Pattern Recognition 45 (2012) 231–240

Contents lists available at ScienceDirect

Pattern Recognition

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

Representing and recognizing objects with massive local image patches

Liang Lin [a], Ping Luo [a], Xiaowu Chen [b], Kun Zeng [a,*]

[a] School of Software, Sun Yat-Sen University, Guangzhou 510006, China
[b] School of Computer Science and Engineering, Beihang University, Beijing 100191, China

ARTICLE INFO

Article history:
Received 14 October 2010
Received in revised form
9 May 2011
Accepted 16 June 2011
Available online 19 July 2011

Keywords:
Object recognition
Object detection
Generative learning

ABSTRACT

Natural image patches are fundamental elements for visual pattern modeling and recognition. By studying the intrinsic manifold structures in the space of image patches, this paper proposes an approach for representing and recognizing objects with a massive number of local image patches (e.g. 17×17 pixels). Given a large collection ($> 10^4$) of proto image patches extracted from objects, we map them into two types of manifolds with different metrics: explicit manifolds of low dimensions for structural primitives, and implicit manifolds of high dimensions for stochastic textures. We define these manifolds grown from patches as the "ε-balls", where ε corresponds to the perception residual or fluctuation. Using these ε-balls as features, we present a novel generative learning algorithm by the information projection principle. This algorithm greedily stepwise pursues the object models by selecting sparse and independent ε-balls (say 10^3 for each category). During the detection and classification phase, only a small number (say 20) of features are activated by a fast KD-tree indexing technique. The proposed method owns two characters. (1) Automatically generating features (ε-balls) from local image patches rather than designing marginal feature carefully and category-specifically. (2) Unlike the weak classifiers in the boosting models, these selected ε-ball features are used to explain object in a generative way and are mutually independent. The advantage and performance of our approach is evaluated on several challenging datasets with the task of localizing objects against appearance variance, occlusion and background clutter.

1. Introduction

Representing and recognizing patterns with natural image patches is a fundamental research topic in computer vision and pattern recognition. In recent years, a number of patch-based models have been proposed [20,5,1,11], and these studies show that image patches are distributed in the space of various manifolds corresponding to structural primitives, stochastic textures, and hybrid patterns. This paper studies a general approach for representing and recognizing complex object patterns with a massive number of image patches.

1.1. Related work

In the literature of object recognition, there has been a large amount of efforts on studying features with image patches [20,8,11,25,5,19,29] and learning object models with discriminative or generative methods [4,24,13,9,21,6].

From the feature designing point of view, we consider three types of descriptors according to different image properties. Fragment or primitive features [9,1,18] explicitly localize locations of edges/bars and capture information of geometric structures or boundaries. Stochastic textural or cluttered image patches tend to be well captured by histogram descriptors [2,12,19]. In addition, there are a large amount of hybrid image patches, particularly from high level meaningful objects, containing both structures and textures; to precisely define patterns on these image patches, compositional or hierarchical representations are widely adopted [16,21]. Essentially, these different types of features (or representations) correspond to different metrics of complexity.

From the point of view of object model learning, many discriminative methods, such as Adaboost [7,24,10] and SVM-based algorithms [6,29], achieve very good performance on several public benchmarks. Discriminative learning can be viewed as sequentially adding new predictors to approximate the conditional distribution of the class label. Despite the acknowledged success, their modeling capability is limited since they are focusing on classification boundaries rather than the generation process of the data [23]. In contrast, the generative learning methods, like FRAME model [30], Induction learning model [3] and POE model [27], are more desirable as they pursue the underlying distribution of image data of interest. However, the

* Corresponding author.
E-mail addresses: linliang@ieee.org (L. Lin), chen@buaa.edu.cn (X. Chen), zengkun@gmail.com (K. Zeng).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.06.011

generative learning methods often limited by the inefficiency due to the following reasons: (i) the loss function in generative learning is often based on image likelihood that leads to complex computation for normalization and (ii) a sampling process is often needed in the feature selection stage.

1.2. Method overview

In this paper, we present a method to learn the generative object model by spanning (growing) two types of manifolds, namely "ε-balls", with massive image patches. The key contributions of this paper are summarized: (i) a comprehensive discussion of growing manifolds from massive image patches and (ii) a novel generative learning algorithm by maximizing information gain.

(I) We study the manifolds of image patch space based on this observation that the primitive features are often generated by base functions to reconstruct the observed image patches and the histogram features are described by statistical constraints to distinguish image patches against noise images.

We study two types of pure manifolds: a low-dimensional manifold by a generative function with a small number of variables, called "explicit manifold", and a high-dimensional manifold by histograms, called "implicit manifold". The former is often referred as the "texton" model in coding theories and the latter as the Julesz Ensemble theory. Inspired by the previous work of image representation [2,15], we define explicit manifold by a dictionary of 2D Gabor wavelets and implicit manifold by histograms of oriented gradients.

In our method, we first extract a massive collection ($>10^4$) of image patches from the training object instances. The image patch size ($n \times n$) is in a small range, i.e. $n \in [17,23]$, and the aspect ratio is not limited, accounting for object transformation. Then we map these image patches with the two types manifold metrics. Each patch is spanned to a few atomic manifolds plus an additive residual or fluctuation ε. We call these manifolds as "ε-balls" in the sense that they explain objects independently and generatively. To encode object spatial layout and variance, these ε-balls are position-sensitive with respect to the objects, inspired by the implicit shape model [14].

Moreover, the mutual correlation of these ε-balls is calculated simultaneously in the ε-ball generating process, which play an important role in learning object models. In practice, we can simply prune redundant ε-balls, which overlap to others heavily.

(II) A generative learning algorithm, together with the ε-balls, is proposed for learning generative object model. This algorithm stepwisely selects sparse ε-balls by an information gain criterion. The information gain for each feature (ε-ball) is defined based on its consistency within the training examples and its distinctiveness from the statistics of reference examples (i.e. generic images or examples from other categories). In the algorithm, we first calculate the initial information gain for each feature, and then iterate two steps for pursuing the object model: (i) selecting the most informative feature that has maximum information gain and (ii) updating the information gains of the rest features according to their correlations to the one just been selected.

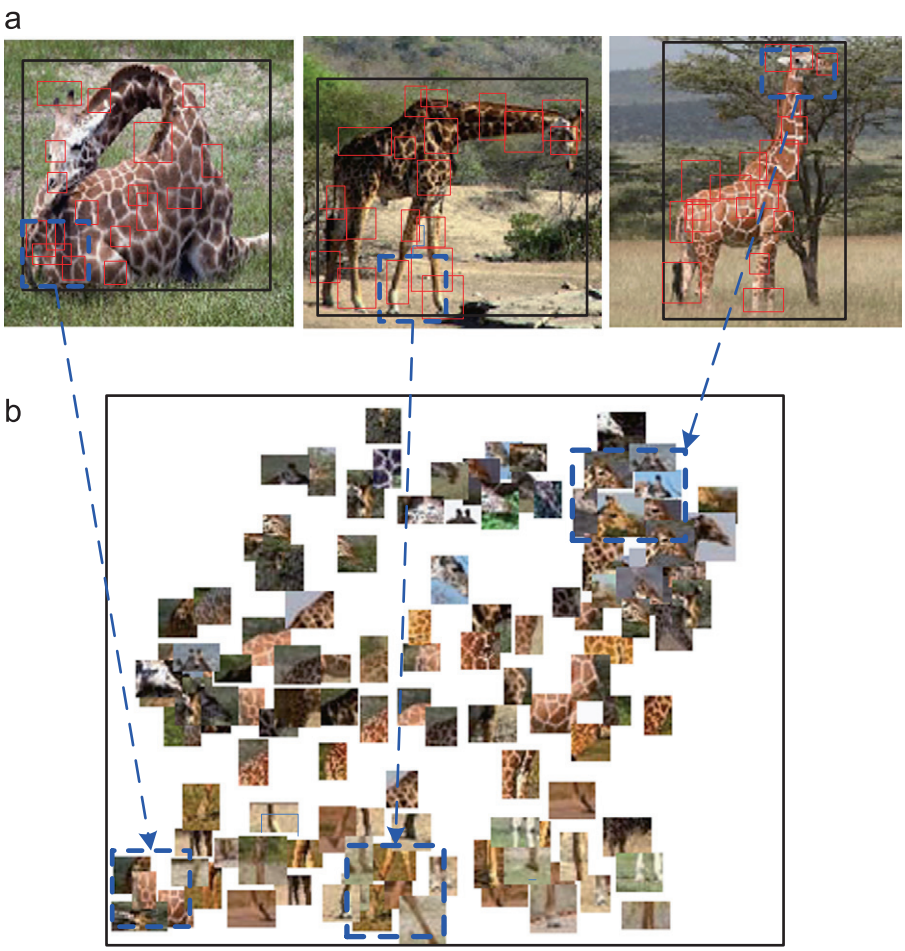

Fig. 1. Learning object sparse models from massive image patches. (a) A few activated patches (i.e. 15), denoted by red rectangles, when detecting and localizing objects from images. (b) A learned object model with a number (i.e. 1000) of ε-balls (manifolds spanned from patches). The relative locations of ε-balls are encoded with respect to objects (the black boxes). The dashed blue boxes and arrows imply how the object variance is covered or explained by the ε-balls. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

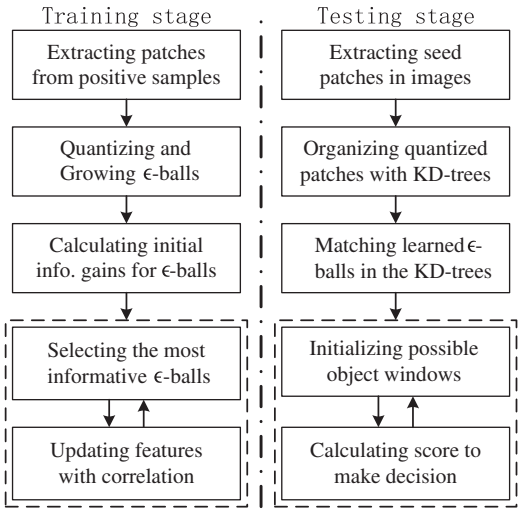

Fig. 2. The summarized sketch of the proposed method. The two dashed frames indicate the iteration processes.

Compared to the previous generative learning approaches, our algorithm has the advantages in computation efficiency by the following aspects. (i) We calculate the image likelihood (i.e. matching ε-balls into images) only once for the initial information gain. (ii) In the object model pursuit procedure, the feature weights and normalization term can be fast and analytically computed using the feature correlations.

As illustrated in Fig. 1, the proposed model consists of a batch of ε-balls from a massive patch collection, and is able to detect objects by activating very sparse and few patches (see Fig. 1(a)). If we draw samples to form a distribution from a learned model as shown in Fig. 1(b), it implicitly includes several object patch-based templates for covering variance in object category.

In summary, the pipeline of the proposed method with the training and testing stages is presented in Fig. 2. The two dashed frames indicate the iteration processes. In the training stage, we first extract the proto image patches from only positive object images and generate a large number of ε-ball features with the manifold metrics, and stepwisely select features to pursue the object model in an iteration of two steps. In the testing stage, instead of exhaustively sliding windows, we present a heuristic method for proposing detecting windows using the KD-tree technique [17]. Given a testing image, we first extract a batch of seed patches over scales and locations, and map them with the two manifold metrics. These patches are saved in 2 KD-trees. We match the ε-balls of object models into the 2 KD-trees by nearest neighbor searching, and then propose a number of possible object windows based on a few matched seed patches. The decision in each window is finally made with the complete object model.

The remainder of this paper is arranged as follows. We first present the image patch mapping with two types of manifold metrics in Section 2, and follow with a description of ε-ball features in Section 3. The learning algorithm for pursuing object models is introduced in Section 4. The quantitative experimental results are shown in Section 5 and the paper is concluded with a summary in Section 6.

2. Manifold metrics of image patches

In this work, we are not trying to capture the manifolds of image patches, but to span image patches into two types of atomic manifolds and learn generative and sparse object models with them.

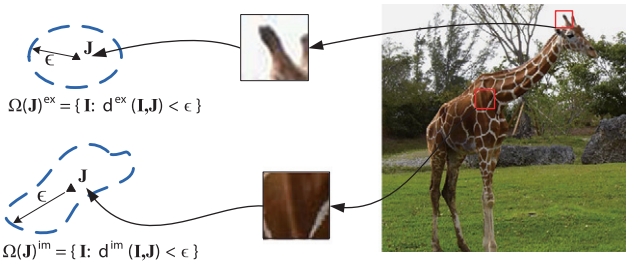

Fig. 3. Two examples of explicit and implicit manifolds spanned from image patches. The former is in low dimension (implied by a regular ellipse) and the latter is in high dimension (implied by a irregular circle).

Given a set of object images, we first randomly extract a large number of image patches, namely proto patches, at a range of size and aspect ratio

$$\mathbf{S}^{proto} = \{\mathbf{J}_1, \mathbf{J}_2, \ldots, \mathbf{J}_M\}, \tag{1}$$

where an instance of proto patch, $\mathbf{J}_i = (u_i, v_i, \Lambda_i)$, includes the width and height (u_i, v_i), and the image domain Λ_i (both u_i and v_i are in a small discreted range, 17−23, in our experiments). The flatness patches that are smooth and textureless (e.g. patches from white wall) can be simply removed due to being non-informative. In practice, the number of proto patches $M \geq 10^4$.

Then we introduce mapping these patches with two types of manifold metrics, as two exemplars in Fig. 3. In the figures of this paper, we use regular ellipses to indicate the explicit manifolds in lower dimensions and use the irregular circles (with variational radiuses) to indicate the implicit manifolds in higher dimensions.

2.1. Explicit manifolds

Definition. An explicit manifold $\Omega(\mathbf{J})^{ex}$ spanned from a proto patch $\mathbf{J}$ is defined by a low-dimensional function $\Phi(\mathbf{J})$ with a small residual ε

$$\Omega(\mathbf{J})^{ex} = \{\mathbf{I} : \mathbf{d}^{ex}(\mathbf{I},\mathbf{J}) = |\Phi(\mathbf{I}) - \Phi(\mathbf{J})| < \varepsilon\}. \tag{2}$$

Intuitively, $\Omega(\mathbf{J})^{ex}$ is an equivalence class of patch $\mathbf{J}$, which the instances belonging to share the similar structure pattern with $\mathbf{J}$.

Here we define $\Phi(\mathbf{J})$ by the recently proposed Active Basis model [26] that well represents deformable shapes for image patches. This model is a natural generalization of the sparse coding model and texton model, which uses a dictionary of specified Gabor wavelets to encode the structures of image patches.

As illustrated in Fig. 4(b), a Gabor wavelet is a linear filter used for edge detection. Frequency and orientation representations of Gabor filter are similar to those of human visual system [15]. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. We denote the dictionary of wavelet elements as $\mathcal{A} = \{g_{x,y,s,\alpha}, \forall(x,y,s,\alpha)\}$ including the attributes: central position, scale, and orientation. (x,y,s,α) are densely sampled: (x,y) with a fine sub-sampling rate (e.g. every 2 pixels), and α with every 18°.

Thus we define the low dimension function $\Phi(\mathbf{I})$ by matching (convoluting) Gabor wavelets with the image patch $\mathbf{I}$,

$$\Phi(\mathbf{I}) = \sum_{g_i \in \mathcal{A}} |\langle \mathbf{I}, g_i \rangle|^2, \tag{3}$$

where $\langle \cdot, \cdot \rangle$ is the convolution function. Two image patches including similar structure patterns can be measured by the distance metric,

$$\mathbf{d}^{ex}(\mathbf{I},\mathbf{J}) = -1/n(\mathcal{A}) \sum_{g_i, g_j \in \mathcal{A}} \delta(|\langle g_i, \mathbf{I} \rangle|^2 - |\langle g_j, \mathbf{J} \rangle|^2), \tag{4}$$

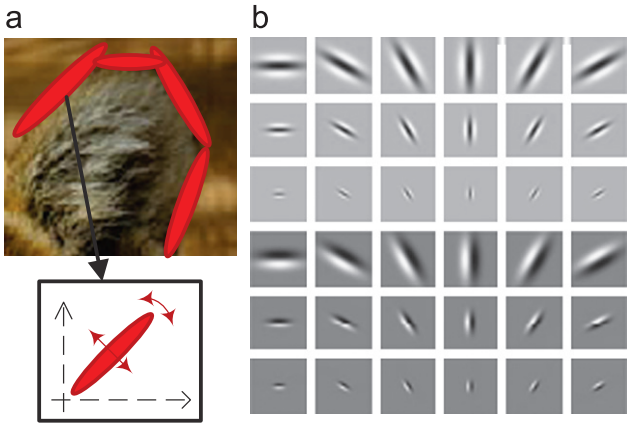

Fig. 4. A dictionary of Gabor wavelets is adopted to represent image patches in an explicit manifold. Each wavelet (denoted by red ellipses) is allowed to slightly perturb at location and orientation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

subject to the constraints, $g_i \approx g_j$. The total number of Gabor wavelets for the image patches, $n(\mathcal{A})$, is used for normalization. δ denotes a sigmoid-like transformation, (i.e. a monotone function), which is adopted to make the distance continuous. $g_j \approx g_i$ indicates that the two wavelets are equivalent under a perturbation as

$$\begin{cases} |x_i - x_j| \le \xi \cdot \sin \alpha_i, \\ |y_i - y_j| \le \xi \cdot \cos \alpha_i, \\ |\alpha_i - \alpha_j| < \kappa, \end{cases} \tag{5}$$

where (ξ,κ) are the perturbation parameters. We allow the local perturbation of wavelets to tolerate local structural difference, as illustrated in Fig. 4(a). In our experiments, we set $\xi = 2$ pixels and $\kappa = \pi/36$. Note that we need to normalize patches with the same size, say 20×20, before computation.

2.2. Implicit manifolds

Definition. An implicit manifold $\Omega(\mathbf{J})^{im}$ spanned from a proto patch $\mathbf{J}$ is defined by a histogram $\mathbf{Hist}(\mathbf{J})$ of Markov random fields or feature statistics plus a small statistical fluctuation ε

$$\Omega(\mathbf{J})^{im} = \{\mathbf{I} : \mathbf{d}^{im}(\mathbf{I},\mathbf{J}) = \mathcal{K}(\mathbf{Hist}(\mathbf{I}) \| \mathbf{Hist}(\mathbf{J})) < \varepsilon\}, \tag{6}$$

where $\mathcal{K}(\cdot)$ is the Kullback–Leibler divergence. This type of manifolds is very different with the explicit manifolds. The image patches in $\Omega(\mathbf{J})^{im}$ cannot be explicitly identified by a small number of variables, but tend to have similar textural appearance with patch $\mathbf{J}$ in terms of a number of implicit descriptions (constraints).

In recent study of texture recognition, the most popular feature is the HoG descriptor (histogram of oriented gradients) [2]. In the HoG representation, the image domain is divided into a number of, i.e. 6×8, regular cells; at each pixel, a gradient is calculated, and a histogram is pooled over each cell for different orientations. The histograms from the cells are concatenated into a long descriptor for recognition. In our method, we simplify the HoG descriptor by discarding the cell division, and directly extract the orientation histogram over pixels. Moreover, following the recently proposed work in image feature [19], we calculate the oriented gradients in RGB color space instead of image intensity, and transform color channels by normalizing the pixel value distributions, which has been verified to well increase illumination invariance and discriminative power. The RGB values in the image patch are transformed by

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} \frac{R-\mu_R}{\sigma_R} \\ \frac{G-\mu_G}{\sigma_G} \\ \frac{B-\mu_B}{\sigma_B} \end{pmatrix}, \tag{7}$$

where μ and σ denote the mean and standard deviation of the distribution in each channel. Then the histogram of oriented gradients is pooled over the image domain at three transformed color channels, which is discretized into 24×3 bins.

3. Image features via ε-balls

We visually define the explicit and implicit manifolds from proto patches as the ε-balls, and ε is referred to the precision of representation or perception. For each ε-ball, the precise value of ε can be decided in a growing process.

For each proto patch $\mathbf{J}$, we generate k (i.e. $k = 3$ in our experiments) ε-balls, according to the number of other patches falling into the ball. In Fig. 5, we illustrate the idea of growing a ε-ball with two types of manifold metrics, based on a proto patch $\mathbf{J}$. We consider two big similarity matrices in the two types of manifold metrics as

$$\mathbf{D}^r = [\mathbf{d}^r(\mathbf{J}_i,\mathbf{J}_j)], \quad \mathbf{d}^r(\mathbf{J}_i,\mathbf{J}_i) = 0, \ r \in \{\text{'ex'},\text{'im'}\}, \tag{8}$$

where we can explore the pairwise similarity among patches. For example, for an ε-ball spanned from a proto patch $\mathbf{J}$, if $\mathbf{d}^r(\mathbf{J},\mathbf{J}') < \varepsilon$, then the patch $\mathbf{J}'$ falls into the ball. Therefore, starting from an initial small value ε_0, we increase ε for each ε-ball to count the number of other patches falling to the ball. In our work, for each proto patch, the discretized value of ε is decided by the ball containing 0.1%, 0.2% and 0.4% amount of total proto patches. Assuming we have a number M of proto patches, we can thus obtain $2 \times 3 \times \mathrm{M}$ ε-balls in total.

For simplicity, we denote all manifolds (including both explicit and implicit) by $\Omega_z, z = 1,\ldots,6M$, and define the dictionary of ε-balls as

$$\mathbb{B} = \{\mathbf{B}_z = \Omega_z = (\mathbf{J}_z,\varepsilon_z,r_z,X_z),\ z = 1,\ldots,6M\}, \tag{9}$$

where $r_z \in \{\text{'ex'},\text{'im'}\}$ indicates the type of manifold metrics. X_z is the relative location of patch $\mathbf{J}$ with respect to object center.

An ε-ball $\mathbf{B}_z$ is essentially a generative image basis for representing objects. Letting $\mathbf{O}_i$ be an object with its label $l_i \in \{0,1\}$, $\mathbf{B}_z$ is written in the form of a feature,

$$\mathbf{h}_z(\mathbf{O}_i) = \begin{cases} 1, & \mathbf{d}^{r_z}(\mathbf{J}_z,\mathbf{I}_j) \le \varepsilon_z, \ \exists \mathbf{I}_j \text{ s.t. } X_j \approx X_z, \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

where $\mathbf{I}_j$ is a patch from object instance $\mathbf{O}_i$ at location X_j, and $X_j \approx X_z$ indicates the locations of $\mathbf{I}_j$ and $\mathbf{J}_z$ are roughly matched with respect to the object. Unlike the discriminative boundary, the ε-balls "turn off and keep silence" (equal to zero) to the instance falling out of them.

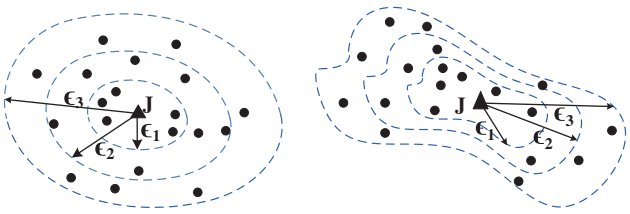

Fig. 5. The growing of ε-balls in two types of manifold metrics.

In addition, the mutual pairwise correlations of these ε-balls are calculated simultaneously in the growing process, which play an important role in learning object models. Following the theory of Pearson Correlation Coefficient in Statistics, we define the mutual correlation between two ε-balls based on the observed data (i.e. representing all the proto image patches) as

$$C(\mathbf{h}_i|\mathbf{h}_j)=\frac{\sum_{m=1}^{M}\mathbf{h}_i(\mathbf{J}_m)\cdot\mathbf{h}_j(\mathbf{J}_m)}{\sum_{m=1}^{M}\mathbf{h}_j(\mathbf{J}_m)}. \tag{11}$$

Note the correlation of each ε-ball pair is a non-symmetric measurement. For example, if the ball $\mathbf{B}_i$ is "included" by $\mathbf{B}_j$, then $C(\mathbf{h}_i|\mathbf{h}_j)=1$ and $C(\mathbf{h}_j|\mathbf{h}_i)<1$. In practice, we can simply prune the redundant ε-balls that are included by others, and the ones that heavily overlap to others.

4. Generative object models pursuit

Together with the ε-balls as features, we propose a generative learning algorithm by information projection principle to pursue object models that consist of a batch of sparse and independent ε-balls to cover or explain object variance.

4.1. Training procedure

Let the training set includes N object instances,

$$\mathbf{S}^{Train}=\{(\mathbf{O}_i,l_i),\ i=1,\ldots,N\},\quad l_i\in\{1,0\}, \tag{12}$$

where l_i indicates the object instance $\mathbf{O}_i$ is positive or reference. It is worth mentioning that we randomly select a few object instances from $\mathbf{S}^{Train}$ to extract proto image patches.

Given a massive collection of ε-ball features $\{\mathbf{h}_z=\mathbf{B}_z, z=1,\ldots,6M\}$, we present a generative learning algorithm for pursuing object models by iterative feature selection. There are two key steps: (i) calculating initial feature information gain and (ii) updating the object model by stepwise feature pursuit. Compared to the previous generative learning algorithms, our method has the advantages in computation efficiency.

Intuitively, pursuing generative and sparse object models can be viewed as finding a sequence of ε-balls from a starting or initial model q_0. At each step t, the model p is updated to graduate approach the underlying target model f,

$$q_0=p_0\rightarrow p_1\rightarrow\cdots\rightarrow p_t \text{ to } f, \tag{13}$$

in terms of minimizing the Kullback–Lebler divergence $\mathcal{K}(f\|p)$. In the manner of iterative pursuing, the new model p_t is updated by adding a new ε-ball features $\mathbf{h}_t$ based on the current model p_{t-1}:

$$p_t^*=\arg\min \mathcal{K}(p_t||p_{t-1}), \tag{14}$$

subject to a new constraint, using new feature $\mathbf{h}_t$

$$E_{p_{t-1}}[\mathbf{h}_t]=E_f[\mathbf{h}_t]. \tag{15}$$

Intuitively, we search the optimal new model p_t^* closest to current model p_{t-1}, minimizing $\mathcal{K}(p_t||p_{t-1})$, because the previous constraints in p_{t-1} should be preserved. $E_f[\mathbf{h}_t]$ denotes the expectation of the feature $\mathbf{h}_t$ over the target model, i.e. the marginal distribution projected into $\mathbf{h}_t$, which can be always approximated by the sample mean. $E_{p_{t-1}}[\mathbf{h}_t]$ denotes the feature expectation on current model, which can be directly calculated.

Solving this constrained optimization problem by Lagrange multiplier in Eq. (14), we have

$$p_t=\frac{1}{\mathbf{Z}_t}p_{t-1}\exp\{\lambda_t\mathbf{h}_t\}, \tag{16}$$

λ_t is the weight for feature $\mathbf{h}_t$ and Z_t normalizes the probability to 1. Thus we can further derive the probabilistic object model as (here we write with a notation complete form)

$$p(\mathbf{O}_i)=\frac{1}{\mathbf{Z}}\exp\left\{\sum_{t=1}^{T}\lambda_t\mathbf{h}_t\right\}q_0(\mathbf{O}_i), \tag{17}$$

where $\mathbf{Z}=\prod\mathbf{Z}_t$ is a normalization term and T is the iteration number for model updating. The more details of the derivation can be referred in [30,3].

The feature selection at each round t is equivalent to finding the most informative feature having maximum information gain to the current model $p(\mathbf{O}_i)_{t-1}$,

$$\mathbf{h}_t^*=\arg\max_{h\in\mathbb{B}}\mathcal{G}_t,$$

$$\mathcal{G}_t=\mathcal{K}(p_t||p_{t-1})=(\lambda_t\mathbf{h}_t-\log\mathbf{Z}_t) \tag{18}$$

and

$$\lambda_t=\log\frac{f_t^{on}(1-p_{t-1}^{on})}{p_{t-1}^{on}(1-f_t^{on})},$$

$$\mathbf{Z}_t=e^{\lambda_t}p_{t-1}^{on}+1-p_{t-1}^{on}, \tag{19}$$

where we denote $f_t^{on}=E_f(h_t)$ and $p_{t-1}^{on}=E_{p_{t-1}}(h_t)$ for notation simplicity. $\mathcal{G}_t$ measures the information gain after adding the new feature $\mathbf{h}_t$. The information gain essentially measures the contribution of each feature to object model pursuit, and it can be obtained by calculating feature response over all training samples. The detailed derivation is provided in the Appendix section.

In some generative learning methods [30,3], it is high cost for calculating p_{t-1}^{on} for each round of model pursuit, since they need to draw samples for synthesizing current distribution from the model p_{t-1}. For efficiency consideration, we propose another alternative approach to update information gains of features by their mutual correlations.

At the initial stage, given the reference model q_0, the information gains of all features $\{\mathcal{G}_{z,1}\}$ can be easily computed as well as $(\lambda_{z,1},\mathbf{Z}_{z,1})$. Once a feature $\mathbf{h}_1$ is selected at the first round, if we can obtain the information gains of the rest features against the current model, $\mathcal{K}(p_{z,2}\|p_1)$, then the best feature at the second round can be then deterministically selected. As shown in Fig. 6, we present an intuitive explanation for updating information gain with correlations, where $\mathbf{h}_1$ and $\mathbf{h}_2$ are two correlated features in (a). The learning procedure in a geometric point of view is illustrated in Fig. 6(b) and the theoretical supports can be founded in [3] (the famous Pythagorean theorem). p_1 is the model by selecting feature $\mathbf{h}_1$, p_2 is the model by adding feature $\mathbf{h}_2$ to p_1, and $p_{2,0}$ indicates the model by selecting feature $\mathbf{h}_2$ without $\mathbf{h}_1$. The model $p_{2,0}$ is not really needed to compute, and the gain $\{\mathcal{G}_{2,1}\}=\mathcal{K}(p_{2,0}||q_0)$ can be obtained at the initial stage. By introducing an auxiliary model $Q_{2,1}$ that implies the correlated information from p_1 to $Q_{2,1}$, we obtain $\mathcal{K}(p_2||p_1)\approx\mathcal{K}(p_{2,0}||Q_{2,1})$. There is also a distinct explanation. The information gain by applying $\mathbf{h}_2$ to model p_1 is equivalent to the initial gain of $\mathbf{h}_2$ subtracted by the

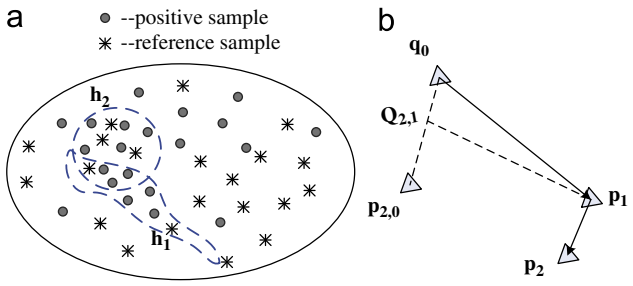

Fig. 6. An illustrative example of the information gain updating. (a) $\mathbf{h}_1$ and $\mathbf{h}_2$ are selected features at the first two rounds; (b) the geometric point of view of learning procedure.

correlated information with $\mathbf{h}_1$. That is, the additive information gain for each round can be approximated by initial gain with correlations. Interestingly, the correlation information between features are explicitly calculated in the ε-ball growing process.

Therefore, by analogy, once the feature $\mathbf{h}_t$ is selected at the round t, we can update the gains of the rest features according to their correlation to $\mathbf{h}_t$,

$$\mathcal{G}_{z,t+1} = (1-C(\mathbf{h}_z|\mathbf{h}_t))\mathcal{G}_{z,t} = \prod_{j=1}^{t}(1-C(\mathbf{h}_z|\mathbf{h}_j))\mathcal{G}_{z,0}. \tag{20}$$

Then the information gains of features are kept updating in each round, and we select the features by ranking the current information gains $\mathcal{G}_{z,t}$. $(\lambda_t,\mathbf{Z}_t)$ for selected feature $\mathbf{h}_t$ can be thus analytically solved in a very simple form

$$\lambda_t = \prod_{j=1}^{t-1}(1-C(\mathbf{h}_t|\mathbf{h}_j)) \cdot \lambda_{t,0},$$

$$\mathbf{Z}_t = \mathbf{Z}_{t,0}^{\prod_{j=1}^{t-1}(1-C(\mathbf{h}_t|\mathbf{h}_j))}. \tag{21}$$

In Fig. 7, we illustrate the process of selecting feature with updating weights for learning models of tiger category (more results are in Section 5). The weights of the 50 most informative features in the initial stage are plotted for the first six round iterations. Once a feature is selected (denoted by red color) at each round, the weights of correlated features will be decreased. Thus the independent and sparse features tend to be selected.

The learning process stops when no informative features exist, i.e. the information gains of all candidate features are within statistical fluctuation. We use a universal threshold τ on information gain as the stopping criterion of feature selection. τ is a very small number approaching to zero. For efficiency consideration, we set $\tau = 0.05$ in the implementation. The training procedure is summarized in Algorithm Box 1.

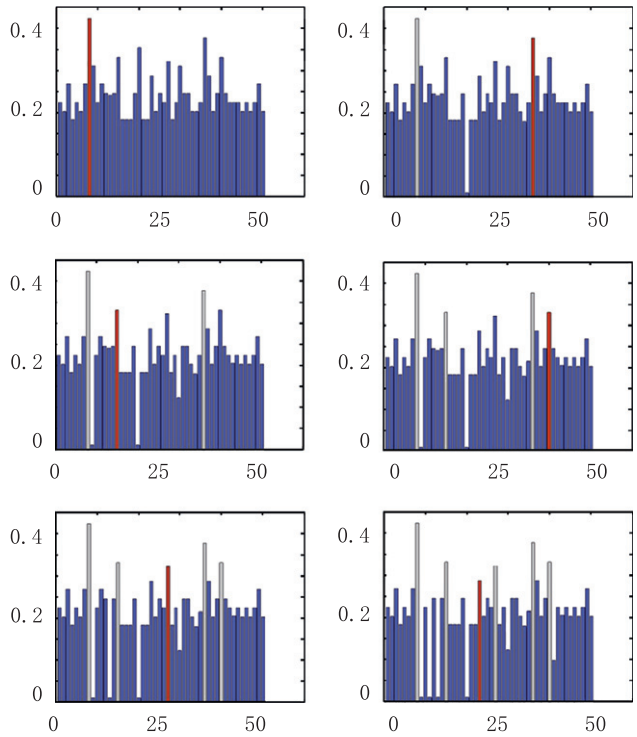

Fig. 7. Feature weights updating at the first six round iterations. The horizontal axe indicates 50 most informative features at initial stage, and the columns represent feature weights. In each cell, the horizontal axis and vertical axis represent the feature index and the corresponding feature weight, respectively; the features that have been selected are turned into gray, and the selected one at current model is denoted in red. Once a feature is selected at each round, the weights of correlated features will be decreased. Thus the selected features have little correlations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Testing procedure

With the learned object model, the object detection is formulated as calculating a log-likelihood ratio against background. Letting $\mathbf{I}^w$ denote the image in the testing window, the detect score is defined as

$$\log\frac{p(\mathbf{I}^w)}{q_0} = \sum_{t}^{T}\lambda_t\mathbf{h}_t(\mathbf{I}^w) - \log\mathbf{Z} > \eta, \tag{22}$$

where η is equal to zero in the theory and can be adjusted in practice.

Algorithm 1. Training procedure.

Input: A set of proto image patches $\mathbf{S}^{proto}$; A training set $\mathbf{S}^{Train}$.
Output: An object model $\mathbf{H}(\cdot)$.

1. Generate features with $\mathbf{S}^{proto}$;
   (1) Growing a set of ε-balls $\mathbb{B}$;
   (2) Computing correlation of ε-balls by Eq. (11);
2. Calculate initial information gain $\mathcal{G}_t$ with $\mathbf{S}^{Train}$;
3. Repeat for $t = 1$ to T;
   (1) Deterministically select a feature having maximum information gain;
   (2) Calculate the feature weight λ_t using Eq. (19);
   (3) Update the info. gains of the rest features according to their correlation using Eq. (21);
   (4) Stop when λ_t is lower than the stopping criterion τ;
4. Output the model as Eq. (17).

Instead of exhaustively sliding window for detecting in many approaches [24,13,6], we present a more efficient and flexible scheme by proposing detect windows with seed patches. This approach includes the following key steps. Its benefit is demonstrated in the experiments.

(1) Given a testing image, we first extract small patches $\{\mathbf{I}_i\}$ at the sampled locations (i.e. every 4–6 pixels). The patch size (u_i,v_i) is set as the same range as extracting proto patches $\mathbf{S}^{proto}$ in the training stage. These patches are treated as seeds for proposing candidate detect windows.

(2) We map the seed patches with two types of manifold metrics and organize them with 2 KD-trees [17], one for the explicit metric and the other for the implicit metric. Following [17], the randomized trees are built by choosing the split dimension randomly from the first D dimensions on which data has the greatest variance. We use the fixed value $D = 8$ in our implementation.

(3) The ε-balls in object models are matched to the patches in the 2 KD-trees. As reported by [17], the KD-trees provide large speedups with only minor loss in accuracy, due to reducing the feature dimensions in retrieval. Then a number of ε-balls are activated ("turned on") by the nearest neighbor searching. Note that no position information is taken into account when comparing ε-balls with image patches in KD-trees. For each ε-ball, we can further prune the candidate matches by testing with full dimension metric.

(4) Since the ε-balls are location-sensitive with respect to the object center, the detecting windows can be proposed according to the activated ε-balls in the testing images, as illustrated in Fig. 8(a). For example, assuming one ε-ball $\mathbf{B}_z=(\mathbf{J}_z,\varepsilon_z,r_z,X_z)$ matched to a patch $\mathbf{I}_i$ at location X_i of image, the location of generated detecting windows is X_i+X_z. In practice, we can propose windows by more than one matched patches, for the efficiency consideration, as illustrated in Fig. 8(b).

Fig. 8. Proposing detect windows with seed patches. The blue boxes indicate the matched seed patches by ε-balls and the red dashed boxes indicate the proposed detecting windows. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

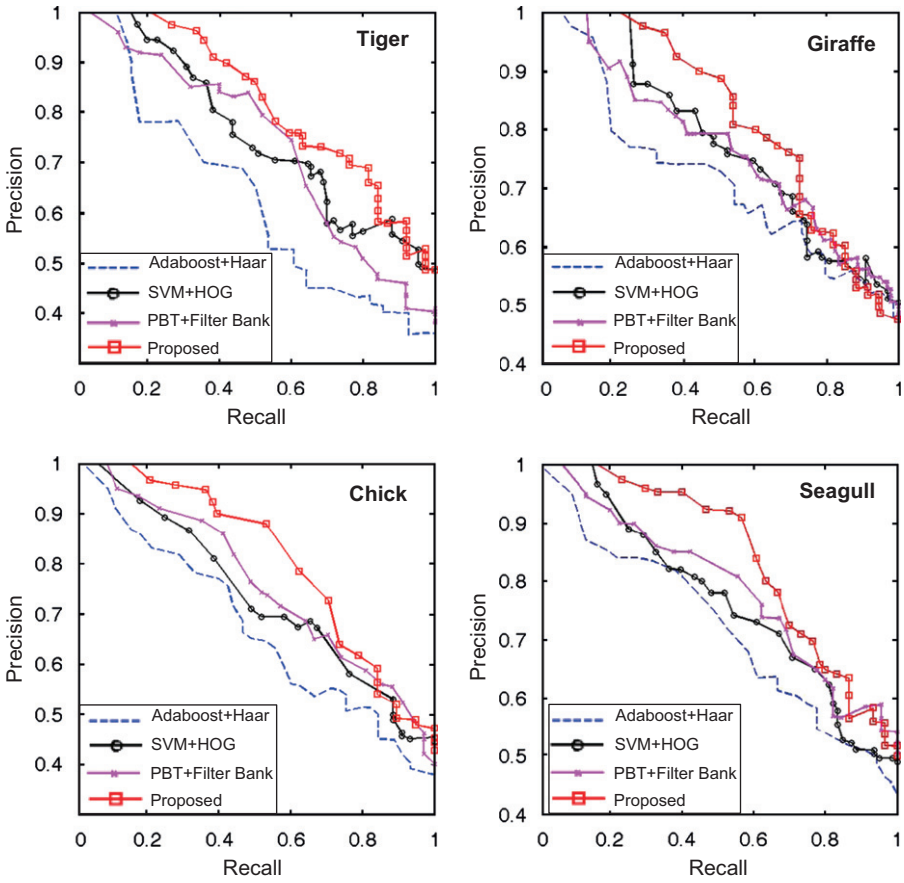

Fig. 9. The precision–recall curves of four categories on LHI database. Three discriminative methods, Cascaded Adaboost with Haar features [24], SVM with HOG features [2], and PBT with filter bank [22] are performed for comparison.

5. Experiments

We test our method with the vision task of detecting object from images, and compare with the state-of-art approaches. The number of selected ε-balls for each category is reported, as well as the number of activating ε-balls when detecting. The efficiency of our testing scheme is also quantitatively demonstrated. The experiment data come from two public datasets: LHI database [28] and PASCAL VOC 2008.

LHI database: We select images from four object categories with large intra-class variance, the tiger, giraffe, chick and seagull. In the LHI database, each image includes only one object. For each category, we have 165 images and randomly split them into two sets: 105 images for extracting proto patches and training object model, and 60 images for testing. For the training images, the objects are roughly annotated using an interactive segmentation algorithm. We scale object sizes into 150×150 for proto patch extracting. The patch size is in a range, $(17\times17)-(23\times23)$, and the aspect ratio is not limited. For each category, we obtain more than 2×10^4 proto patches. In the testing stage, for efficiency consideration, the size proposed detecting windows is limited and sampled in a range, $(60\times60)-(180\times180)$. The efficiency is also comparative. Our system is implemented in the Matlab environment with a common desktop PC, and has large room for

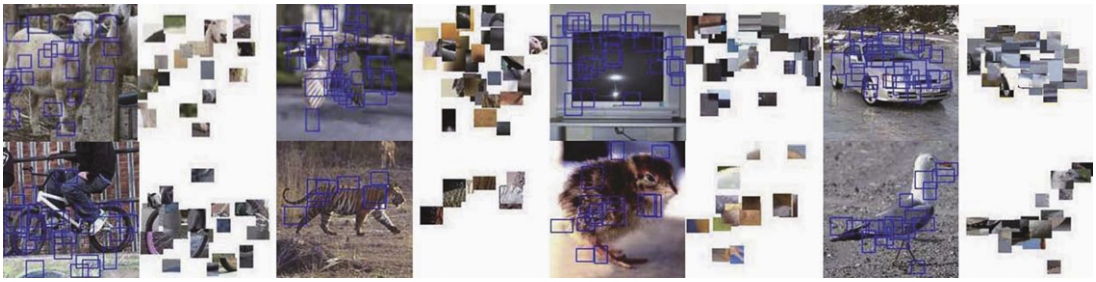

Fig. 10. A few representative results of object localization with our approach. The matched patches from models are also present for each result. The data are from LHI database and PASCAL VOC 2008.

Table 1
Detection performance (AP) compared to the state-of-the-art on the PASCAL VOC 2008 dataset. The detection performance is evaluated by AP, average precision over the entire range of recall. The results in three bottom rows are from our system.

Methods	Bird	Car	Sheep	Tv	Bike	Bottle
CASIA-Det	9.8	17.6	2.8	14.6	14.6	6.3
Jena	0.3	1.3	0.4	13.7	1.4	0.1
MPI-struct	10.1	10.6	9.3	1.5	8.0	0.1
Oxford	–	29.1	–	–	24.6	–
UoCTTIUCI	11.3	32.0	16.1	**37.1**	**42.0**	28.2
XRCE-Det	1.4	4.0	6.1	6.8	10.5	0.0
Harzallah [13]	10.7	36.6	19.4	35.6	33.8	23.3
Combined	**12.6**	**37.6**	**22.9**	33.0	25.6	**29.6**
Explicit	10.3	31.5	14.6	30.6	23.2	28.6
Implicit	9.6	19.8	16.9	13.1	11.4	5.9

improvement. It takes around 5−6 h for training a model. The average running time for a 450×300 testing image is around 20−40 s.

For quantitative comparison, three discriminative methods, Cascaded Adaboost [24], SVM with HOG features [2], and Probabilistic Boosting Tree [22] are implemented. These three methods use 105 images for training and 60 for testing. The precision–recall curves are shown in Fig. 9, and we also present the number of selected features (ε-balls) for each category. A few example images and results are provided in Fig. 10, as well as the matched image patches.

PASCAL VOC 2008: The detection dataset from PASCAL VOC 2008 is more challenging due to various object poses and occlusions, compared to the LHI database, and each image may include more than one object instance. In the PASCAL VOC 2008, the goal of object detection is to predict the bounding boxes of objects,[1] and the training and testing dataset for each object category have been provided beforehand. The data statistics of this database can be found online,[2] and we select six object categories to evaluate our method. We extract proto image patches and learning object model in the training set and perform detecting in the testing set. The experiment setting and details are the same as in the LHI database. Table 1 summarizes the results of our approach, together with several state-of-art systems, where we use the average precision (AP) over the entire range of recall to evaluate the performance. Our method achieves very good performance on five object categories. Result on category bike is of unsatisfactory. The main reason is that the bikes in this dataset have large variance in pose (viewing) which might not be very suitable with our view-based object model.

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/
[2] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/dbstats.html

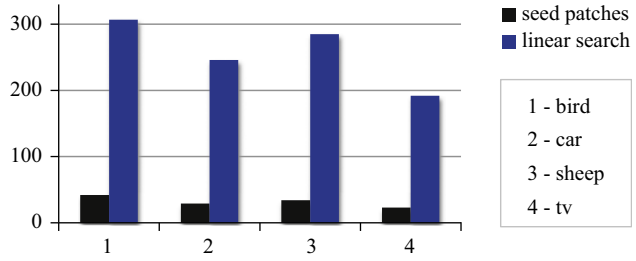

Fig. 11. The efficiency benefit of the seed patches. The vertical and horizontal axes represent, respectively, the time consumption (in seconds) and the object categories. The black (darker) column indicates the efficiency of our approach using the seed patches and the blue column indicates the traditional linear searching method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
The numbers of selected ε-balls over categories as well as the number of activated ε-balls when localizing objects. The top six categories are from PASCAL dataset and the bottom four categories are from LHI database.

Bird	Car	Sheep	Tv	Bicycle	Bottle
1629	1470	1200	760	1307	573
20–30	20–38	17–23	12–28	18–28	10–22
–	Tiger	Giraffe	Chick	Seagull	–
–	973	893	702	1025	–
–	10–20	11–24	11–20	15–25	–

To reveal the recognition power of two types of manifolds, we also show the performance using either explicit or implicit ε-balls. The empirical results are very reasonable, as the explicit ε-balls well capture structural information and the implicit ε-balls are suitable for textural objects, and the combination of them achieve the state-of-the-arts performance.

To quantitatively evaluate the efficiency of our testing procedure compared to the traditional sliding windows approach, we show the timing cost on detecting four categories in Fig. 11.

The numbers of selected manifolds (ε-balls) of object models and the numbers of activated ε-balls when localizing objects are summarized in Table 2. Intuitively, the numbers imply the appearance variances of object categories. In addition, we plot the weights of top 40 explicit and implicit manifolds for four categories in Fig. 12. It reveals the distributions of explicit and implicit ε-balls over object categories as well as the complementarity of them.

6. Summary

This paper proposes an approach for learning sparse object models from two types of manifolds spanned from massive image

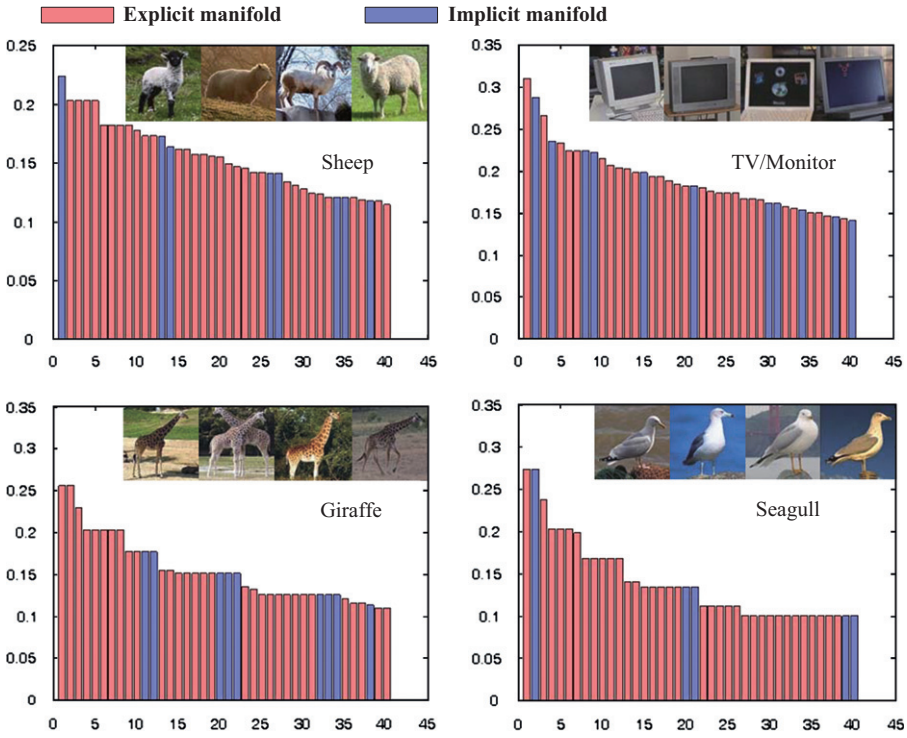

Fig. 12. The top 40 informative manifolds for four categories. In each cell, the horizontal and vertical axes represent the feature index and the corresponding feature weight, respectively. The explicit and implicit manifolds are shown in different colors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

patches. A generative learning algorithm is proposed to pursue object model stepwise by maximizing information gain. By defining and calculating feature correlations, we can directly update the feature weights without sampling distributions or re-weighting data. The empirical evidence shows this method works well with the task of object detection and localization from images, although the approach of updating feature weights with correlations is an approximation and not mathematically tight. We will explore tighter approximation in future work.

Acknowledgment

This work was supported by National High Technology Research and Development Program of China (under Grant no. 2008AA01Z126), National Natural Science Foundation of China (under Grant nos. 60970156 and 60933006), Fundamental Research Funds for the Central Universities (Grant nos. 2010620003162041 and 2011620003161038), and SYSU-Sugon high performance computing typical application projects (Grant no. 62000-1132001). This work was also partially funded by Science and Technology Planning Project of Guangdong Province, China (Grant no. 2011B040300029). The authors would like to thank Professor Song-Chun Zhu for extensive discussions.

Appendix A

Proof of Eq. (19). Since $f_t^{on}=E_f[h_t]$, $p_{t-1}^{on}=E_{p_{t-1}}[h_t]$ and $h_t=\mathbf{1}(d^{r_t}(J_t,I_i)\le\varepsilon_t)$, we derive that

$$\begin{aligned}\mathbf{Z}_t &= \sum_{\forall \mathbf{O}\in \mathbf{S}^{Train}} p_{t-1}\exp(\lambda_t h_t)=\sum_{d^{r_t}} p_{t-1}\exp(\lambda_t h_t)\\ &= \sum_{d^{r_t}\ge\varepsilon_t} p_{t-1}\exp(\lambda_t h_t)+\sum_{d^{r_t}<\varepsilon_t} p_{t-1}\exp(\lambda_t h_t)\\ &= \sum_{d^{r_t}\ge\varepsilon_t} p_{t-1}\exp(\lambda_t)+\sum_{d^{r_t}<\varepsilon_t} p_{t-1}=e^{\lambda_t}p_{t-1}^{on}+1-p_{t-1}^{on}\end{aligned} \tag{A.1}$$

and

$$\begin{aligned}E_{p_{t-1}}[h_t] &= \sum_{\forall \mathbf{O}\in \mathbf{S}^{Train}} p_{t-1}h_t=\sum_{\forall \mathbf{O}\in \mathbf{S}^{Train}} \frac{1}{\mathbf{Z}_t}p_{t-1}\exp(\lambda_t h_t)h_t\\ &= \frac{1}{\mathbf{Z}_t}\sum_{d^{r_t}} p_{t-1}\exp(\lambda_t h_t)h_t\\ &= \frac{1}{\mathbf{Z}_t}\sum_{d^{r_t}\ge\varepsilon_t} p_{t-1}\exp(\lambda_t h_t)h_t+\frac{1}{\mathbf{Z}_t}\sum_{d^{r_t}<\varepsilon_t} p_{t-1}\exp(\lambda_t h_t)h_t\\ &= \frac{1}{\mathbf{Z}_t}e^{\lambda_t}p_{t-1}^{on}.\end{aligned} \tag{A.2}$$

Letting $E_{p_{t-1}}[h_t]=E_f[h_t]=f_t^{on}$ combine with Eq. (A.1), we obtain

$$\lambda_t=\log\frac{f_t^{on}(1-p_{t-1}^{on})}{p_{t-1}^{on}(1-f_t^{on})},\quad \mathbf{Z}_t=e^{\lambda_t}p_{t-1}^{on}+1-p_{t-1}^{on}. \tag{A.3}$$

References

[1] S. Agarwal, D. Roth, Learning a sparse representation for object detection, Proceedings of European Conference on Computer Vision, vol. 4, 2002, pp. 113–130.
[2] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2005.
[3] S. Della, V.D. Pietra, J. Lafferty, Inducing features of random fields, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (4) (1997) 380–393.
[4] B. Epshtein, I. Lifshitz, S. Ullman, Image interpretation by a single bottom-up top-down cycle, Proceedings of the National Academy of Sciences of the United States of America, vol. 105, no. 38, 2008, pp. 298–303.
[5] 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, 2005.
[6] L. Lin, X. Liu, S.-C. Zhu, Layered graph matching with composite cluster sampling, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (8) (2010) 1426–1442.
[7] Y. Freund, R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computation and System Science 55 (1) (1997) 119–139.

[8] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2003.
[9] V. Ferrari, F. Jurie, C. Schmid, Accurate object detection with deformable shape models learnt from images, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[10] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Technique Report, Stanford Univ., 1998.
[11] C. Gu, J.J. Lim, P. Arbelaez, J. Malik, Recognition using regions, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009.
[12] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.
[13] H. Harzallah, F. Jurie, C. Schmid, Combining efficient object localization and image classification, in: Proceedings of IEEE International Conference on Computer Vision, 2009.
[14] B. Leibe, A. Leonardis, B. Schiele, Combined object categorization and segmentation with an implicit shape model, in: ECCV'04 Workshop on Statistical Learning in Computer Vision, Prague, 2004.
[15] T.S. Lee, Image representation using 2d gabor wavelets, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (10) (1996) 959–971.
[16] L. Lin, T. Wu, J. Porway, Z. Xu, A stochastic graph grammar for compositional object representation and recognition, Pattern Recognition 42 (7) (2009) 1297–1307.
[17] M. Muja, D.G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, in: Proceedings of International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2009.
[18] B.A. Olshausen, D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381 (1996) 607–609.
[19] K. Sande, T. Gevers, C. Snoek, Evaluation of color descriptors for object and scene recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[20] K. Shi, S.C. Zhu, Mapping the ensemble of natural image patches by explicit and implicit manifolds, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[21] S. Todorovic, N. Ahuja, Unsupervised category modeling, recognition, and segmentation in images, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (12) (2008) 2158–2174.
[22] Z. Tu, Probabilistic boosting tree: learning discriminative models for classification, recognition, and clustering, in: Proceedings of IEEE International Conference on Computer Vision, 2005.
[23] Z. Tu, Learning generative models via discriminative approaches, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[24] P. Viola, M. Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2004) 137–154.
[25] S. Winder, M. Brown, Learning local image descriptors, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[26] Y.N. Wu, Z. Si, H. Gong, S.C. Zhu, Learning Active Basis model for object detection and recognition, International Journal of Computer Vision 90 (2) (2010) 198–235.
[27] M. Welling, G. Hinton, S. Osindero, Learning sparse topographic representations with products of student-t distributions, in: Proceedings of Neural Information Processing Systems, 2002.
[28] B. Yao, X. Yang, L. Lin, M.W. Lee, S.C. Zhu, I2T: image parsing to text description, Proceedings of IEEE, vol. 98, no. 8, 2010, pp. 1485–1508.
[29] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: a comprehensive study, International Journal of Computer Vision 73 (2) (2007) 213–238.
[30] S.C. Zhu, Y.N. Wu, D. Mumford, Minimax entropy principle and its applications to texture modeling, Neural Computation 9 (8) (1997) 1627–1660.

Liang Lin received the B.S. and Ph.D. degrees from Beijing Institute of Technology (BIT), Beijing, China in 1999 and 2008, respectively. He studied in the Department of Statistics, University of California, Los Angeles (UCLA), as a visiting Ph.D. student during 2006–2007. He was a Postdoctoral Research Fellow at the Center for Image and Vision Science (CIVS), UCLA, and a Research Scientist at Lotus Hill Institute, China, during 2007–2009. Currently, he is an Associate Professor at Sun Yat-Sen University (SYSU), Guangzhou, China. His research interests include but are not limited to object recognition, graph and shape matching, image parsing, and visual tracking.

Ping Luo received his M.E. and B.E. degrees in School of Software in 2008 and 2010, respectively, from Sun Yat-Sen University, Guangzhou, China. His research interests include computer graphics, computer vision, and machine learning.

Xiaowu Chen is a Professor in State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University. He received his Ph.D. degree at Beihang University in 2001, and was a post-doc research fellow at University of Toronto. His research interests include computer graphics, computer vision, virtual reality and augmented reality.

Kun Zeng received his Ph.D degree from National Laboratory of Pattern Recognition Institute of Automation, Chinese Academy of Sciences in 2008. He is now an Assistant Professor at Sun Yat-Sen University (SYSU), Guangzhou, China. His research interests are in computer vision, multimedia and non-photorealistic rendering.