Noname manuscript No.
(will be inserted by the editor)

Stel Component Analysis: joint segmentation, modeling and recognition of objects classes

Nebojsa Jojic[1] · **Alessandro Perina**[1,2] · **Marco Cristani**[2,3] · **Vittorio Murino**[2,3]

the date of receipt and acceptance should be inserted later

Abstract Models that captures the common structure of an object class have appeared few years ago in the literature [30,64]; they are often referred as "stel models". Their main characteristic is to segment objects in clear, often semantic, parts as a consequence of the modeling constraint which forces the regions belonging to a single segment to have a tight distribution over local measurements, such as color or texture. This self-similarity within a region in a single image is typical of many meaningful image parts, even when across different images of similar objects, the corresponding parts may not have similar local measurements. Moreover, the segmentation itself is expected to be consistent within a class, although still flexible. These models have been applied mostly to segmentation scenarios.

In this paper, we extent those ideas 1) proposing to capture correlations that exist in structural elements of an image class due to global effects, 2) exploiting the segmentations to capture feature co-occurrences and 3) allowing the use of multiple, eventually sparse, observation of different nature. In this way we obtain richer models more suitable to recognition tasks.

We accomplish these requirements using a novel approach we dubbed *stel component analysis*. Experimental results show the flexibility of the model as it can deal successfully with image/video segmentation and object recognition where, in particular, it can be used as an alternative of, or in conjunction with, bag-of-features and related classifiers, where stel inference provides a meaningful spatial partition of features.

Keywords Object class modeling, Statistical learning, Bag of Words and beyond, Segmentation, Recognition

1 Introduction

Understanding the semantics of an image is one of the most challenging problems in computer vision. A useful object recognition system needs to recognize thousands of objects and learn about new object classes from a relatively small number of examples. It is also essential that learning does not require any human involvement in annotating training images. Image models thus need to encode invariances that would hold across all images, as this reduces the amount of training data.

One of the greater problems in recognizing object classes is that there can be significant changes in appearance from one object instance to another. Real-world imaging conditions and the characteristics of the sensing devices often result in strong fluctuations of pixel intensities, even for similar objects. In addition, the changes in appearance may be due to the variation in material properties among instances of an object. For all these reasons, images are often represented by indices referring to a set of possible local features, more or less invariant to illumination conditions, e.g. SIFT [39], Harris affine regions [26], Maximally Stable Extremal Regions [41], and SURF [4] to name a few.

In most of the cases, the spatial configuration of the features is not considered: this is the case of bag of words (BoW) models [22,17]. In such models, each image class has a distinctive distribution of features, sometimes called "palette". This distribution is typically found in most instances of the class, although the image locations in which these features are found may substantially vary (see Fig. 1a for an example). In any case, the distribution over the feature indices describes an entire image class. Examples of such models are the generative Naive Bayes classifier [17], or the generative hierarchical models presented in [22,27].

[1] Microsoft Research, Redmond, WA
[2] University of Verona, Verona, Italy
[3] Italian Institute of Technology, Genova, Italy

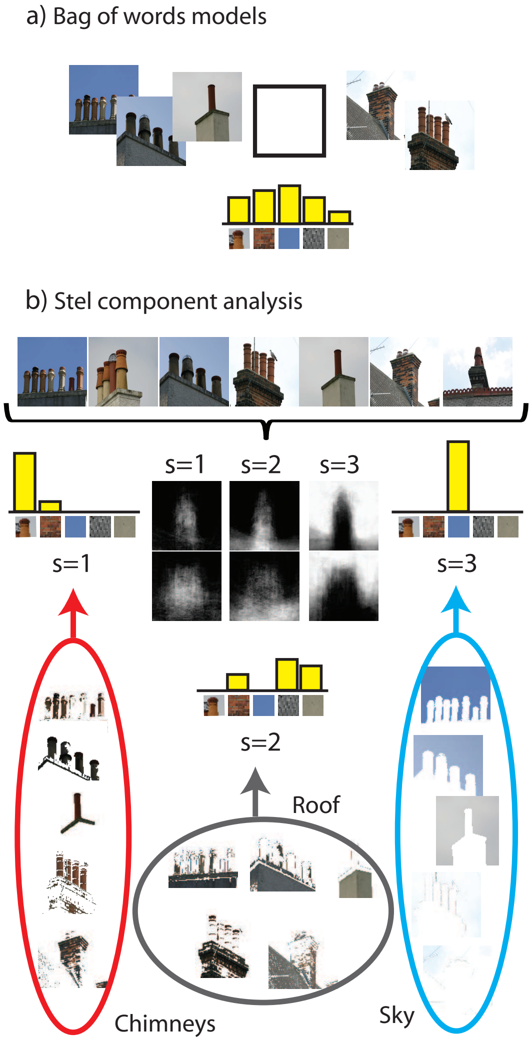

Fig. 1 a) Bag of words models extract features from the whole images disregarding the spatial information; an histogram of expected feature distribution is then calculated pooling together the features from all the samples. Bag of words is equivalent to stel component analysis with S=1. b) The adaptive-learned feature organization provided by Stel Component Analysis.

An improvement over the bag of words models are the index map models [30,64]. Their basic assumption is that within an object class, each image exhibits local intensity patterns that are repeated in nearby image locations in a similar relative geometric layout [2]. In other words, although the intra-class appearances of the objects may vary, the object shape is consistent, and the variability of color/texture within a single instance of an object is limited. The literature refers to this assumption as “self-similarity”. Beside object class modeling [30,64,3], it has also found its way into object recognition in the form of local [53] or global [19] descriptors.

Under this assumption, another approach to invariance to appearance variation can be used: palette invariance through *index map* modeling.

We formally define *index maps* as ordered sets of indices $s_i \in 1,\ldots,S$, linked to spatially distinct areas $i \in 1,\ldots,N$ of images where N is the number of such image areas, e.g., pixels (see Fig.2a.). We define an area of an image with the same assigned index s as a *structure element*.

These indices point to a table of S possible local measurements, referred to as palette. In such model the feature palette is pertinent only to a single instance of a class, while the indexing configuration is assumed to be relevant to the entire class of images.

While the index map models has the advantage over BoW models in terms of capturing the spatial structure of images, it does not deal with 1) the possibility of estimating consistent palettes across images of the class (e.g., in a video sequence, subsequent images do have similar palettes), and 2) possible dependencies among the indices s_i.

We address both problems by proposing a new model, named *Stel Component Analysis* (SCA). The model of index variations described above is so enriched by this analysis which, inspired by principle component analysis and other subspace methods, captures correlated variations in discrete indices through a model of blending of several component index maps based on real valued weights y. This is visually described in Fig. 2c, where three components (k=1,2,3) are shown and where the blending of them allows for a better agreement of the observed facial image with the model.

In more complex categories, the proposed model benefits from learning a prior over individual image palettes, which is similar to what is achieved in the BoW models, except that these appearance models can now be part-specific. In Fig. 1b, the SCA model is applied to the more complex category of roof images, where the prior over individual image palettes is represented by different histograms over image features in the three different stels. This example illustrates the advantages of the model presented here over both BoW and index models. Where the image class does indeed have consistent features across its instances, our model, captures this property through a prior over the palettes. But, unlike BoW models, it keeps separated the features typical of different image parts, and the more consistent segmentation for modeling the image class is inferred jointly with these feature distribution through learning of an SCA model.

In summary, this paper proposes a novel generative model, Stel Component Analysis (SCA), built upon the probabilistic index map [30]. By means of its components, SCA captures correlations of pixels, placing an image in the simplex defined by its component mixing strengths y^t, so allowing a dramatic increase of the modeling capability.

Stel component analysis, and in particular the effect of its components, is tested on several image parsing tasks proving its capabilities and flexible usages. In particular, 1) we

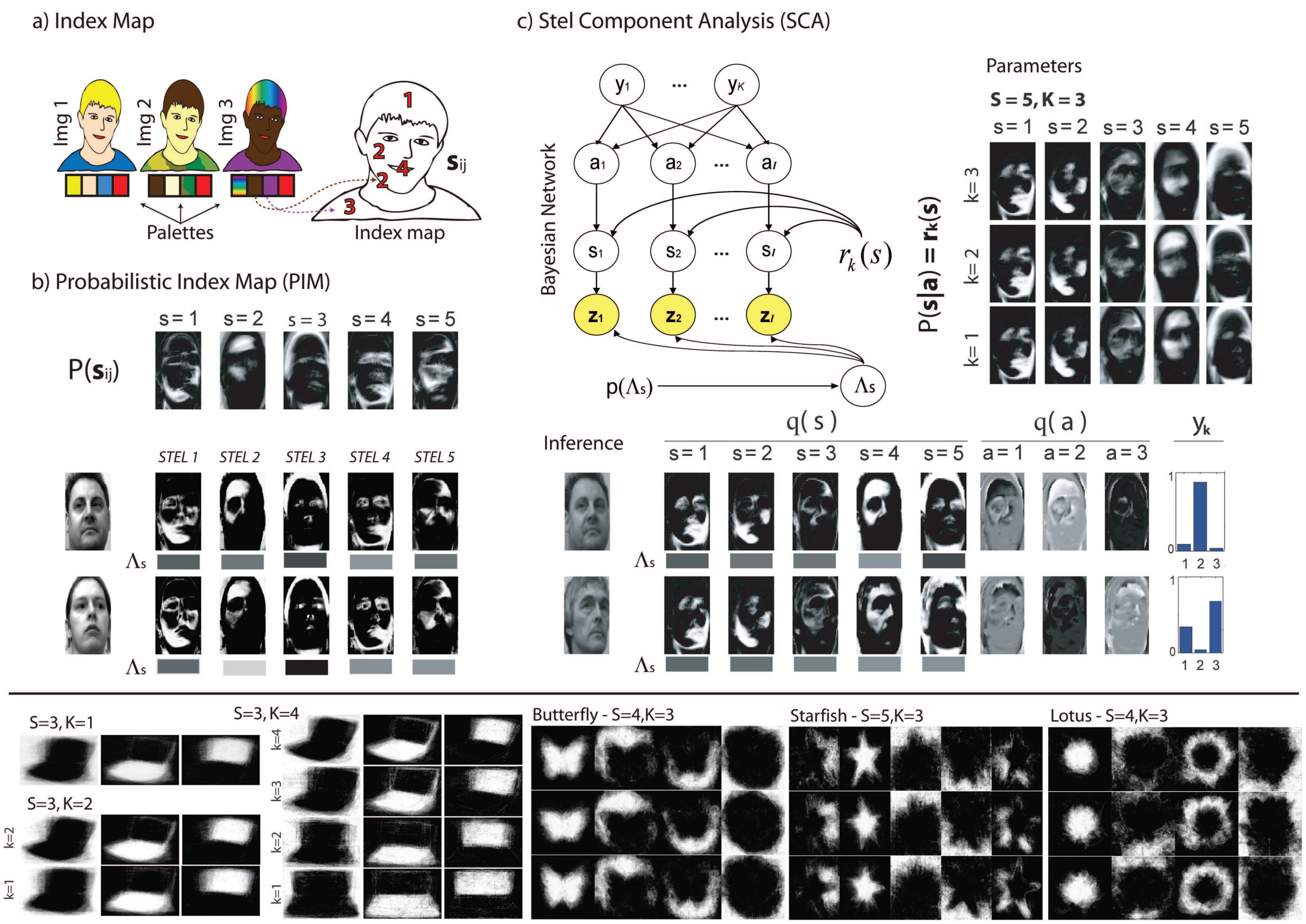

Fig. 2 a) Illustration of an index map. b) Probabilistic index map with a palette of size S=5 – a special case of Stel Component Analysis with K=1. c) Stel component analysis, Bayesian network and inference illustration. SCA with K=3 components allows modeling spatially correlated changes in discrete stels. On the bottom, the inferred stels $q(s)$, the image palettes, as well as the component strengths y_k for three different faces with varying poses. The strengths y_k turn out to be linearly correlated with the pose angle and outperform PCA as pose estimators. Site-specific blending variables a allow nonuniform mixing of components. Their posterior distributions $q(a)$ are also shown.

analyze object class modeling power of SCA, 2) we use the image partitions to group features into meaningful sets for a better use with discriminative classifiers, 3) we group the stels to perform a foreground extraction or segmentation tasks, 4) we use the stel partitions $q(s^t)$ as feature to capture the global self-similarity of objects.

A previous version of this paper has appeared in [29], this paper extends it in several aspects. We extend the experimental section using Caltech 101 instead of an handmade dataset. We compared SCA on image segmentation with [64, 1, 51] and we evaluated SCA in independent one-vs-all classification tasks.

As technical novelty, we introduce here the derivation for the conjugate priors on the three kind of palette considered and we added a Dirichlet prior on the components y_k, an overall prior on the stels useful to automatically select the number of stels, and a Markov Random Field prior to prefer that near pixels had the same label. The benefits of these priors will be discussed in the following. Moreover, as further novelty, we introduce the Stel Kernel, which represents the missing piece of research between the “S-Bags of words” paradigm introduced in [29], and our following work [49].

The remainder of the paper is organized as follows. Section 2 presents the related work. The details of the architecture of the method and the learning algorithm are presented in Sec. 3. Subsequently, in Sec.4, experimental results are illustrated. Finally, a critical discussion concludes the paper in Sec. 5.

2 Related Work

Previous probabilistic approaches to learning and understanding object classes can be divided based on how they deal with spatial arrangement of local features.

In recent years, bag of words models [6], have been successfully used for vision tasks [22,9,17,63,32,16]. These models are particularly attractive due to the computational efficiency and simplicity achieved by ignoring spatial relationships of the image patches or object parts. After extracting the features from images, the features are clustered and a discrete "codewords" is assigned to each feature descriptor. An image is then described by a histogram over the codebook entries.
Topic models [6,28], maybe the most famous example among BoW approaches, assign a topic to each codeword based on their co-occurrence in images. A representation that lies between the template and the BoW is the epitome [32,33,44]. It is learned by compiling patches drawn from input images into a condensed image model. The balance between visual resemblance and generalization of image video can be adjusted by the sizes of epitome and patch. Subsequently, [14, 13] present improved epitome models which combined the patch appearance information with some spatial information.
Finally, in [48] the authors present the Counting Grids. The paper shows how much of the variability in vision datasets is better modeled in terms of multidimensional thematic shifts, rather than outright topic mixing proper of topic models [6]. Similarly to epitomes, the features are arranged in a window, which is then embedded in some hypothetical bigger scene; certain features are dropped and others added as a consequence of movement in this scene.

In contrast other models explicitly encode spatial information, often at a considerable computational cost. Among these methods, we can identify methods that keep spatial relationships between feature locations and method that spatially organize the features.

In [55] is presented a strategy aimed at forming vocabularies from pairs of nearby features called "doublets" or "bigrams". This method, beside taking co-occurrences into account, captures some geometric invariance, but it is too demanding since many doublets have to be estimated. The approach proposed in [37] learns a codebook of local appearances that contains information about which local structure may appear on objects of a particular class. It also specifies where a particular codeword may occur on the object. Despite being invariant to rigid transformations, it needs additional supervision through human-supplied object positions and ground truth segmentations.
Generative part-based models [50,62,52,56,58,57], are very nice conceptually and learnable from unsegmented images, but they require a computationally demanding combinatorial hypothesis search. The most famous example is the constellation model [62] which attempts to represent an object class by a set of different parts under mutual geometric constraints. The same idea was later used to represent and learn generic 3D object categories [52,56,58].
In [57], it is argued that multi-object recognition systems should be based on models which consider the relationships between different object categories. The approach builds upon the constellation model and it demonstrates that objects classes can be described in terms of shared parts without increasing the size (hence, the complexity) of the representation.

It is also possible to model *where* the features are present in terms of absolute image locations [12,40,64,35,60,47, 30,31]. In [12] propose a generative model, called Spatial-LTM, which assigns topics to an images by incorporating meaningful spatial coherency among the patches. The idea is that pixels should share the same latent topic assignment if they are in a neighboring region with similar appearance.
The well-known Spatial Pyramid Kernel [35], extends the BoW paradigm providing a locally orderless representation at several levels of resolution. This is obtained by grouping the features following a hierarchical fixed partition of the images.
[40] describes a method to infer the object-background segmentation of test images, starting from labeled training images (i.e., the object extraction mask is known). It classify objects with SVM by weighting the features according to the segmentation.
Index map models [30,64,31] also fall in this category. These models aim at capturing the consistent spatial layout of the classes (see Fig.2, bottom). The probabilistic index map (PIM) approach [30,31] relaxes the hard assumption of the index maps that indices that model the same location across the images should be equal, by allowing them instead to following the same distribution. For an example see Fig.2a (index map) -
reffig:illb (probabilistic index map). In this model, each image location i is associated with a distribution over indices $p(s_i = s)$ and each image has its own index map $q(s_i^t)$[1] governed by $p(s_i = s)$. These indices point to a table of S possible local measurements, referred to as palette.
The PIM model allows for complete freedom of choosing in image-specific color palettes, and thus full palette-invariance of the indexing model. Fig. 2b illustrates a PIM model and it emphasizes the difference between the prior over stels $p(s)$ and the inferred indices for individual images $q(s_i^t)$, which depend on both the prior and the self-similarity properties of an individual image.
In [64], the authors build their model, called LOCUS, on the same palette invariance assumption of PIM [30], proposing the use of a more expressive color distribution in the entries of the palette, exploiting both appearance and shape (edges), and learning complex priors that capture appearance, edge

[1] This modeling is due to the variational inference that has been employed, see Sec. 2.

and color distribution of a class.
Another example of index map model is [45], where a Latent Dirichlet Allocation model is learned in each segment.

The ideas behind Stel Component Analysis which we present in this paper extend the literature on object class probabilistic models in several respects. First, SCA deals with multiple and sparse features generalizing [64, 30] which only relies on dense discrete edge/color and continuous appearance, respectively. We also extended the basic appearance models (palettes) of PIM, presenting much more complex palettes that can capture segments with a multi-modal feature distribution. In contrast to [30], we cope with the full color invariance, learning the appropriate conjugate priors, that serve to capture interesting co-occurrence of features in the stels. For example, see in Fig. 1b how the third stel for the chimney class models the sky, and many (all) images will present a "bluish" palette in $\Lambda_{s=3}$.
Second, we consider the index maps more broadly than several previous techniques for modeling spatial correlations in index map-like approaches [54, 22, 35, 30, 64]. Previous models have mostly been limited by three basic characteristics. In one class of approaches, a Markov random field is used to define several potentials governing spatial local correlations among few image features (e.g., [54]). In the second class of methods, a spatial distribution in the image is given for each feature, and this imposes probabilities of seeing a particular index in a particular spot [22, 35]. In the third one, site-specific distribution over indices, assuming independence in index variation across image locations, are enriched with transformation and deformation models or are used within a mixture model [30, 64]. Stel component analysis is more flexible than these models, as it captures higher-order statistics than Markov random fields, and can adapt, through its components to a variety of image deformations without parameterizing them ahead of time, as in [30, 64].

3 Stel component analysis

3.1 The generative model

To make image models invariant to changes in local measurements, while sensitive to changes in image structure, a measurement $\mathbf{z}_{\mathbf{i}}^t$ (e.g. pixel intensity or feature) at location $\mathbf{i} = (i, j)$ in the t-th image of a certain class (object category or a video clip, for instance), is considered to depend on a hidden index $s_{\mathbf{i}}^t \in \{1, ..., S\}$, Fig. 2B:

$$p(\mathbf{z}_{\mathbf{i}}^t | s_{\mathbf{i}}^t = s) = p(\mathbf{z}_{\mathbf{i}}^t | \Lambda_s^t) \qquad (1)$$

The s-th structure element (stel) indicates pixels $i | s_{\mathbf{i}}^t = s$ which follow a shared distribution over local measurements (palette) with parameters Λ_s^t. In the example in Fig 2C, each palette entry defines a single Gaussian model with its mean and variance over intensity levels, $\Lambda_s^t = \{\mu_s^t, \phi_s^t\}$, as was previously done in [30]. The inferred means μ_s^t of such palette entries for several facial images are shown in the lower part of each stel. Palettes Λ_s^t are considered hidden variables in the model, each defining a limited diversity of local measurements within each image. However, the stels are generated from a single distribution shared among all images of the class $p(\{s_{\mathbf{i}}\})$. Fig.2B shows the estimated distribution of the form:

$$p(\{s_{\mathbf{i}}\}) = \prod_i p(s_{\mathbf{i}}). \qquad (2)$$

To visualize the class stel distribution, in Fig. 2B we show the estimated $p(s_{\mathbf{i}} = s)$ for $s \in \{1, 2, 3, 4, 5\}$ as five images where in the s-th image each location indicates the probability of that pixel being mapped to index s for the face pose dataset. The image locations that tend to have similar colors within each individual single image (but not across images) are grouped into stels.

The pixel intensity $z_{\mathbf{i}}$ tends to be uniform within a stel in a single image, and can be inconsistent across different images, as they may be darker or brighter, for instance. However, the stels are relatively consistent over facial images and they represent interesting image structure beyond intensity levels. For example, stel s=2 captures parts of forehead and cheek that have similar surface normals, while the eyebrows and the hair are grouped into stel s=3, despite the variability in hair color across images.
The PIM model assumes independence of distributions over indices across different image locations, ignoring the correlations in index variation which often arise even from simple structural variation in the image, such as variation in face proportions, or out of plane head rotation. In models of variation in real valued, rather than discrete, arrays, such correlations are often captured using a subspace model, e.g. PCA, which achieves this through a linear combination of several components. Since we are concerned here with modeling a distribution over discrete indices, we develop a discrete analogue to eigen images, similar in spirit to multinomial PCA [11], non-negative matrix factorization [36], latent Dirichlet analysis models [6, 36] or the generalization of PCA presented in [15], but with some important distinctions. Our model is a full probability model of the observed data, meant to capture spatial structure, and thus it is designed for *ordered* index sets, and it also allows *spatially nonuniform* mixing of the components.
In stel component analysis the components r_k, $k \in \{1, ..., K\}$ are of the same form as (2), that is

$$r_k(\{s_{\mathbf{i}}\}) = \prod_i r_k(s_{\mathbf{i}}) \qquad (3)$$

An example of learned components is shown in Fig.2C. These components are combined to define the distribution $p(\{s^t\})$ using component strengths $y_k^t \in [0..1]$, so that $\sum_k y_k^t = 1$. The components strengths are real valued hidden variables for image t, rather than component priors in a mixture model as in a mixture of probabilistic index maps. Each image is thus defined by a hidden point in the simplex defined by $\sum_k y_k^t = 1$, and this point will rarely fall in a vertex as illustrated in the two inference examples of Fig. 2C, whereas in a mixture model, each image will have a discrete pointer to a single component. In other words, while stel component analysis blends the K components, a mixture of K-PIMs would chooses one of the K components.

To achieve this, as well as to allow spatially nonuniform mixing of components based on real valued component strengths, we add a layer of discrete hidden variables $a_{\mathbf{i}} \in \{1, .., K\}$ which act as mixture component indicators, *but only locally* for their corresponding image locations $\mathbf{i}$ (see the Bayesian network of SCA in Fig. 2C). Hidden component strengths y_k, shared across the pixels of an image, then act as prior probabilities in these local mixture models:

$$p(\{s_{\mathbf{i}}^t\}|\{a_{\mathbf{i}}^t\}) = \prod_{\mathbf{i}} p(s_{\mathbf{i}}^t|a_{\mathbf{i}}^t)$$
$$p(s_{\mathbf{i}}^t|a_{\mathbf{i}}^t = k) = r_k(s_{\mathbf{i}}^t); \quad p(a_{\mathbf{i}}^t = k|y_k^t) = y_k^t. \tag{4}$$

By summing over hidden variables a, a desired mixing of components with real valued weights y_k^t is achieved to form a differently mixed stel distribution for each image. Since different hidden variables $a_{\mathbf{i}}$ can have different values (and thus choose different components r_k in different parts of the image), the mixing is spatially nonuniform, and each variable y_k influences only the total number of image locations $\mathbf{i}$ that choose $r_k(s_{\mathbf{i}})$ as the local prior on the index. This allows dramatically more flexible mixing than in PCA models, making object part alignment across images much easier to achieve without global image transformations.

Again, the use of the layer of hidden variables $a_{\mathbf{i}}$ makes the model different from a simple mixture of site-specific models. Index probabilities from different components are blended differently in different parts of the image, which simple mixture models do not allow. This gives the model more flexibility in parsing images, and, as desired, allows for variable mixing of the components for different images to model smooth geometric changes (see Fig. 2C).

The joint distribution over all observed variables $\mathbf{z} = \{\mathbf{z}_{\mathbf{i}}^t\}$, and hidden variables $\mathbf{h}$ =$\{\{y_k^t\}, \{a_{\mathbf{i}}^t, s_{\mathbf{i}}^t\}, \{\Lambda_s^t\}\}$ is

$$p(\mathbf{z}, \mathbf{h}) = \prod_t \Bigg(p(\{y_k^t\}_{k=1}^K) p(\{\Lambda_s^t\}) \prod_i p(z_{\mathbf{i}}^t|s_{\mathbf{i}}^t, \{\Lambda_s^t\})$$
$$\prod_k (y_k^t r_k(s_{\mathbf{i}}^t))^{[a_{\mathbf{i}}^t = k])} \Bigg) \tag{5}$$

where $[\cdot]$ is the indicator function. The prior on y_k can be kept flat (as in our experiments), or learned in a Dirichlet form.

3.2 Inference

Following the variational inference recipe, we 1) introduce a tunable distribution $q(\mathbf{h})$ over the hidden variables/parameters, 2) define as a bound on the log likelihood $-\log p(\mathbf{z})$, the negative free energy $-F = \sum_{\mathbf{h}} q(\mathbf{h}) \log \frac{q(\mathbf{h})}{p(\mathbf{z},\mathbf{h})}$, and 3) pursue the strategy of minimizing this free energy iteratively. We used the simplest of the algorithms from this family, where the approximate posterior distribution $q(\mathbf{h})$ is fully factorized

$$q(\mathbf{h}) = \prod_k q(r_k) \prod_{i,t} q(a_i^t) q(s_i^t) \prod_t q(y_k^t) q(\{\Lambda_s^t\}) \tag{6}$$

with $q(r_k)$, $q(y_k^t)$ and $q(\Lambda_s^t)$ being Dirac functions centered at the optimal values (or vectors) $\hat{r}_k$, $\hat{y}_t^k$, $\{\Lambda_s^t\}$. As a result, the (approximate) inference reduces to minimizing the following free energy,

$$F = \sum_t p(\{\hat{\Lambda}_s^t\}) + \sum_{t,i,s} q(s_i^t = s) \log p(z_{\mathbf{i}}^t|s_{\mathbf{i}}^t, \{\hat{\Lambda}_s^t\}) +$$
$$+ \sum_{t,i,k} q(a_i^t = a) \log \hat{y}_a^t +$$
$$+ \sum_{t,i,a,s} q(a_i^t = a) q(s_i^t = s) \log \hat{r}_k(s_i^t = a), \tag{7}$$

which is reduced by the following steps, also illustrated in Fig. 3a:

1. The palettes for different stels in a single image t are assigned so as to balance the need to agree with the prior $p(\Lambda)$ with the statistics of the local measurements within a (probabilistic) stel in the image:

   $$\hat{\Lambda}_s^t = \arg\max \log p(\{\hat{\Lambda}_s^t\}) + \tag{8}$$
   $$+ \sum_{i,s} q(s_i^t = s) \log p(z_i^t|s_i^t = s, \{\hat{\Lambda}_s^t\}).$$

   More details will be given below, when we will introduce complex palettes.

2. Stel segmentation of image t is based on the similarity of observed local measurements to what is expected in a particular stel s according to the estimated palette $\hat{\Lambda}_s^t$ in this particular image, as well as the expected stel assignment based on mixed components $r_k(s)$. These mixed components are mixed differently in different parts of the image, and the mixing is defined by $q(a)$:

   $$q(s_i^t = s) \propto p(z_i^t|s_i^t, \{\hat{\Lambda}_s^t\}) e^{\sum_a q(a_i^t = a) \hat{r}_a(s_i^t = s)}. \tag{9}$$

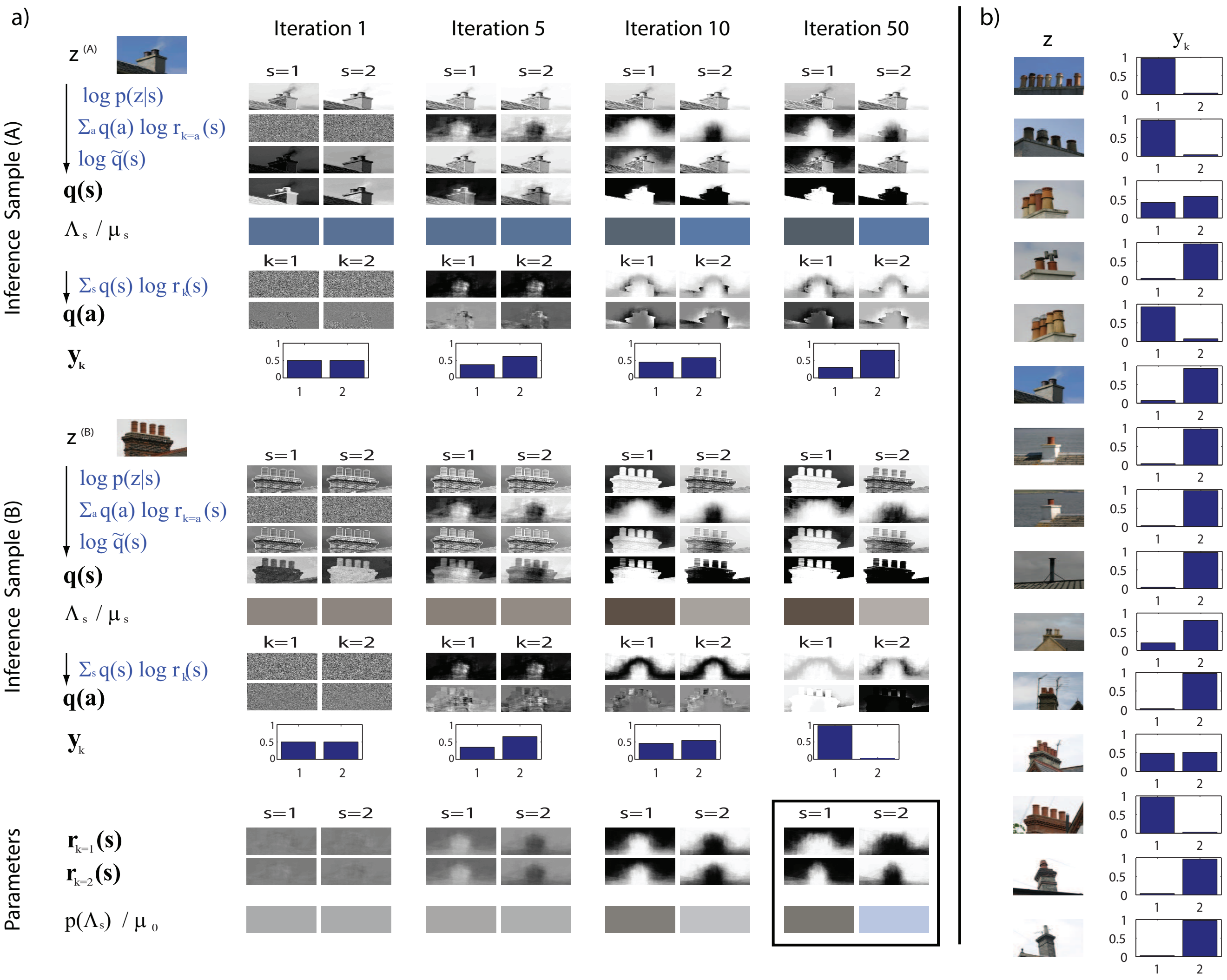

Fig. 3 a) Illustration of the learning process across the iterations; we show the posterior distributions ($q(s)$, $q(a)$, y_k and Λ_s, we used black text), the "addends" that compose them (blue text) and the model parameters. b) Several images (z) and their components strength (y_k). The final parameters are boxed in the right-bottom most corner of part "a)".

To avoid numerical underflow, it is useful compute $\log \tilde{q}(s)$ in the log-domain and then normalize. This reduces to

$$\log \tilde{q}(s_i^t = s) = \log p(z_i^t | s_i^t, \{\hat{\Lambda}_s^t\}) + \sum_a q(a_i^t = a)\hat{r}_a(s_i^t = s). \tag{10}$$

In Fig. 3a, we showed all the steps necessary to get the stel segmentation $q(s)$ for a couple of images.

3. The spatially nonuniform component mixing, defined by $q(a)$, is updated so as to balance the agreement with the overall strength y_a^t of the component a in the particular image t, with the agreement of the stel assignment with the stel component r_a:

$$q(a_i^t = a) \propto y_a^t e^{\sum_s q(s_i^t = s) \log r_a(s_i^t = s)}. \tag{11}$$

As for the previous update rule, to keep numerical precision, one should work in the log-domain.

4. The stel component strengths y_a are assigned proportionally to their use in the image:

$$y_a^t \propto \sum_i q(a_i^t = a). \tag{12}$$

5. The stel components r_a are updated to reflect the assignment statistics over all images:

$$r_a(s) \propto \sum_t q(a_i^t = a) q(s_i^t = s). \tag{13}$$

Iterating these updates results in learning the model parameters, as well as inferring the hidden variables, e.g., the consistent parsing of images from a class into its stels. With consistent we mean that the same stel represent the same semantic concept in all the images; for example stel 3 in Fig. 2b represent the hair, stel 3 in Fig. 1 always represent the sky, etc.

Fig. 3a, also shows the behavior of the learning algorithm; palettes and segmentations ($q(s)$) are immediately learnt, and few iterations (See posterior at iteration 10 in Fig. 3) they converge. After that the model separates and learns the K components to better fit with the data; during this time, the segmentations are slightly refined to capture finer details, until convergence which usually occurs in 60-100 iterations. Note in fact, how in Fig. 3 the 2 components are nearly identical after 10 iterations, while they look very different at convergence.

The energy minimization procedure is illustrated with Algorithm 1. To speed-up the inference process, we performed $E_{int} = 2, 3$ internal iterations within the E-step to gain confidence about $q(s)$ and the palettes. If palette priors are learnt, re-initializing every 10-20 iterations the palettes, possibly using the prior as initial guess, also speeds-up the learning procedure.
To avoid local minima, the palette variances/probabilities and the prior and posterior probabilities on the random variables s, a and y_k were not allowed to drop below a small constant $(10^-2/10^-100)$.

Algorithm 1: Free Energy Minimization

```
Input: Images of a class of objects, $z^t$
Output: Class description $r_k(s)$, $p(\Lambda)$
while Convergence do
    % E-Step ;
    foreach Sample $t = 1 \dots T$ do
        foreach $n = 1 \dots E_{int}$ do
            1. Update $q(s)$ (Eq. 24) ;
            2. Update the palette $\Lambda_s$ (Sec. 3.3) ;
        3. Update $q(a)$ (Eq. 22) ;
        4. Update $y_k$ (Eq. 12) ;
    % M-Step ;
    5. Update $r_k(s)$ (Eq. 13) ;
    6. Update the priors (See Secs. 3.3 and 3.4) ;
    7. Compute the Free Energy $F$ (Eq. 7) ;
    8. Check for convergence ;
10. Return $r_k(s)$ and $p(\Lambda)$ ;
```

3.3 Local measurements z^t, palette models $p(z^t|\Lambda_s^t)$ and palette priors $p(\Lambda_s)$

The local measurements z_i may vary depending on the application, and can be scalar or multidimensional, discrete or real valued. To obtain the face model in Fig. 2, as in [30], we assumed that 1) the local measurements are simply the real valued image intensities, 2) the palette model $\Lambda_s = (\mu_s, \phi_s)$ is Gaussian, $p(z_i^t|s_i^t = s, \{\hat{\Lambda}_s^t\}) = \mathcal{N}(z_i^t; \mu_s, \phi_s)$, 3) the prior on the palette Λ_s is kept flat. The palette update is therefore based on sufficient statistics over intensities within stels in individual images, that is

$$\mu_s^t \propto \sum_i q(s_i = s) \cdot z_i^t \tag{14}$$

$$\phi_s^t \propto \sum_i q(s_i = s) \cdot (z_i^t - \mu_s^t) \cdot (z_i^t - \mu_s^t)^T$$

Alternative local measurements include color, disparity, flow, SIFT or some other local features [42]. As more expressive palette models, we introduce here the histogram representation for discrete local measurements, and the mixture of Gaussians for the real valued measurements.

For the case of discrete measurements, we define the palette as a histogram over C possible observations $\{\zeta_j\}$, $j \in \{1, ..., C\}$. The observation distribution is multinomial with parameters $u_j = p(z = \zeta_j)$, and the palettes $\Lambda_s = \{u_{s,j}\}$ are defined by using these probabilities. With a flat prior on Λ, the equation (14) reduces to

$$u_{s,j}^t \propto \sum_i q(s_i = s)[z_i^t = \zeta_j] \tag{15}$$

When measurements consist of different modalities, which are generally uncorrelated at the local level when viewed without regard for the high level context, they are combined by setting

$$p(z_i|s) = \prod_m p(z_{m,i}|\Lambda_{m,s}) = \prod_m \prod_j u_{m,s,j}^{[z_{m,i}=\zeta_{m,j}]} \tag{16}$$

where m denotes different modality, for example available pixel label and discrete texture features associated with each pixel.
To avoid complete palette invariance, we also add a Dirichlet prior on the histogram palette models:

$$p(\Lambda) = p(\{u_j\}) = \frac{1}{Z(\{\alpha_j\})} \prod_j u_j^{\alpha_j - 1}, \tag{17}$$

which is estimated from the data iteratively together with other updates. The effect of this prior on the palette updates in (14) for different modalities m is $u_{m,s,j}^t \propto \alpha_{m,s,j} + \sum_i q(s_i = s)[z_{m,i}^t = \zeta_{m,j}]$, and the appropriate update on palette priors α_j can be shown to be:

$$\{\hat{\alpha}_{s,m,j}\} = \arg\max \sum_t (\alpha_{s,m,j} - 1) \log u_{m,s,j}, \tag{18}$$

$$subject\ \ to \quad \sum_u \frac{1}{Z(\{\alpha_j\})} \prod_j u_j^{\alpha_j - 1} = 1$$

The addition of the (learnable) prior over palette entry allows the model to discover and exploit consistency of local measurements across instances of a class, if there is any.

In case of real valued measurements of arbitrary dimensionality, the palette entry is defined by a mixture of C Gaussians, and the appropriate palette priors are added similarly as in the case of discrete measurements. Being a mixture, each palette entry $\Lambda_{s,c} = \{\pi_{s,c}, \mu_{s,c}, \phi_{s,c}\}$, and thus the generative model, has a hidden variable $c^t_{\mathbf{i}}$ pointing to one of the C Gaussians, which is linked to the observation in the Bayesian network.

With a flat prior on Λ, the equation (14) reduces to

$$\mu^t_{s,c} \propto \sum_i q(s^t_i = s, c^t_i = c) \cdot z^t_i \tag{19}$$

$$\phi^t_{s,c} \propto \sum_i q(s^t_i = s, c^t_i = c) \cdot (z^t_i - \mu^t_{s,c}) \cdot (z^t_i - \mu^t_{s,c.j})^T$$

$$\pi^t_{s,c} \propto \sum_i \sum_s q(s^t_i = s, c^t_i = c)$$

where $q(s^t_i = s, c^t_i = c)$ can be further factorized like in Eq.6, and $\pi^t_{s,c}$ are the mixing proportions.

The appropriate priors in this case are Gaussians with parameters $\mu_{0_{s,c}}, \psi_{0_{s,c}}$ over the means, scaled inverse Gammas of parameters $\mathbf{a}_{s,c}, \mathbf{b}_{s,c}$ over the variances and Dirichlet distributions $\beta_{s,c}$ over the mixing proportions. The effect of the prior on the palette update rules turns to be

$$\mu^t_{s,c} \propto \frac{\sum_i q(s^t_i = s, c^t_i = c) \cdot z^t_i}{\phi^t_{s,c}} + \frac{\mu_{0_{s,c}}}{\phi_{0_{s,c}}} \tag{20}$$

$$\phi^t_{s,c} = \frac{\sum_i q(s^t_i = s, c^t_i = c) \cdot (z^t_i - \mu^t_{s,c}) \cdot (z^t_i - \mu^t_{s,c.j})^T}{\sum_i q(s^t_i = s, c^t_i = c) + 2 \cdot \mathbf{a}_{s,c} - 2} + \frac{2 \cdot \mathbf{b}_{s,c}}{\sum_i q(s^t_i = s, c^t_i = c) + 2 \cdot \mathbf{a}_{s,c} - 2}$$

$$\pi^t_{s,c} \propto \beta_{s,c} + \sum_i \sum_s q(s^t_i = s, c^t_i = c)$$

Like in the previous cases, each prior is estimated from the data; $\mu_{0_{s,c}}, \psi_{0_{s,c}}$ are equal to the mean and the variance of the palette mean $\mu_{s,c}$ respectively, $\mathbf{a}_{s,c}, \mathbf{b}_{s,c}$ can be estimated fitting a Γ-function, and the update for β is the same as in Eq.18.

An example of this learned prior can be found in figure 5, together with the results related to the video segmentation.

When raw local measurements are real valued we can use *both* discrete and real valued models. The former is achieved by discretizing the measurements by a separate clustering of local measurements to create a codebook. In our experiments, mixture modeling within palette entries was superior to a forced discretization of features outside the full model (as also confirmed by [7]), but this increased the computational cost. Finally discrete and real valued modalities can be combined, in the same way the multiple discrete modalities are (see for example [44]).

It is worth mentioning that the use of high dimensional features (e.g. filter bank responses) may cause the observation likelihood to overwhelm the model $r_k(s)$. This is due to the fact that so derived features tend to be correlated, and so the model's treatment of them as independent variables leads to over-counting the evidence. The remedy to this is to either use dimensionality reduction (PCA, pLSA or LDA depending on the nature of the observation) or to scale the likelihood terms as is often done in speech research. Actually, in the latter case, it is a standard practice to raise the observation likelihood in HMMs to a power less than 1, before inference is performed on the test sample, as the acoustic signal would otherwise overwhelm the hidden process modelling the language constraints [18].

3.4 Stel and Components priors

The selection of the number of components K and the number of stels S is the first choice to do before learning a SCA's model. Despite this choice is intuitive and not critic, we introduce some prior to help; the idea is to overestimate S and K, and let the priors annihilate some of the unused stels/components.

The number of components K depend on the variability of the viewpoints/scales in the images of the objects. Nevertheless also using 2 components, we are guarantee to do better than [64, 30]. Empirically, to avoid overtrain, at least 10-20 images per component are needed.

When a larger training set is available, we can introduce a Dirichlet prior on y_k, similar in spirit to LDA [6].

$$p(\{y^t_k\}) = \frac{1}{Z(\{\omega_k\})} \prod_k \left(y^t_k\right)^{\omega_k - 1} \tag{21}$$

At this point, if SCA has sufficient components to model the data well, it becomes relatively invariant to an increase in K i.e., the additional components should be seldom used [61]. The update 22 turns to be

$$q(a^t_i = a) \propto \left(y^t_a e^{\sum_s q(s^t_i = s) \log r_a(s^t_i = s)}\right)^{\omega_k} . \tag{22}$$

Differently from K, the number of stels S, also depends on the particular object class. Before discussing the priors, it is important to note that since objects have difference appearance from their surroundings [2, 38]; this is the same assumption upon which relies [64]. Therefore choosing S=2 will always segment the image in background and foreground[2].

Nevertheless one may want to capture finer details in the object class and want to set S¿2. This is not always possible; for example pedestrian images can be broken several stels one for each leg, one for each arm, one for the head and so

[2] Choosing S=2 was not possible for [30] because of its unimodal palette

on [30], but on the other hand, for less complex classes, like Caltech's barrels, chandeliers or joshua trees, we can only expect to segment the object from the background.
We can solve this issue learning an overall prior on the stel usage and letting it kill some unused stels.

$$\pi_s \propto \sum_t \sum_i q(s_i^t = s) \tag{23}$$

The prior looks very similar to y_ks (Eq. 12) but it works at class level; it represent how much an object class uses a particular stel. The new update rule for 24 becomes

$$q(s_i^t = s) \propto \pi_s \cdot p(z_i^t | s_i^t, \{\hat{\Lambda}_s^t\}) e^{\sum_a q(a_i^t = a)\hat{r}_a(s_i^t = s)}. \tag{24}$$

We also introduced a Markov Random Field (MRF) models which enforce smoothness across pixel labels; this has proven to be very useful when the final application is image segmentation [64, 66].

3.5 Relationship to other models

We can express many other models frequently used in vision and elsewhere as special cases by assuming an appropriate number of stels S, the components K, and the palette entry size C. The color histogram model and the bag of words/features model [6, 22, 17] are achieved with $S = 1$. On the other hand, when $S > 1$, but only a single component $r_k(s)$ is used, $K = 1$, and each palette entry represents a single Gaussian, $C = 1$, and the prior over palettes is fixed to flat, our model reduces to a probabilistic index map (PIM) [30]. Finally, we get the basic ingredient of LOCUS [64] when we set $S = 2$ (foreground/background), and use a large C to represent color histograms in each palette entry. Both LOCUS and PIM contained transformation variables, which capture correlations due to a given set of simple 2D geometric transformations, while stel component analysis learns (approximately) arbitrary correlations in possible index assignments across an image. The palette choices we discuss here apply to all three models. Finally if we fix the stel partition $q(s_i^t)$ to a manual division of the image into square regions, rather than let them be estimated from images themselves, the model becomes similar to [35].

In contrast to histogram/bag-of-words models, our model parses the images so that the such models are applied only in appropriate parts of different images (thus ignoring the variable background [40], for example). In addition, the likelihood depends on the structure of the image, i.e., the extent to which the parsing of an image into stels is consistent with such parsing of other images in the same category. In this way our model is similar to the PIM model, as it can identify structural similarities among images even in presence of high variations in local measurements across images (but not within a single stel in a single image). But, in addition, stel component analysis allows for a more powerful modeling of these structural characteristics, as well as capturing, if any, feature co-occurrences in the same stel across all images.

4 Experiments

The experimental section is divided in three parts. In the first, it is shown how pixels' correlations are captured and how components are blended to better describe an object class. The second part shows how SCA deals with image and video segmentation. In the third part, SCA is used for object classification. We reported classification accuracies on the Caltech 28 dataset [12] showing how increasing the components help the recognition. More important, on Catlech 101 [21], we used $q(s^t)$ directly as features and we defined the "Stel Kernel" (SK) showing how SCA partitions $q(s^t)$ can be used in to organize other classifiers.
Before the description of each test, we show a table to indicate which modality/ies we used, their domain (discrete / continuous), and if we learned (✓) or not (×), the overall prior on the stels π_s, the Markov Random Field prior on s and the palette prior $P(\Lambda_s)$.

In all the tests, we assume the presence of one object in the image and a bounding-box annotation which can be coarse (i.e. not precisely cropped).

4.1 **Evaluation of the components**

4.1.1 How components capture pixels' correlations

Modality	**Domain**	$P(\Lambda_s)$	π_s	MRF
Gray	Cont.	×	×	×

In this experiment, we used a database [25] of 250 images of 18 subjects, each acquired at 25 different head poses (see some views in Fig. 2 and 4). The poses in the images were manually labeled with the estimated out-of-plane rotation angle (from 0 to $45\,\mathrm{deg}$). In five-fold cross-validation, we trained both PCA and SCA models (K=2, S=7, Gaussian palettes), and chose and tested the optimal predictor of the pose angle based on the component strengths of PCA and SCA. In the case of PCA, the predictors we considered used up to the 6 components with highest eigenvalues, and, furthermore, to allow for some illumination invariance, we considered sparse variants discarding the first, the first two, or the first three components, resulting in the total of 12 different sets of top eigen-images used for prediction (i.e., [1 2], [2 3], [3 4], [4 5], [5 6], [1 2 3], [2 3 4], [3 4 5], [4 5 6],

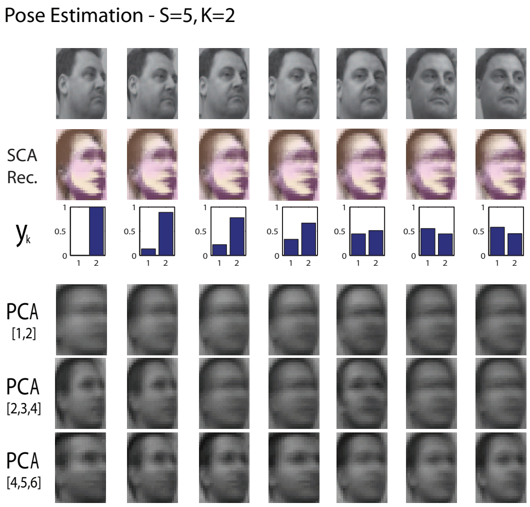

Fig. 4 SCA component strengths y_k, K=2, for a set of images of faces with varying pose, have a single degree of freedom ($y_1 + y_2 = 1$) and this degree of freedom captures the pose angle well. Below the y_k strengths, we show images generated from the model using the y_k inferred from the input. The rest of the figure illustrates the PCA reconstruction, which does not manage to separate pose from other causes of variability.

[1 2 3 4], [2 3 4 5], [3 4 5 6]).

For both PCA and SCA angle prediction, the cross-validation included linear regressor, robust linear regressor, and the nonlinear regressor.

SCA components outperformed the PCA projection as the input to regressors in this test, as the average test error for the optimal PCA-based regressor was $9.2 \deg$, and the optimal SCA-based regressor had a test error of $7.8 \deg$. Moreover, the standard deviation over the folds was twice lower for SCA, and the difference between methods was statistically significant.

As illustrated in Fig. 4, even with a single degree of freedom in the subspace, SCA does not use it to model the illumination differences, since it is palette-invariant. Rather, stels capture facial parts of relatively uniform color (therefore, uniform surface normal quite often), and the variations in y_1 captures the changes of these parts as they undergo significant geometric changes (see also Fig. 2c). Instead, the PCA model captures small geometric transformations as well as large illumination changes, but fails to capture significant structural changes, and no single PCA component captures the majority of the angle variations. In fact, the strength of the most predictive component yielded a prediction error of $13.0 \deg$ vs $7.8 \deg$ of SCA, and instead the angle has to be inferred from multiple components, and this results still lies behind SCA's single component inference.

4.1.2 A Comparison between SCA's Components and the centers of a mixture model

Modality	**Domain**	$P(\Lambda_s)$	π_s	MRF
Gray	Cont.	×	×	×

To illustrate that SCA captures the variability in the data in a substantially different way than a mixture of PIMs (MPIM, [30]) of the same complexity, we considered the Daimler dataset [43]. The full dataset is composed by 10200 gray levels images at a resolution of 18x36 pixels. It is used for pedestrian classification. Pedestrian images were obtained from manually labeling and extracting the rectangular positions of pedestrians in video images. The dataset is challenging because the negative samples (non-pedestrians) present an elongated structure, easily confusable with a pedestrian. For our experiments we randomly chose a subset of 1000 positive and 1000 negative images using half of them to train the models, the rest for testing. We repeated this process 5 times.

We learned a SCA and MPIM models with S = 5, which broke image patches into five stels, and $K \in 1, 2, 3$ which represents the number of centroids for MPIM, and the number of components for SCA.

An increase in K improved performances for both MPIM that assumes that each image is modeled by *only one* of the K components, and the SCA model, which *mixes* the components differently for each image placing it in a simplex of dimension K, to achieve a better and spatially nonuniform blend of components.

However, SCA outperforms MPIM model, with statistical significance in several aspect. First, at the maximum complexity (K=3), we obtained an area under the curve, AUC, of 0,9334 vs 0,9122 in favor of SCA. Second, MPIM has not improved significantly from K=2 to K=3, AUC was 0,9110 and 0,9122, respectively, whereas SCA has reached an AUC of 0,9221 at K=2 and of 0,9331 at K=3. This further illustrates the benefits of spatial mixing of components in SCA - expressive power of the model grows much faster with K, despite the same number of model parameters[3].

Finally we want to highlight how our way of mixing different components is very different from linear mixing, as the latter reduces to a mixture model. Linear combination of probability maps is just a mixture. But it is too rigid a model, and so the learned probability maps, in order to accommodate for the variation in the data, end up too uncertain (blurred). A better mixing model allows more freedom in mixing, which is now spatially varying, but this then results in more certain components. Overall, the latter is better as seen in the experiments.

[3] SCA has an extra layer of variables, a, but these are integrated out.

4.2 **Segmentation**

Using SCA for segmentation, allows one to extract an object only assuming its presence in the image. We show how an increase of the number of components K yields to better segmentation results.

4.2.1 Foreground Extraction from Videos

Modality	**Domain**	$P(\varLambda_s)$	π_s	MRF
Color	Cont.	✓	×	✓
Opt.Flow	Disc.	✓	×	✓

To show at which extent, SCA can deal with rigid transformation we extracted the foreground from the same video sequence used to test the hierarchical model selection strategy of [34]. This sequence is very challenging: it is composed by 220 frames and contains significant illumination changes, background clutter, various and confusing foreground and background motion, as well as dramatic changes in the size and pose of the foreground object (Fig.5A). To analyze the frames of this video using our model, we used two modalities for the local pixel measurements: real valued color (mixture of Gaussians palette) and optical flow (discrete observation) for each pixel, and the model complexity was S=3, K=3,C=3 turning off the priors on s. The comparison is based on the manual segmentation into foreground (FG) and background (BG) of one frame out of 10. The inferred parameters concerning real valued color for the Larry video are shown in Fig. 5 where $\beta_{s,c}$ is the Dirichlet prior on the mixing coefficients $\pi^t_{s,c}$, $\mu_{0_{s,c}}$ and $\phi_{0_{s,c}}$ is the Gaussian prior on the palette means $\mu^t_{s,t}$, and $\varGamma$ is the inverse gamma prior on the palette variance $\phi^t_{s,t}$. Note how in this case the prior avoids the full color invariance helping the segmentation as adjacent frames have similar color. The parsing of our model agrees with the ground truth in 95% of pixels, result comparable with that achieved by the algorithm in [34], which is based on a much more complex hierarchical model with multiple components specialized for video processing. We also show the segmentation that SCA achieves if temporal correlations among components y_k are modeled using a simple Brownian motion model (see Fig. 5, rows D at right). In this case, our model achieves an accuracy of 96%, outperforming [34].

Table 1 Video segmentation results. The difference between our result and [34] is not statistical significant. FG and BG values represent agreement with ground truth of foreground and background pixels. OV stands for the overall segmentation accuracy

Video	Method	FG	BG	OV.
`MSRiu`	TPIM [30]	97,9%	92,7%	96,1%
`MSRiu`	SCA	97,2%	94,7%	**96,6**%
`anaivana`	TPIM [30]	96,1%	92,5%	**95,1**%
`anaivana`	SCA	95,7%	91,5%	94,8%
`larry`	SCA	95,7%	90,5%	95,3%
`larry`	SCA Smoothed	96,7%	92,5%	**96,1**%
`larry`	[34]	96,3%	89,2%	95,4%
`larry`	[30]	73,1%	81,6%	80,0%
`larry`	[46]	82,0%	95,0%	92,0%

To further test the SCA ability to deal with misalignments of the object in video frames, in order to allow to track it, exploiting the ability of the transformed PIM model (TPIM) [30] to do the same.
The latter approach is much more computationally intensive, as it requires a search over many possible image transformations [23]. Even when this search is sped up in case of image translations by reducing many operations to efficiently compute convolutions, the computational burden of TPIM is significantly higher than that of SCA, whose computational cost grows only linearly with the number of components K, and typically only a handful of components is used to manage a wide variety of geometric changes in stels. In addition to `Larry` video, the two approaches were compared on other 2 video sequences: one, `MSRiu`, has been used in [67], the other one, `anaivana`, is available upon request (see some frames in Fig. 5). `MSRiu` video is characterized by a relevant FG (object) translation; instead, in `anaivana`, there is significant change of the FG scale.
In all cases, the FG segmentation using SCA is at least as good as the one achieved by a more expensive search over alignments performed by TPIM. For TPIM, we considered three different scales and nine possible shifts, making the algorithm 27 times slower than the basic PIM, and around 9 times slower than SCA. Results are shown in Fig. 5, and numerically reported in Tab. 2.

4.2.2 Image Segmentation

Modality	**Domain**	$P(\varLambda_s)$	π_s	MRF
Color	Cont.	×	✓	✓

To test the ability of SCA to extract objects from images we considered Caltech 101 annotations (101 Classes, more than 30 exemplars per class, [20]) and the Weizmann horses dataset (327 horses, [8]). This is a standard benchmark to evaluate an algorithm segmentation accuracy [1,12,51]. LOCUS [64] reports only results on the Weizmann horses.

To compute accuracy, we segmented one class at time (for Caltech 101), we learned a model and we considered the posterior distribution $q(s)$.
Given the ground truth data, there is one out of two possible labels for each pixel $l_{ij} \in \{0,1\}$, where 0 refers to

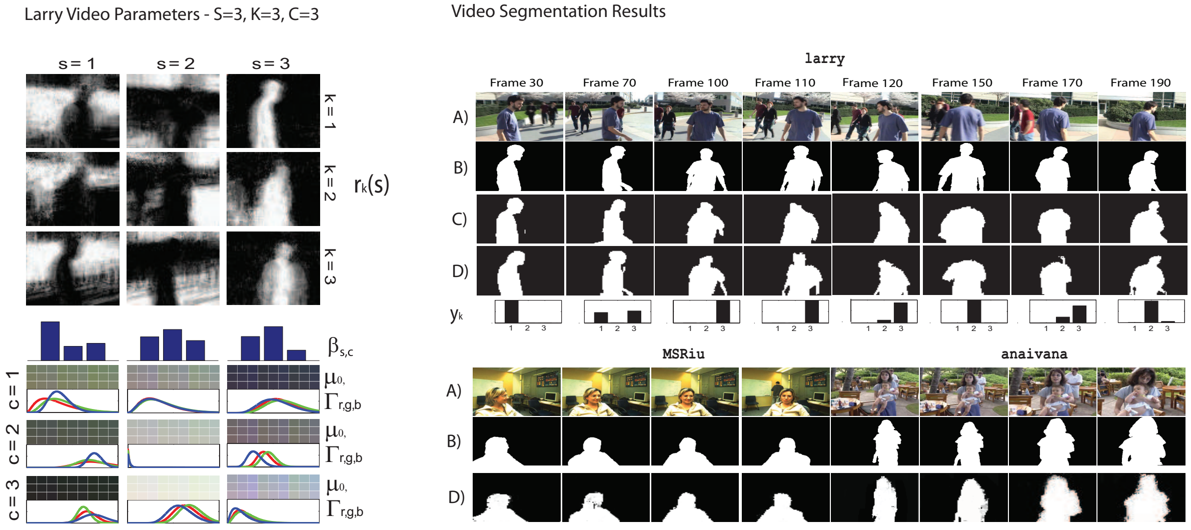

Fig. 5 Video segmentation using SCA. Left: the learned components and the color priors. Right: video segmentation results for three videos `larry`, `MSRiu`, and `anaivana`. For the `larry` video, the rows represent: A) Frames, B) Ground truth masks, C) [34] results, D) SCA with temporal smoothing of y_k.

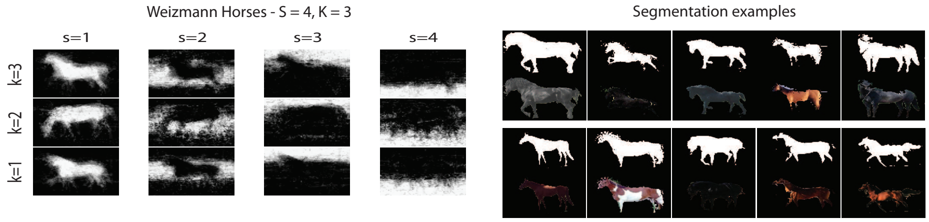

Fig. 6 Image segmentation using SCA. We show the parameters for the Weizmann horses on the left, and some segmentation results on the right. Note how the components capture the characteristic poses of the object class.

Table 2 Image segmentation results

		Caltech segmentation			
SCA K=1	SCA K=2	LOCUS [64]	ClassCut [1]	GrabCut [51]	SCTM [12]
84,71%	88,35%	n.a.	83,60%	81,50%	67,00%

		Weizmann horses			
SCA K=1	SCA K=2	LOCUS [64]	ClassCut [1]	GrabCut [51]	SCTM [12]
86,54%	90,21%	93,10%	86,20%	85,80%	81,80%

background (BG) and 1 refers to foreground (FG). After the learning phase, we have also S labels for each pixel based on the model $s_{ij} \in \{1, \ldots, S\}$. These labels are probabilistic, so we have $q(s_{ij} = s)$ rather than just a discrete s_{ij}. To create a correspondence between s and l, we need a mapping $s \rightarrow l$ to evaluate the segmentation. Since a small value for S is used by the algorithm (2, 3 or 4), it is reasonable to consider as result the best mapping.[4]

Figure 6 shows the learned model for the Weizmann horses. Note how components capture the salient poses; many other poses are obtained by blending them. Segmentation accuracies are reported in Tab. 2; they are measured by the percentage of pixels in agreement with the ground truth segmentation.
On Caltech 101, SCA outperforms [1,12,51] setting the new state of the art on segmentation using pixels. On Weizmann horses SCA does not reach the LOCUS performance (93,2%). Nevertheless LOCUS 1) cannot recognize horses facing opposite directions (in [64] images have been preprocessed to solve this issue). SCA, like [12], does not need to make such assumption, it only needs an appropriate number of components $K > 1$, so that the various poses of the objects can be captured by one component $r_k(s)$. Moreover 2) LOCUS also uses the edges and 3) it deals explicitly with deformations and transformation and it is characterized by several

[4] This operation is performed only once, for all the images and the same happens in LOCUS for example, were a prior one cannot know which one of the 2 segments represent the foreground.

submodules specialized for image segmentation making it very complex. Please also note how SCA's component could also be employed by LOCUS probably further increasing its effectiveness.

The last consideration is on the components. As Tab. 2 show, increasing the number of components K also increases the accuracy as the model becomes more expressive. The difference is statistically significant at $p \leq 0.05$.

4.3 **Object Recognition**

4.3.1 Generative Classification and "S-Bags of words"

Modality	**Domain**	$P(\Lambda_s)$	π_s	MRF
Color	Cont.	✓	×	×
SIFT	Disc.	✓	×	×

To test the performance of our model, we have trained a variety of SCA models ($S \in \{1, 3, 5\}$, $K \in \{1, 2\}$), for the 2117 images of the Caltech 28 dataset.

Caltech 28 was introduced in [12], and it is composed by 28 object classes, selected among the subset of Caltech 101 categories that contain more than 60 images per class. These categories contain objects with thin regions (e.g., flamingo, lotus), peripheral structures (e.g., cup), not well centered objects (e.g., leopards, dalmatians, Joshua trees) and, most important, it does not contain classes characterized by background artefact that makes them easily identifiable.

For each class, we randomly select 30 images for training and 30 images for testing. SIFT features were extracted from 15x15 pixel windows computed over a grid spaced 5 pixels. At the end, these features were mapped to 300 codewords.

We learnt a model for each class and we assigned a sample to the class that provides the lowest free energy. The process is illustrated in Fig.7 were for two classes (trees and faces) we show, the inferred parameters, the prior and the posterior of a negative and a positive sample.

Results are reported in Tab. 3: once again it is evident how the increasing the components help recognition. The number of stels does not affect the accuracy, being it more related to the particular object class. [12] seems to outperform SCA's generative classification, but it requires the foreground masks for each training object.

As second test we only used discretized SIFT features as observation (z_{sift}, in Fig.7). As visible in Tab.3 (Row SCA^{sift}), the accuracy raises of more than 10%. Moreover the model now is more efficient as the images are smaller (SIFT are computed on a grid, every 5 pixels). We repeated the test only using color but the accuracy got worse.

Table 3 Object recognition results. We report SCA generative classification accuracies varying S and K. Note that for S=1 we have a bag-of-word generative classifier. SCA^{sift} only uses discretized sift as observations.

	S=1	S=3	S=5
SCA K=1	12,31% [17]	49,30%	45,54%
SCA K=2	n.a.	54,34%	61,39%
SCA^{sift} K=2	n.a.	70,12%	73,23%
SPK [35]	65,40%	n.a.	n.a.
SCTM [12]	69,30%	n.a.	n.a.
LDA [22]	12,32% [22]	34,98%	32,21%
S-Bag of Words, HI Kernel	56,40% [17]	65,12%	68,21%
S-Bag of Words, Lin. Kernel	51,21% [17]	61,07%	64,64%

As further test, we illustrate the value of spatial parsing of the categories into stels. Therefore, for support vector machines without any spatial structure, we show the results obtainable by utilizing the inferred stel segmentation to relearn separate models in meaningful image parts.

We performed 1-vs-All classification using support vector machines with linear kernel (Lin.) and histogram intersection kernel (HI), concatenating all the entries of the SIFT palettes and using them as image signature (see Fig.7):

$$x^t \rightarrow \left[u^{t,c}_{s=1,sift}, u^{t,c}_{s=2,sift}, \ldots, u^{t,c}_{s=S,sift} \right] \tag{25}$$

Note that since the descriptor $< u^t_{s,sift} >$ depends on the parameters of the class in hand, when we are computing the $c-$th test, e.g., "class c"-vs-All, we must infer the posterior distributions $q(s^t|\theta_c)$ under the parameters of the $c-$th class θ_c (See Algorithm 2). As visible, organizing the feature in stels (we called this organization "S-Bags of words"), helps recognition by improving the performance of the bag of word classifier of more than 10% and matching [12].

Algorithm 2: S-Bags of Words

```
Input: Image descriptors, $\{\theta_c\}$ (a model for each object class)
Output: Classification Labels
foreach Class $c = 1 \ldots C$ do
    foreach Sample $t = 1 \ldots T$ do
        Infer $q(s^t|\theta_c)$ ;
        Compute the feature histogram in each stel $u^t_s$ ;
    Compute the kernel $K_c$;
    Learn a Support Vector Machine using $K_c$;
    $d_c \leftarrow$ Classify the test data storing the decisions ;
Combine the decisions $\{d_c\}^C_{c=1}$;
```

4.3.2 The Stel Kernel

When we tried to classify all the 101 classes of Caltech [21] using the "S-Bags of words" approach, the accuracy

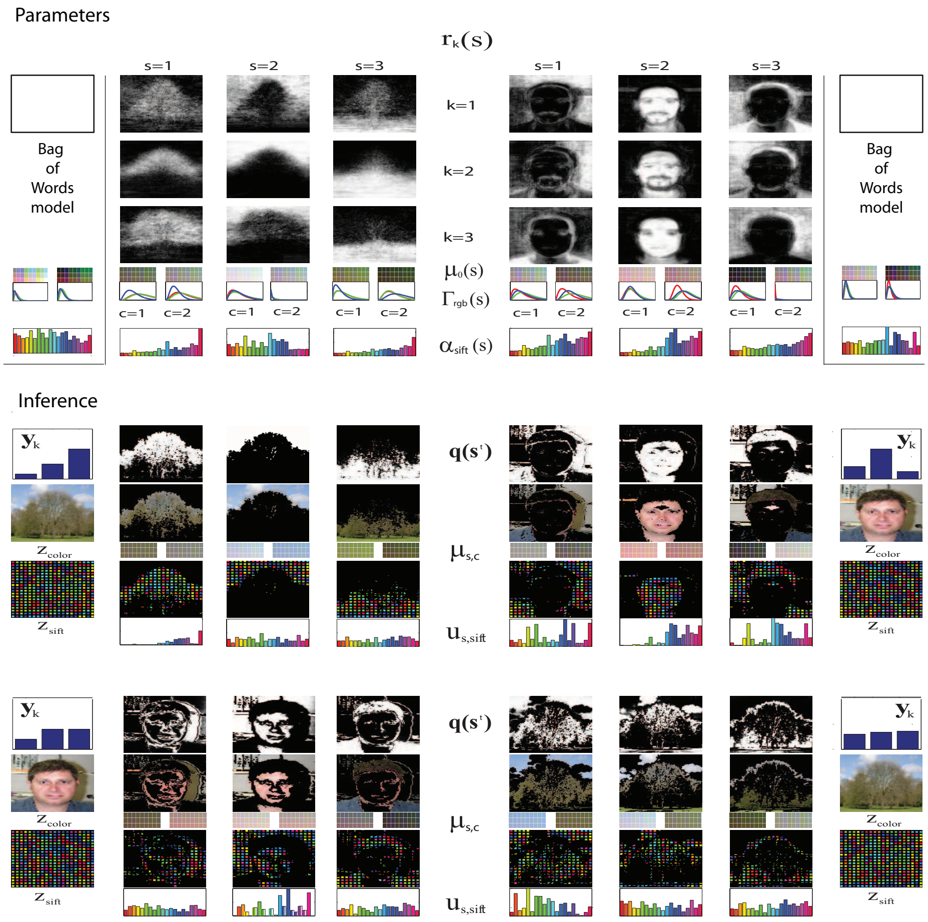

Fig. 7 Object recognition illustration. Top: Model parameters for the face and the tree categories (S=4, K=3) are compared with the BoW model (S=1). The learned Dirichlet priors for color, and SIFT (α_{sift}) are illustrated with bars whose height is proportional to the strength α_j of the different words ζ_j. The rich histogram priors for the standard BoW model (special case of our model with $s = 1$) are broken into tighter priors for appropriately estimated stels $s = 1, ..., 5$. Bottom: Inferred hidden variables under the learned models for two images.

Modality	**Domain**	$P(\Lambda_s)$	π_s	MRF
Color	Disc.	×	✓	×

dropped to 25,93% and 33,52% respectively for the linear and histogram intersection kernels. While these results are far from the state of the art[5], the performance of each one-vs-all classifier, was found to be optimal. To assess this, we performed an Anova one-way test to compare SCA ($S \in \{3, 5\}$, $K \in \{1, 2\}$), the Spatial Pyramid Kernel (two levels, L=2, SPK [35]), the Bag of Words approach (BOW, [17]) and the kernels obtained considering separately level 1 and 2 of [35] (SPK L=1, SPK L=2). We deemed the accuracies of all the classifiers as mutually independent observations.

The computation of the similarity between two images is illustrated in Fig.8b; for SPK we used the original formulation [35] while for the rest we summed the histogram intersections computed separately in each segment. The boxplot in Fig.8a summarizes the results of the Anova tests; the central red mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually with a red cross. As visible, in separate one-vs-all tests, "S-Bags of words" outperforms with a very large margin the other methods. These differences are statistically significant (p-values lower than 0.05).

The problem in combining the classifiers decisions is probably due to the fact that each support vector machine, is learned with a different kernel: this results in scaling problems. This not happens for [35, 17] where the partition is fixed, the kernel is computed just once and only the labels are changed to learn the SVMs.

Fortunately we can improve the results employing the same technique of the Hierarchical Stel Kernel (HSK, [49]). The authors of [49], firstly presented the hierarchical stel model (hSM), a relative of PIM based on a linked hierarchy of stel segmentations. Each level of the stel-hierarchy is characterized by a different number of stels. Then, exploit-

[5] We used 15 training images

Table 4 Stel Kernel Results (15 training images).

	S=1	S=3	S=5
SCA K=1	33,02% [17]	53,99%	54,05%
SCA K=2	n.a.	54,12%	55,03%
SPK [35]	53,41%	n.a.	n.a.
HSK [49]	58,92%		

ing those hierarchical segmentations, they defined a kernel as the weighted sum of the histogram intersections in each stel (see Fig. 8). As in [35] they added the level 0 (Bag of Words), and they weighted differently each level to reward more the similarities in the higher, finer ones. Being based on stel models, each class induces a kernel; the hierarchical stel kernel (HSK) is defined as the sum of all the kernels induced by the classes.

The most surprising finding of [49] is that this sum of kernels yields to a a pixel-wise weighting scheme. A feature match between the i-th pixel in one image and j-th pixel in the other, is weighted by how many times the two locations share the same stel across the hierarchy of classes.

As [49] we have that each class induces a kernel K_c defined as

$$K_c(A, B) = \sum_s \sum_k \min(u_s^A(k), u_s^B(k)) \tag{26}$$

The final kernel, named the "Stel Kernel" (SK) is defined as the sum of the kernels.

$$K^{SK} = \sum_c K_c(A, B) \tag{27}$$

The classification procedure is also shown in Algorithm 3.

It is straightforward to understand how all the properties of the HSK are valid also for the present case.

The differences between HSK and SK are that SK do not consider the level 0 (bag of words), and it presents a single segmentation for each class. On the other hand, HSK does not capture correlations and therefore it is computed from poorer segmentations.

Recognition accuracies for the full Caltech dataset are shown in Tab. 4; for the sake of comparison, we re-run the algorithm of [35] with our features[6]. We randomly select 30 images from each category: 15 of them are used for training and the rest are used for testing. We repeated this process 5 times and we averaged the results.

It is evident that each partition improves over the bag of features by over 20%. Surprisingly also SVMs benefit from the components showing how a better segmentation helps recognition. As noted in [49] for Caltech 28, the difference between SPK and SK is not statistically significant.

Table 5 Caltech 101 Dataset: Best multikernel results based on [35].

Mehtod	No. Features	No. Kernels	Best Accuracy
[24]	8	39	70,4%
[10]	4	4	70,4%
[59]	6	7	71,1%
[65]	5	10	73,2%

```
Algorithm 3: Stel Kernel
Input: Image descriptors, $\{\theta_c\}$ (a model for each object class)
Output: Classification Labels
foreach Class $c = 1 \ldots C$ do
    foreach Sample $t = 1 \ldots T$ do
        Infer $q(s^t|\theta_c)$ ;
        Compute the feature histogram in each stel $u_s^t$ ;
    Compute the kernel $K_c$ using the histogram intersection
    (Eq.26) ;
$K = \sum_c K_c$ ;
Learn a Support Vector Machine using $K$;
Classify the test data ;
```

The algorithms which produces the best accuracies on on Caltech dataset, use multiple feature and/or multikernel approaches, e.g., [10, 24, 59, 65]. In Tab5 we report some statistic. All of them use the Spatial Pyramid Kernel feature organization [35]; Tab. 4 demonstrates that kernels based on stel partitions outperform [35] with a large margin therefore it would be interesting using the stel kernel in conjunction with such methods.

4.3.3 $q(s)$ as feature

Modality	**Domain**	$P(\Lambda_s)$	π_s	MRF
Color	Cont.	×	×	×

To investigate further the capability of SCA to identify discriminant features, we considered Caltech 101, for which the best features, used discriminatively, provide classification rates of 40-59% as shown in the Tab. 6. We compare our model with several others on this dataset. In [5], GB features correspond to geometric blur which captures some of the spatial configurations in the feature distributions, and App. Color and Gray are SIFT features calculated from color and gray-level images utilized in [9, 39], and Self Similarity is introduced in [53]. The rest of the features capture gradient orientations, so mostly representing local shape features [68].

Here we only used color as a local measurement, but we performed classification using only the inferred stel segmentation, $q(s^t)$, without parts of the likelihood that have to do with matching of image measurements to those expected for the category. This corresponds to dropping out the first two

[6] The result reported in [35] (56,40%) is slightly better than our result (53,41%).

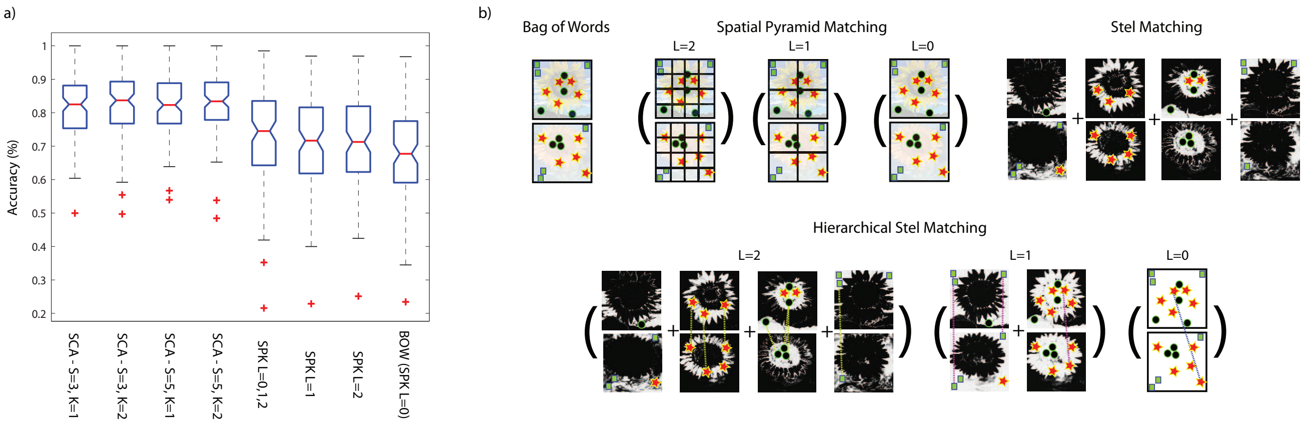

Fig. 8 On the left comparisons between the one-vs-all tests on Caltech 101. SCA outperforms with a large margin the competitors and an increase of the components yield to an increase in accuracy. On the right the four feature matching schemes considered to compute the kernels. We are considering the histogram intersection therefore we are counting the number of feature matches in each segment. Two features match if they have the same index (symbol in the figure).

terms from the free energy in Eq. (7), which deal with evaluating the uniformity of the observed features $z_{\mathbf{i}}$ and their agreement with the prior over the entire class defined by Λ. Therefore, the only terms kept are the last two terms concerned with the KL distance between the prior $r_k(s)$ and the inferred stel tessellation for the image $q(s^t)$. Such classification yields accuracy of 24,7% (18,9% at K=1) against the 17,3% obtained by [21], which represents the only reported result obtained by a generative model.

Table 6 Feature comparison on Caltech 101 dataset. We used 30 training samples.

Shape GB 1 [5]	Shape GB 2 [5]	Self Sim. [53]	Shp 180 [68]	Shp 360 [68]	Sift Col. [9]	Sift Gray [39]	$q(s)$ SCA
57%	59%	55%	48%	50%	40%	52%	50,60%

However, the discriminative use of the inferred stel, through SVM classification using only inferred stels as features resulted in classification accuracy of 50,62%, making the global shape features defined by stel segmentation of comparable quality to the top features used in object classification. This is rather encouraging as these features capture rather different aspects of the images.

It is also important to note here that for an already trained SCA model, inference of stels $q(s^t)$ for any new given image consists of only a 4-5 iterations of Eq. (24)-(12), as the SCA components r_k, and palette priors are linked to the entire category, not to a single image. Thus, inference for a single image is linear in the number of pixels, and is in fact more computationally efficient than the computations involved in methods that require a large number of filter banks or SIFT extraction. SCA does not require it when the image measurement considered is just color.

5 Conclusions

In this paper, we have proposed a novel model able to summarize a class of object 1) segmenting images in meaningful parts (stels), 2) capturing the correlations of the spatial structure, 3) identifying interesting co-occurrences of local measurements among the images of the same class.
Instead of relying on consistency of image features across images from the same class, the model mines self similarity patterns within individual images, which helps in the inference of a consistent segmentation of images into structural elements (stels), shared across the entire class, even when the images differ dramatically in local colors and features.

Significant variations in stels can be tolerated by a subspace modeling framework, Stel Component Analysis (SCA), which captures correlated changes in image structure and thus avoids over-generalization whose the PIM model (a SCA ancestor) is prone when faced with significant structural variations. The model can be inferred from the data in only with a bounding box annotation (eventually coarse), so affording significant advantages to this image representation in a variety of computer vision tasks, some of which have been illustrated above.

SCA demonstrated good performances in modeling object classes thanks to its capability to organize and select the related features, also coping with severe different aspects of the same object, in increasing the performances of discriminative methods when applied to SCA outcomes, and in segmenting images and video sequence of a certain complexity. In all these applications, it has been shown how the components helped to improve the accuracy.

References

1. Alexe, B., Deselaers, T., Ferrari, V.: Classcut for unsupervised class segmentation. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 380–393 (2010)
2. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 73–80 (2010)
3. Bagon, S., Brostovski, O., Galun, M., Irani, M.: Detecting and sketching the common. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 33–40 (2010)
4. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 404–417 (2006)
5. Berg, A.C., Malik, J.: Geometric blur for template matching. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 607–614 (2001)
6. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. Journal of machine Learning Research 3, 993–1022 (2003)
7. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2008)
8. Borenstein, E., Ullman, S.: Learning to segment. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 315–328 (2004)
9. Bosch, A., Zisserman, A., Munoz, X.: Scene classification via plsa. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 517–530 (2006)
10. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 1–8 (2007)
11. Buntine, W.: Variational extensions to em and multinomial pca. In: Proceedings of European Conference on Machine Learning (ECML). pp. 23–34 (2002)
12. Cao, L., Li, F.F.: Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 1–8 (2007)
13. Cheung, V., Jojic, N., Samaras, D.: Capturing long-range correlations with patch models. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2007)
14. Chu, X., Yan, S., Li, L., Chan, K.L., Huang, T.: Spatialized epitome and its applications. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 311–318 (2010)
15. Collins, M., Dasgupta, S., Schapire, R.: A generalization of principal component analysis to the exponential family. In: Advances in Neural Information Processing Systems (NIPS). pp. 617–624 (2001)
16. Cristani, M., Perina, A., Castellani, U., Murino, V.: Geo-located image analysis using latent representations. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2008)
17. Dance, C., Willamowski, J., Fan, L., Bray, C., Csurka, G.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision (in Conjunction with European Conference on Computer Vision). pp. 1–12 (2004)
18. Deng, L., O'Shaughnessy, D.: Speech Processing: A Dynamic and Optimization-Oriented Approach. CRC Press (May 2003)
19. Deselaers, T., Ferrari, V.: Global and efficient self-similarity for object classification and detection. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1633–1640 (2010)
20. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 594–611 (2006)
21. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding Journal (2007)
22. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 524–531 (2005)
23. Frey, B., Jojic, N.: Transformation-invariant clustering using the EM algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(1), 1 – 17 (2003)
24. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 221–228 (2009)
25. Graham, D., Allinson, N.: Characterizing virtual eigensignatures for general purpose face recognition. Face Recognition: From Theory to Applications pp. 446–456 (1997)
26. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the Alvey Vision Conference. pp. 147–151 (1988)
27. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the annual international ACM conference on Research and development in information retrieval (SIGIR). pp. 50–57 (1999)
28. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning Journal 42(1/2), 177–196 (2001)
29. Jojic, N., Perina, A., Cristani, M., Murino, V., Frey, B.: Stel component analysis: Modeling spatial correlations in image class structure. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2044–2051 (2009)
30. Jojic, N., Caspi, Y.: Capturing image structure with probabilistic index maps. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 212–219 (2004)
31. Jojic, N., Caspi, Y., Reyes-Gomez, M.: Probabilistic index maps for modeling natural signals. In: Proceedings of conference on Uncertainty in artificial intelligence (UAI). pp. 293–300 (2004)
32. Jojic, N., Frey, B.J., Kannan, A.: Epitomic analysis of appearance and shape. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 34–41 (2003)
33. Jojic, N., Perina, A., Murino, V.: Structural epitome: a way to summarize ones visual experience. In: Advances in Neural Information Processing Systems. pp. 1027–1035 (2010)
34. Jojic, N., Winn, J., Zitnick, L.: Escaping local minima through hierarchical model selection: Automatic object discovery, segmentation, and tracking in video. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 117–124 (2006)
35. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2169–2178 (2006)
36. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–791 (1999)
37. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: Workshop on statistical learning in computer vision (in conjunction with European Conference on Computer Vision). pp. 17–32 (2004)
38. Liu, T., Sun, J., Zheng, N.n., Tang, X., Shum, H.y.: Learning to detect a salient object. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2007)
39. Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 1150–1157 (1999)

40. Marszałek, M., Schmid, C.: Spatial weighting for bag-of-features. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2118–2125 (2006)
41. Matas, J., Chum, O., Martin, U., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of British Machine Vision Conference (BMVC). pp. 384–393 (2002)
42. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. International Journal of Computer Vision 60(1), 63–86 (2004)
43. Munder, S., Gavrila, D.M.: An experimental study on pedestrian classification. IEEE Trans. Pattern Anal. Mach. Intell. 28(11), 1863–1868 (2006)
44. Ni, K., Kannan, A., Criminisi, A., Winn, J.: Epitomic location recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(12), 2158–2167 (2009)
45. Perina, A., Cristani, M., Murino, V.: 2lda: Segmentation for recognition. In: Proceedings of International Conference on Pattern Recognition (ICPR). pp. 995–998 (2010)
46. Perina, A., Cristani, M., Murino, V., Jojic, N.: Capturing video structure with mixture of probabilistic index maps. In: ECCV 2008 Workshops: The 1st International Workshop on Machine Learning for Vision-based Motion Analysis - MLVMA'08. Marseille France (2008)
47. Perina, A., Cristani, M., Murino, V.: Learning natural scene categories by selective multi-scale feature extraction. Image and Vision Computing Journal 28, 927–939 (2010)
48. Perina, A., Jojic, N.: Image analysis by counting on a grid. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1985–1992 (2011)
49. Perina, A., Jojic, N., Castellani, U., Cristani, M., Murino, V.: Object recognition with hierarchical stel models. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 15–28 (2010)
50. Quattoni, A., Collins, M., Darrell, T.: Conditional random fields for object recognition. In: Advances in Neural Information Processing Systems (NIPS). pp. 1097–1104 (2004)
51. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics 23, 309–314 (2004)
52. Savarese, S., Fei-Fei, L.: 3d generic object categorization, localization and pose estimation. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 1–8 (2007)
53. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2007)
54. Shotton, J., Winn, J.M., Rother, C., Criminisi, A.: *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 1–15 (2006)
55. Sivic, J., Russell, B., Efros, A.A., Zisserman, A., Freeman, B.: Discovering objects and their location in images. In: International Conference on Computer Vision (ICCV 2005) (October 2005)
56. Su, H., Sun, M., Fei-Fei, L., Savarese, S.: Learning a dense multiview representation for detection, viewpoint classification and synthesis of object categories. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 213–220 (2009)
57. Sudderth, E.B., Torralba, A., Freeman, W.T., Willsky, A.S.: Learning hierarchical models of scenes, objects, and parts. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 1331–1338 (2005)
58. Sun, M., Su, H., Savarese, S., Fei-Fei, L.: A multi-view probabilistic model for 3d object classes. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1247–1254 (2009)
59. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 606–613 (2009)
60. Vogel, J., Schiele, B.: Semantic modeling of natural scenes for content-based image retrieval. International Journal of Computer Vision 72, 133–157 (2007)
61. Wallach, H.M., Mimno, D., McCallum, A.: Rethinking lda : Why priors matter. In: Advances in Neural Information Processing Systems (NIPS). pp. 1973–1981 (2009)
62. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 18–32 (2000)
63. Willamowski, J., Arregui, D., Csurka, G., Dance, C., Fan, L.: Categorizing nine visual classes using local appearance descriptors. In: Workshop on Learning for Adaptable Visual Systems (in Conjunction with International Conference on Pattern Recognition). pp. 1–4 (2004)
64. Winn, J., Jojic, N.: Locus: learning object classes with unsupervised segmentation. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 756–763 (2005)
65. Yang, J., Li, Y., Tian, Y., Duan, L., Gao, W.: Group-sensitive multiple kernel learning for object categorization. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 436–443 (2009)
66. Yang, Y., Hallman, S., Ramanan, D., Fowlkes, C.: Layered object detection for multi-class segmentation. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3113–3120 (2010)
67. Yin, P., Criminisi, A., Winn, J.M., Essa, I.A.: Tree-based classifiers for bilayer video segmentation. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2007)
68. Zhang, H., Berg, A.C., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2126–2136 (2006)