longleftarrow (1-\alpha_n)Q_{n-1}(s_z, a_h) + \alpha_n[r + \gamma \max_{a'} Q_{n-1}(s_{z+1}, a')], \tag{5}$$

$$\alpha_n = \frac{1}{1 + \textbf{visits}_n(s_z, a_h)}, \tag{6}$$

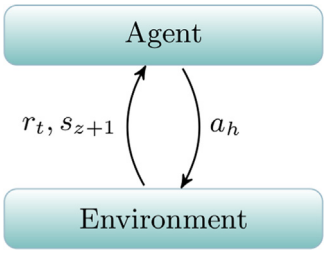

Fig. 1. Illustration of interaction between agent and the environment.

where $0 \leq \gamma < 1$ is a discount factor for future reinforcements. Eq. (6) is the value α_n for a nondeterministic environment and *visits* is the total number of times the pair (s_z, a_h) has been visited including the nth iteration.

The environment can be deterministic or nondeterministic, but, in both cases, the convergence is achieved. In a deterministic environment, the convergence is achieved when:

$$|Q_{n+1}(s,a) - Q(s,a)| = \gamma \Delta_n, \tag{7}$$

is true. On the contrary, in the nondeterministic environment does not exist an accepted formula to the convergence. The convergence for a nondeterministic environment is assumed when a user defined number of iterations is reached. The theorem "Convergence of Q-learning for nondeterministic Markov decision processes" [23] shows that a nondeterministic MDP converges when there is a bounded reward $(\forall(s,a), |r(s,a)| \leq c$ and $0 < \alpha_n \leq 1)$. Let $n(i,s,a)$ be the number of times the learning process has been applied, where i represents the iterations in the tuple (s_z, a_h). Eq. (8) is true in the ith iteration when $n \longrightarrow \infty$ with probability 1:

$$\sum_i^{\infty} \alpha_n(i,s,a) = \infty, \quad \sum_i^{\infty} [\alpha_n(i,s,a)]^2 \leq \infty. \tag{8}$$

In a nondeterministic environment, the convergence process is quite time consuming; for example, in [23] we can see that the Tesauro's TD-GAMMON needs for training 1.5 million of backgammon games iterations and each of them contains tens of state-action transitions.

4. Proposed method

This paper proposes a method to learn the best descriptor for each image. Fig. 2 shows an illustration of the proposed scheme. In particular, Fig. 2(*left*) presents a classical BoF (i.e., [7,3,8,9]) while Fig. 2(*right*) shows the proposed RL based scheme. In fact, we propose a new multi-table RL based strategy to select the best descriptor for each image from a set that contains the most widely used according to the literature (i.e., Spin, SIFT, SURF, C-SIFT and PHOW). This section is organized as follows. First, Section 4.1 introduces the elements of the tuple. Next, the architecture used to learn the best descriptor for each image is explained in Section 4.2. Finally, the combination of Q-tables is given in Section 4.3.

It should be noticed that the proposed approach does not follow classical RL schemes, where an action (a_h) is applied and results in a reward/punishment (r_t) and a new state (s_{z+1}). In the proposed method, after applying the action only a reward is returned (does not result in a new state); hence after applying the action and updating the Q-table, a new image from the database is considered. In this work, the state s_z is only affected by the previous visits but not by the future ones since the MDP is of first order [14]. Then, in this case, the Eq. (5) is re-written as follows:

$$Q_n(s_z, a_h) \longleftarrow (1-\alpha_n) Q_{n-1}(s_z, a_h) + \alpha_n [r + \gamma \max_{a'} Q_{n-1}(s_z, a')]. \tag{9}$$

4.1. Tuple definition

The agent of RL learns the action that maximizes the reward in each state. In order to do this, the RL uses the $\langle S, A, \delta, \tau \rangle$ tuple. This section describes the elements of the MDP tuple.

4.1.1. State definition

In this work a state is a representation of the image. There are several characteristics of the images that can be used as states and each state definition achieves different results. This section explains the four state definitions used to deal with the visual object classification. The first three state definitions (see $L^*a^*b^*$ *based state definition*, *Entropy based state definition* and *Gradient based state definition*) use the scheme proposed in Fig. 3 that consists of a vector with 13 elements. The first image (Fig. 3(a)) contributes with one value, the second image (Fig. 3(b)) with 4 and the last image (Fig. 3(c)) with 8 elements, resulting in the vector with 13 elements. Then, the last state definition (see Section 4.1.1) consists of a vector of 50 elements obtained from a representation of visual words.

The steps for computing states are:

1. For each image in the training set, extract the vector of characteristics.
2. Using the K-means algorithm build K clusters. The center of the clusters are the states $(S = \{s_z\}_{0 < z < K})$

$L^*a^*b^*$ *based state definition*: This state definition uses the $L^*a^*b^*$ color space. This color space is obtained by converting RGB to XYZ and then, XYZ to $L^*a^*b^*$ (see [25,26] for more details). Fig. 4(b) shows a picture in $L^*a^*b^*$ color space while Fig. 4(a) depicts the original image in RGB color space. The L^* represents the luminance, a^* represents the difference between the red and green colors, and b^* is the difference between the yellow and blue. This state definition results in a vector of characteristics of 39 elements. This vector is computed using the scheme presented in Fig. 3, which corresponds

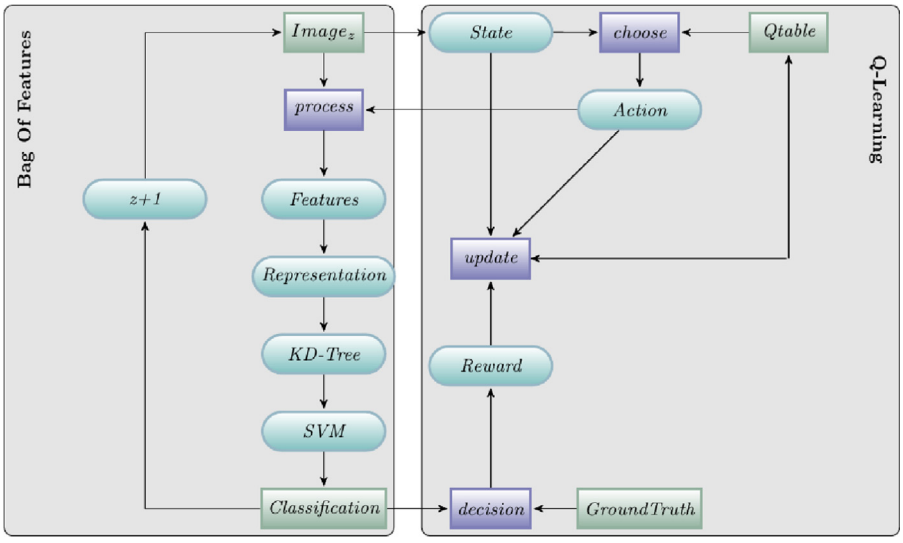

Fig. 2. Illustration of learning the best descriptor for each image using Q-learning.

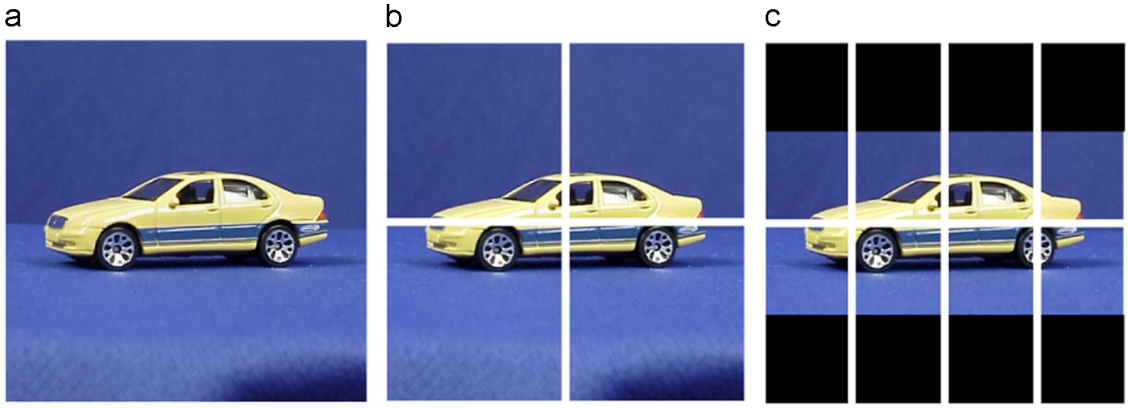

Fig. 3. (a) Image from ETH database [24] (all the images from the database are resized to 128×128). (b) Image split up into four squares. (c) Image split up into sixteen squares, note that only eight of them are used.

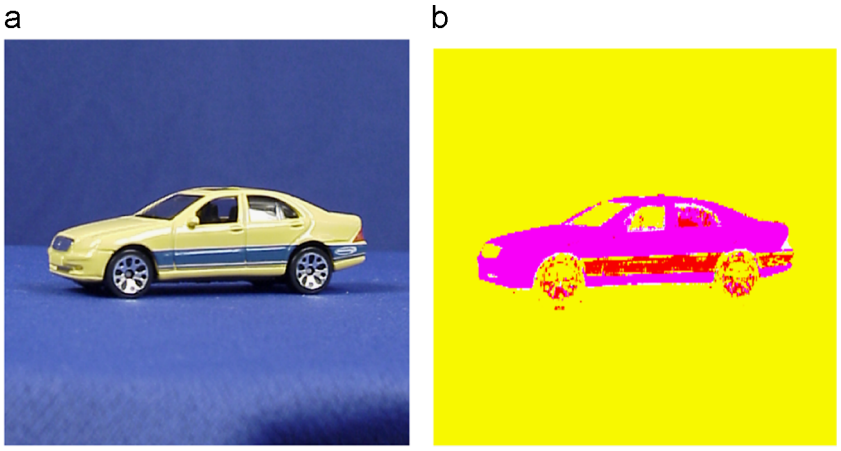

Fig. 4. (a) Image from ETH database and (b) conversion of RGB image to $L^*a^*b^*$ color space. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

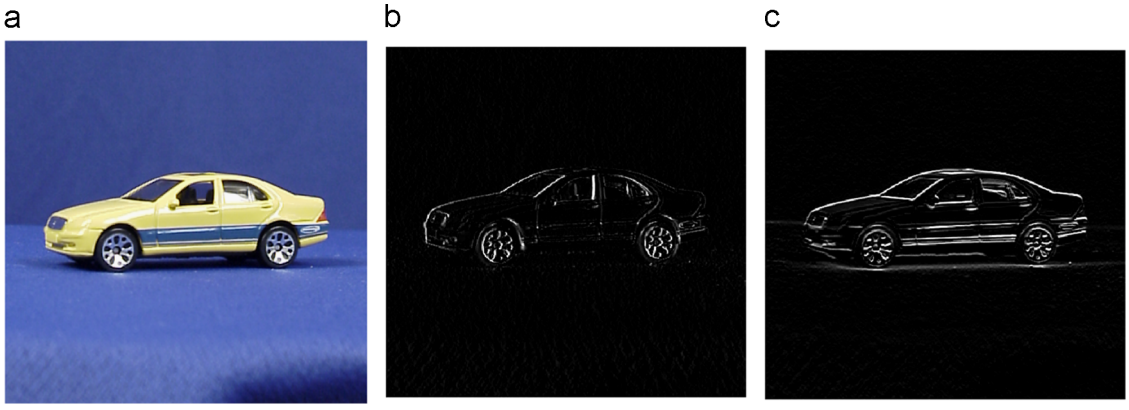

Fig. 5. (a) Image from ETH database, (b) image gradient in x direction, and (c) image gradient in y direction.

to 13 squares for every given image, actually for each square this state computes three median values, one for each channel of $L^*a^*b^*$.

Entropy based state definition: This state definition is based on the uncertainty of the information in the image. In this case, the vector of characteristics also has 13 elements (Fig. 3). This state definition uses the gray level representation and, for each element of the partition image, extracts the entropy as follows:

$$E = -\sum_{i=1}^{N} p_i \log_2 (p_i), \tag{10}$$

where p_i is the histogram of image and N is the number of pixels in the given region.

Gradient based state definition: This state definition uses the gray level representation. The vector of characteristics has 26 elements (13 elements per each direction). For each element of the partition image computes the gradient in x and y direction. Then, the state vector is built with the median of gradient values contained in each image partition (see Fig. 3). Fig. 5(b) shows the edges extracted using the gradient in x direction and Fig. 5(c) using the y direction.

Histogram of interest point based state definition: This state definition uses all the descriptors of the set of actions to compute the vector. This state definition extracts a vector of characteristics containing 50 elements. The elements of this vector are obtained according to the five actions defined in the next section. For each action (descriptor), a dictionary of visual words is built using the first 2 steps of the classical BoF. First, the features are extracted and then, using these features a dictionary of ten visual words is built. The vector results from the process of concatenating the representations of the image using a histogram of ten visual words for each descriptor.

4.1.2. Action definition

In this work, the actions came from a set of descriptors that the RL technique learns during the training process; as a result the best descriptor for each image is found. A large number of descriptors have been proposed in the literature in recent years [27]. In this work five frequently used descriptors were selected; they are detailed below:

- SIFT: Scale-Invariant Features Transform is a descriptor that finds interest points and describes them. The points are found by using space scale of Gaussian filters and are described by means of the gradients around it [2].
- PHOW: Pyramid Histogram Of visual Words is a descriptor based on SIFT. This descriptor does not find interest points; on the contrary, it uses a dense grid at different scales [9].
- C-SIFT: Color-SIFT is a descriptor based on SIFT that incorporates the color of the image [28].
- SURF: Speed Up Robust Feature is a descriptor based on the sum of 2D Haar wavelet. This descriptor is efficiently computed using integral images [8].
- Spin: This descriptor makes a histogram of the intensity values of the pixels [29].

4.1.3. δ function

The classical RL involves the δ function as follows:

$$\delta : S \times A \rightarrow S \tag{11}$$

where, given a state (s_z) and applying an action (a_h) results in a new state (s_{z+1}). In this framework, given an image (I_{s_z}) and applying an action (a_h), the δ function does not return a new state, instead, after applying an action to an image the output is a new image description ($I_{s_{z'}}$). The new image description is obtained using the descriptors mentioned above. The features described are intrinsic for each image, hence, two images with the same state and applying the same action could result in different image descriptions (see Fig. 6).

As mentioned before, the δ function is deterministic when, given a state (s_z) and applying an action (a_h) always returns the same new state (s_{z+1}). Even knowing that in this work δ behaves different from the classical it is still non-deterministic.

4.1.4. τ function

In this work the τ function is also a nondeterministic function. During the process we cannot ensure that two images at the same state (s_z) and doing the same action (a_h) lead to the same $\tau(s_z, a_h)$ [23]. For instance, in Fig. 6(a)–(c) applying the action SIFT over the three images we obtain the same reward. In all of them, the classification matches with the ground truth. However, if we change the action the rewards could be different; for instance, if SURF is applied to the same set of images, Fig. 6(a) differs from the ground truth and returns 'tomato'. Fig. 6(b) matches with the

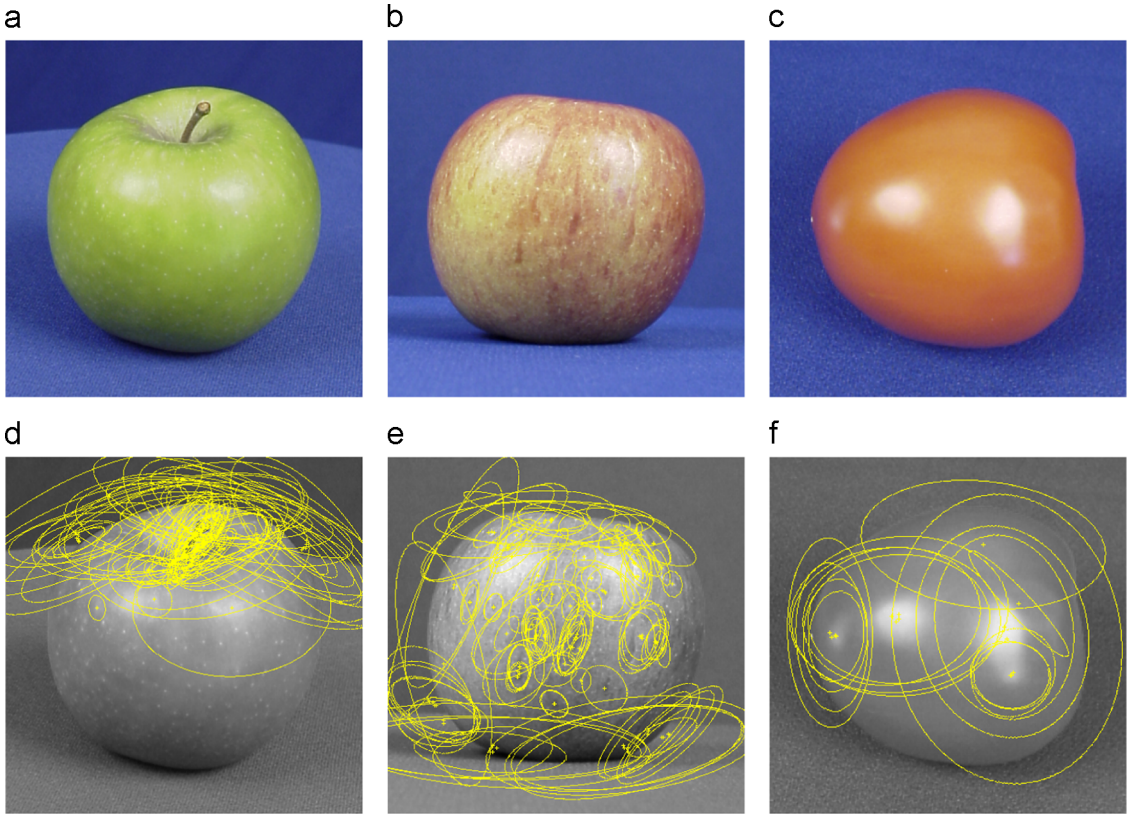

Fig. 6. (*top*) Illustration of three images from the ETH database, assume that all of them are in the same state. (*bottom*) The image patches obtained after applying the same action: SIFT.

class, and finally, Fig. 6(c) does not match with the ground truth; in this case the process of classification returns 'apple'. Therefore, applying SURF, to the second image (Fig. 6(b)) gives a reward, but applying it in to the other two images (Fig. 6(a) and (c)) results in a punishment.

4.2. Training to learn the best descriptor

This section presents the architecture proposed for selecting the best descriptor. Our VOC scheme aims at using the minimum information and improving the classification rate. The proposed architecture learns the best descriptor for each image. The proposed method is based on BoF and, in concrete, this method is focussed on the first step of the BoF.

In order to train and test our approach, the database is split up into three sets: BoF training set (BoFTS), Q-table training set (QTTS) and testing set. The learning process has two steps: The first training step is performed using the BoFTS where, for each descriptor from the set of actions a kd-tree (T_{a_h}) is built and a SVM is used to classify the objects [30].

The second training step is depicted in Fig. 2. First of all, the Q-table is initialized with a "0" in all the cells and then the process starts. Given an image from the QTTS extracts the characteristics to find the state (s_z). The agent extracts the action (a_h) using the

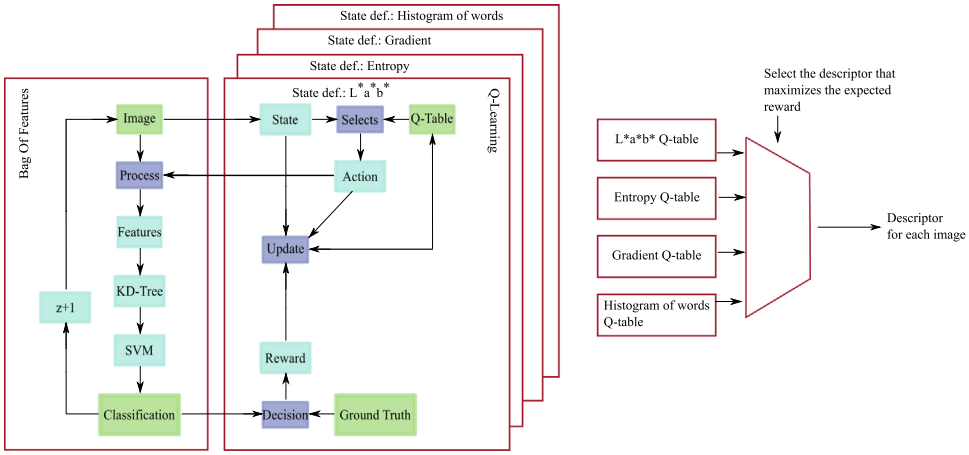

Fig. 7. Multi-table combination.

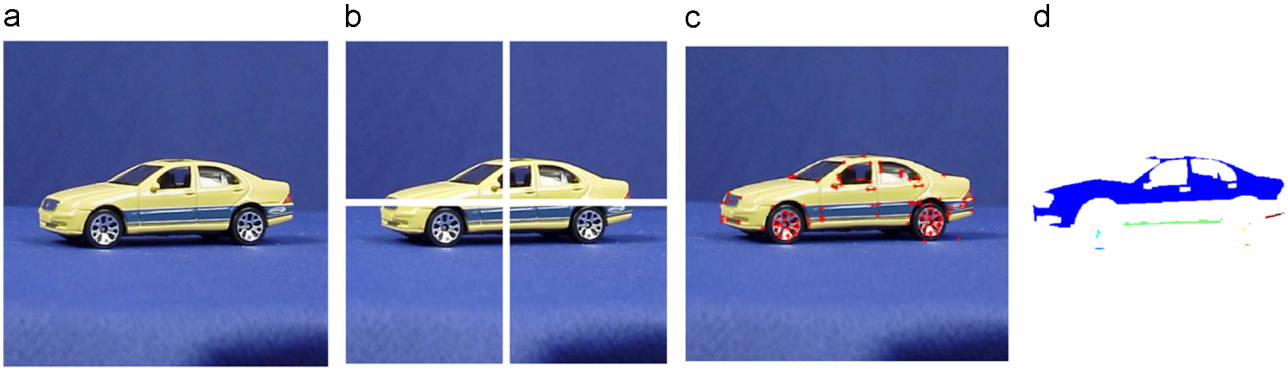

Fig. 8. (a) Image from ETH database, (b) image split up into four equally sized squared blocks, (c) corners detected in the image and (d) blobs detected in the image.

Fig. 9. Some of the objects contained in the nine classes of ETH database.

exploration/exploitation trade-off. It applies the action (a_h) to the image (s_z) and the system obtains the features of this image using the selected descriptor. Then, these features are used to compute the histogram of this image according to the tree trained above (T_{a_h}, a_h). Finally, a SVM is used to classify the histogram by returning a label (the classification of the image). The agent compares the label of the image with the ground truth and obtains a reward/punishment (r_t). Finally, the Q-table is updated using Eq. (9) and the process starts again with a new image from the QTTS.

4.3. Combination of Q-tables

After explaining the learning process in Section 4.2, it should be noticed that in Section 4.1.1 four different states definition have been proposed; hence, the next question is how to select the right state definition. A *brute-force* idea could be to concatenate all the state definitions; however, as will be shown in Section 5 this strategy does not always reach the best performance.

This section proposes a method to learn the best descriptor from only one state definition for each image (e.g., states *L*a*b* based state definition*, *Entropy based state definition*, *Gradient based state definition* and *Histogram of interest point based state definition*). Therefore, the process proposed above is repeated four times (one for each state definition) and it finishes with a Q-table for each state definition. Now the objective is to decide the action for each image using the information from the Q-tables. In this work, we propose a simple voting strategy for combining the four Q-tables. The strategy consists in selecting the best action proposed by the Q-tables for each image. The best action from the Q-tables is the action that maximizes the reward (Fig. 7 illustrates this multi-table scheme).

5. Experimental results

The proposed method has been evaluated using two different databases (ETH and COIL). The evaluation framework compares the results using:

- A unique descriptor for the whole database.
- All the descriptors concatenated in a single one.
- The RL-based approach presented in [11].
- The RL-based approach with different state definitions.
- All the states concatenated.
- The information provided by the Q-tables combined (Fig. 7).

These experiments have been performed with the first database (ETH database) and then repeated with the second one to validate them.

In [11] a single Q-table is considered and a state definition using image content statistics is proposed. This state definition is based on the use of the gray scale image information together with additional image detectors as presented below. The resulting state (vector) contains 17 elements obtained from the following four groups.

1. Mean, standard deviation and the median gray values for each image (Fig. 8(a)) (this contributes with three elements to the vector).
2. Mean, standard deviation and the median gray values for each of the four equally sized squared blocks (as depicted in Fig. 8(b)) (resulting in 12 elements in the vector).
3. The number of corners (Fig. 8(c)) obtained from Harris corner detector [5] (it contributes just with 1 element to the vector).
4. The number of blobs (Fig. 8(d)) obtained converting gray scale to black and white using OTSU threshold [31], and then, a labeling algorithm (*bwlabel*) with a connectivity of 8 neighbors [32] (this is the last element in the vector).

The first experiments were performed using the ETH database. Fig. 9 shows the nine classes of the database (i.e., apple, car, cow, cow-cup, cup, dog, horse, pear and tomato). In order to do the experiments, we randomly select 45 images from each class. As mentioned above, the database was split up into three sets: 15

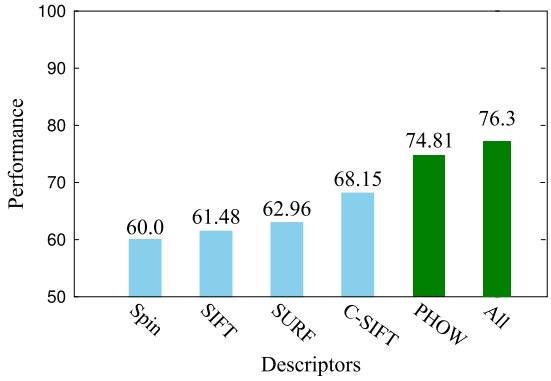

Fig. 10. Classification performance of a single descriptor and a combination of all descriptors for ETH database.

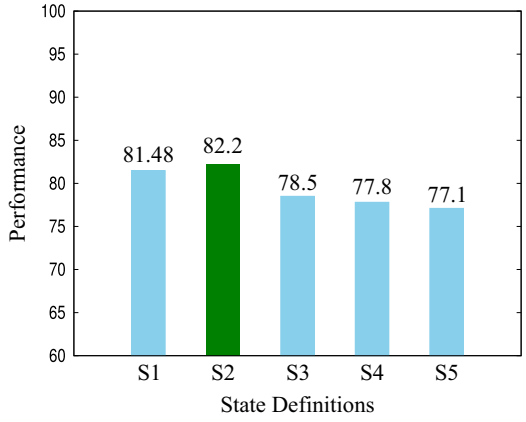

Fig. 11. Classification performance using BoF with RL and different state definitions from the ETH database. S1: state definition from [11]. S2: *L*a*b**. S3: Entropy. S4: Gradient. S5: Histogram of interest point.

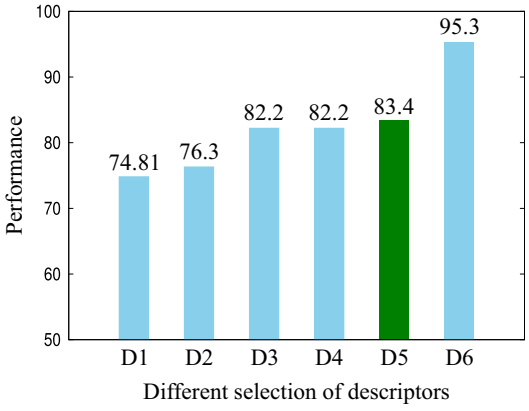

Fig. 12. Comparison of different methods to select the descriptors from the images of the ETH database. D1: PHOW descriptor. D2: All descriptors concatenated in a single vector. D3: The best descriptor selected by RL with *L*a*b** as state. D4: The best descriptor selected by RL with a concatenation of states: S2, S3, S4 and S5. D5: The proposed multi-table approach. D6: Oracle.

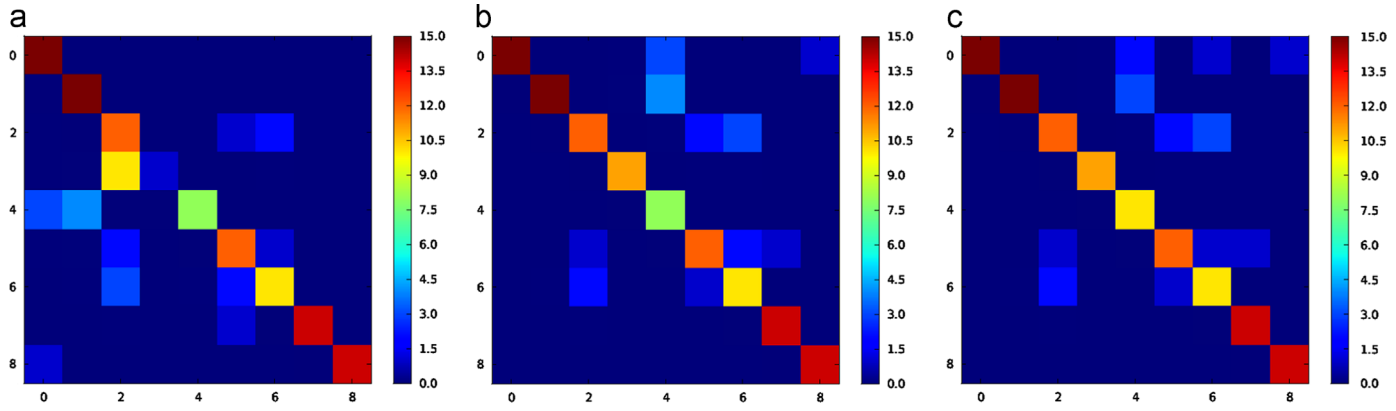

Fig. 13. Different confusion matrices for ETH database: (a) Using only the PHOW descriptor (74.81%), (b) using $L^*a^*b^*$ state definition (82.2%) and (c) using the proposed approach (83.7%).

Fig. 14. Illustration from COIL database, one object per class is depicted.

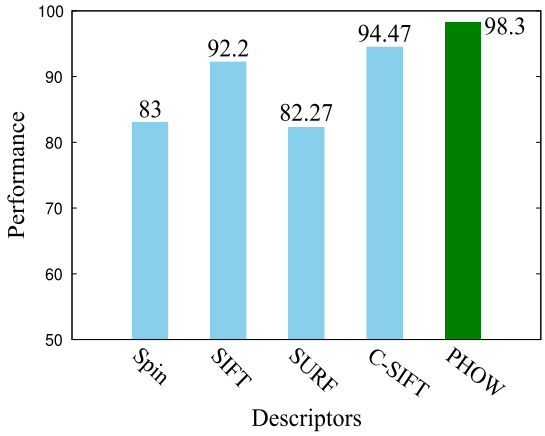

Fig. 15. Classification performance of BoF using a single descriptor for COIL database.

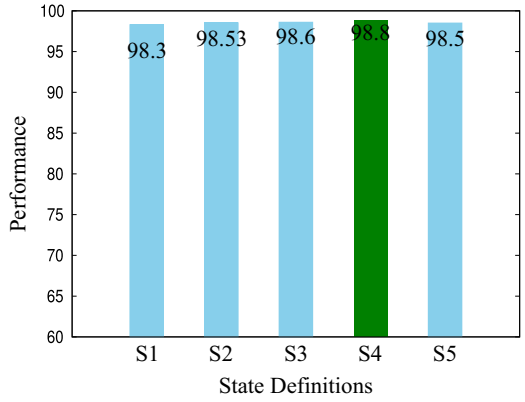

Fig. 16. Classification performance using BoF with RL and different state definitions for COIL database. S1: State definition from [11]. S2: $L^*a^*b^*$. S3: Entropy. S4: Gradient. S5: Histogram of interest point.

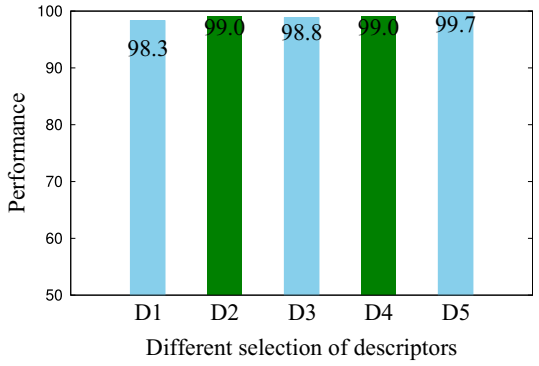

Fig. 17. Comparison of different methods to select the descriptors from the COIL database. D1: PHOW descriptor. D2: All descriptors concatenated in a single vector. D3: The best descriptor selected by RL with Gradient as state. D4: Multi-table approach. D5: Oracle.

images for training the BoF, 15 images for training the Q-table and finally, and 15 images for testing. The value of γ is 0.9 and the process uses a ε-greedy strategy of exploration–exploitation trade-off ($\varepsilon = 0.2$).

In order to compare the results obtained using the proposed approach, first of all, the performance for each descriptor and the combination of all them are computed (see Fig. 10). It can be appreciated that the best single descriptor is PHOW with a performance of 74.81% of correctly recognized objects and, using the combination of all the descriptors, this ratio is increased up to 76.3% of classification. In this work, the *oracle* means the best performance that can be reached only if each image were described by its best descriptor. In this particular database, the oracle improves the results in 19% reaching 95.6% of performance.

Fig. 11 shows the performance using the proposed RL based scheme with the different state definitions. The best performance is obtained using the $L^*a^*b^*$ state reaching 82.2% of classification

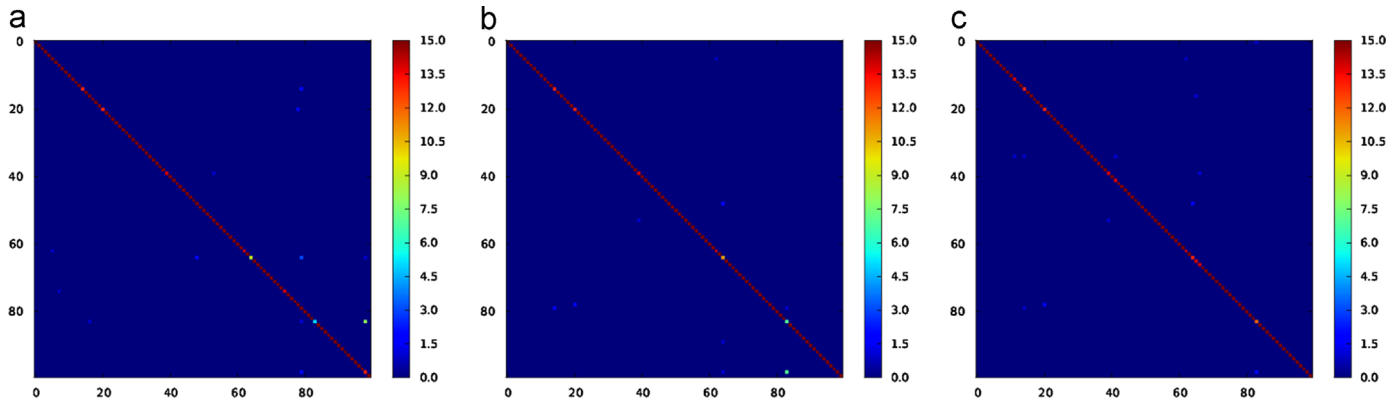

Fig. 18. Different confusion matrices for COIL database: (a) Using only PHOW as a descriptor (98.31%), (b) using gradient as a state definition (98.8%) and (c) using the proposed method (99.0%).

rate. Fig. 12 summarizes the proposed evaluation. The first two bins are simple BoFs, the first bin corresponds to the result obtained if the best single descriptor (PHOW 74.81%) is considered; the second one corresponds to the performance using the concatenation of all descriptors in a single vector; it reaches 76.3% of classification rate. Next, the result from the best single state definition, in this case the $L^*a^*b^*$, is presented. Then, the next two bins correspond to the results obtained concatenating all the states presented in Section 4.1.1 and the results obtained combining the states using the strategy presented in Section 4.3. In the case of state concatenation an 82.2% of classification rate is obtained while with the proposed approach an 83.7% is reached. Finally, the last bin corresponds to the result from the oracle in this database. Fig. 13 depicts the performance D1, D3 and D5 of Fig. 12 as confusion matrix. The first confusion matrix is obtained using the PHOW descriptor (Fig. 13(a)). Fig. 13(b) shows the performance using BoF with RL and $L^*a^*b^*$ state definition. Fig. 13(c) shows the results obtained when the proposed combination of Q-tables is used.

In order to validate the proposed approach, the experiments are repeated with a second database. This database is COIL [33]. The COIL database contains 100 classes. Fig. 14 shows one image per class. Each class contains 45 images and was split up into three sets: 15 images for training the BoF, 15 images for training the Q-table and 15 images for testing.

The performance of BoF using a single descriptor is depicted in Fig. 15 where the best option is PHOW with 98.3% of classification rate. Fig. 16 shows the performance for each of the state definition described above (Section 4.1.1) and Fig. 17 summarizes the proposed evaluation. It can be appreciated that in this case the best state definition corresponds to the gradient, which reaches a 98.8% of classification rate. It can be seen that the combination of Q-tables reaches 99.0% of classification rate (see Fig. 17).

Fig. 18 shows the confusion matrices with the best results. Fig. 18(a) shows the confusion matrix of PHOW as a descriptor (98.3% of classification rate). The second confusion matrix uses BoF with RL and gradient as a state definition, in this case the classification rate is 98.8%. Finally, Fig. 18(c) shows the confusion matrix for the proposed approach that combines multiple Q-tables.

In this second experiment the behavior of the approaches is similar, because all state definitions improve the classification rate with respect to the single descriptor case. Also, the combination of all descriptors reaches the same classification rate as the combination of Q-tables, but we work with less information from the image, thus, we use less computer resources (space and time). The scenes classified by our approach (D2) can be different from those correctly classified in D4. Note that although there is still space for improving, the obtained result is quite near to the best result, which is only reached by the oracle (99.7% of classification rate).

6. Conclusions and future work

This paper presents a novel framework for visual object classification. In particular, it is focussed on the selection of the best image feature descriptor. It is based on the combined use of a bag of features scheme together with a reinforcement learning technique, implemented trough the Q-learning approach. Note that any visual classification method (based on image descriptors) can substitute the BoF in this approach.

The proposed method combines different state definitions in a multi-table strategy that guarantees the selection of the best action (image descriptor). Experimental results using two public databases and comparisons with state of the art are provided showing the performance of the proposed approach. Note that, the presented approach uses only the best descriptor, thus, it significantly reduces the computational resources (space and time).

The future work will be focused on the combination of image descriptors and a modification to the framework in order to incorporate a human feedback to the learning step. There are scenes that cannot be recognized using only one descriptor. Our current architecture is not able to cope with these cases. However, the architecture can be modified in order to learn a combination of descriptors for each image. Exploring supervised reinforcement learning is another field of interest. The apprenticeship in reinforcement learning interacts with a human in order to reach a better learning step (e.g., faster learning, improved performance).

Acknowledgements

This work was partially supported by the Spanish Government under Project TIN2011-25606. Monica Piñol was supported by Universitat Autònoma de Barcelona grant PIF 471-01-8/09.

References

[1] J. Peng, J. Peng, B. Bhanu, Local reinforcement learning for object recognition, in: Proceedings of Fourteenth International Conference on Pattern Recognition, vol. 1, Washington, DC, USA, 1998, pp. 272–274.
[2] D. Lowe, Distinctive image features from scale invariant keypoints, Int. J. Comput. Vis. 2 (2004) 91–110.
[3] L. Fei-Fei, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, San Diego, CA, USA, 2005, pp. 524–531.

[4] L. Bo, X. Ren, D. Fox, Depth kernel descriptors for object recognition, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 2011, pp. 821–826.
[5] C. Harris, M. Stephens, A combined corner and edge detector, in: Alvey vision conference, vol. 15, Manchester, UK, 1988, p. 50.
[6] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors, Found. Trends Comput. Graph. Vis. 3 (3) (2008) 177–280.
[7] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on Statistical Learning in Computer Vision, Proceedings of the European Conference on Computer Vision, vol. 1, Prague, Czech Republic, 2004, pp. 1–22.
[8] H. Bay, T. Tuytelaars, L.V. Gool, Surf: speeded up robust features, in: Proceedings of the European Conference on Computer Vision, Springer, Graz, Austria, 2006, pp. 404–417.
[9] A. Bosch, A. Zisserman, X. Muñoz, Image classification using random forests and ferns, in: Proceedings of IEEE International Conference on Image Processing, IEEE, Rio de Janeiro, Brazil, 2007, pp. 1–8.
[10] I. Everts, J.C. van Gemert, T. Gevers, Per-patch descriptor selection using surface and scene properties, in: Proceedings of the European Conference on Computer Vision, Springer, Florence, Italy, 2012, pp. 172–186.
[11] M. Piñol, A.D. Sappa, A. López, R. Toledo, Feature selection based on reinforcement learning for object recognition, in: Adaptive Learning Agent Workshop, Valencia, Spain, 2012, pp. 4–8.
[12] M. Shokri, H.R. Tizhoosh, A reinforcement agent for threshold fusion, Appl. Soft. Comput. 8 (1) (2008) 174–181.
[13] F. Sahba, H.R. Tizhoosh, M. Salama, Application of opposition-based reinforcement learning in image segmentation, in: IEEE Symposium on Computational Intelligence in Image and Signal Processing, Honolulu, HI, USA, 2007, pp. 246–251.
[14] M.T. Harandi, M.N. Ahmadabadi, B.N. Araabi, Face recognition using reinforcement learning, in: Proceedings of IEEE International Conference on Image Processing, vol. 4, IEEE, Singapore, 2004, pp. 2709–2712.
[15] S. Ebert, M. Fritz, B. Schiele, Ralf: a reinforced active learning formulation for object class recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, USA, 2012, pp. 3626–3633.
[16] K. Häming, G. Peters, Learning scan paths for object recognition with relational reinforcement learning, in: Proceedings of IASTED International Conference on Signal Processing, Pattern Recognition and Applications, vol. 678, Innsbruck, Austria, 2010, p. 253.
[17] S. Jodogne, Reinforcement learning of perceptual classes using q learning updates, in: Proceedings of the 23rd IASTED International Multi-Conference on Artificial Intelligence and Applications, Innsbruck (Austria), 2005, pp. 445–450.
[18] S. Jodogne, J.H. Piater, Interactive selection of visual features through reinforcement learning, in: Proceedings of the 24th SGAI Internal Conference on Innovative Techniques and Applications of Artificial Intelligence, vol. 21, Cambridge (UK), 2004, pp. 285–298.
[19] R. Bianchi, A. Ramisa, R. de Mántaras, Automatic selection of object recognition methods using reinforcement learning, Adv. Mach. Learn. I 262 (2010) 421–439.
[20] T. Leopold, G. Kern-Isberner, G. Peters, Belief revision with reinforcement learning for interactive object recognition, in: Proceedings of the European Conference on Artificial Intelligence, vol. 178, Patras, Greece, 2008, pp. 65–69.
[21] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, 1st Edition, MIT Press, Cambridge University Press, MA(USA), 1998.
[22] C.J.C.H. Watkins, Learning from delayed rewards (Ph.D. thesis), King's College, Cambridge UK, 1989.
[23] T.M. Mitchell, Machine Learning, McGraw-Hill Science/Engineering/Math, 1997.
[24] B. Leibe, B. Schiele, Analyzing appearance and contour based methods for object categorization, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE Computer Society, Los Alamitos, CA, USA, 2003, p. 409.
[25] M.A. Ruzon, C. Tomasi, Edge, junction, and corner detection using color distributions, IEEE Trans. Pattern Anal. Machine Intell. 23 (11) (2001) 1281–1295.
[26] D.R. Martin, C.C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, IEEE Trans. Pattern Anal. Mach. Intell. 26 (5) (2004) 530–549.
[27] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 27 (10) (2005) 1615–1630.
[28] K.E.A. van de Sande, T. Gevers, C.G.M. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1582–1596.
[29] S. Lazebnik, C. Schmid, J. Ponce, A sparse texture representation using local affine regions, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1265–1278.
[30] A. Vedaldi, B. Fulkerson, Vlfeat: an open and portable library of computer vision algorithms, ⟨http://www.vlfeat.org/⟩2008.
[31] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1979) 62–69.
[32] R. Haralick, L. Shapiro, Computer and Robot Vision, vol. 1, Addison-Wesley, 1992.
[33] S.A. Nene, S.K. Nayar, H. Murase, Columbia Object Image Library (COIL-100), Technical Report (February 1996).

Monica Piñol Naranjo received the computer science degree from the Universitat Autònoma de Barcelona, Barcelona, Spain, in 2009. In the same year, she joined the Computer Vision Center, Barcelona; in 2010 she received the Master degree in Computer Vision and Artificial Intelligence from the same university. Currently she is pursuing her Ph.D. degree working on reinforcement learning approaches applied to computer vision domain.

Angel Domingo Sappa received the electromechanical engineering degree from the National University of La Pampa, General Pico, Argentina, in 1995 and the Ph.D. degree in industrial engineering from the Polytechnic University of Catalonia, Barcelona, Spain, in 1999. In 2003, after holding research positions in France, U.K., and Greece, he joined the Computer Vision Center, Barcelona, where he is currently a Senior Researcher. His current research focuses on stereo image processing and analysis, 3-D modeling, and dense optical flow estimation. His research interests span a broad spectrum within the 2-D and 3-D image processing. Dr. Sappa is a member of the Advanced Driver Assistance Systems Group, Computer Vision Center.

Ricardo Toledo received the degree in Electronic Engineering from the Universidad Nacional de Rosario (Argentina) in 1986, the M.Sc. degree in image processing and artificial intelligence from the Universitat Autònoma de Barcelona (UAB) in 1992 and the Ph.D. in 2001.

Since 1989 he has been giving lectures at the Computer Science Department of the UAB and participating in R+D projects. Currently he is a full time associated professor. In 1996 he participated in the foundation of the Computer Vision Center (CVC) at the UAB. Ricardo has participated in national and international R+D projects being the leader of some of them, and is coauthor of more than 40 articles, all these in the field of computer vision, robotics and medical imaging.