Multi-Class Cosegmentation

Armand Joulin[1,2,3] Francis Bach[1,4] Jean Ponce[2,3]

[1]INRIA
23 avenue d'Italie,
75214 Paris, France.

[2]Ecole Normale Supérieure
45 rue d'Ulm
75005 Paris, France.

Abstract

Bottom-up, fully unsupervised segmentation remains a daunting challenge for computer vision. In the cosegmentation context, on the other hand, the availability of multiple images assumed to contain instances of the same object classes provides a weak form of supervision that can be exploited by discriminative approaches. Unfortunately, most existing algorithms are limited to a very small number of images and/or object classes (typically two of each). This paper proposes a novel energy-minimization approach to cosegmentation that can handle multiple classes and a significantly larger number of images. The proposed cost function combines spectral- and discriminative-clustering terms, and it admits a probabilistic interpretation. It is optimized using an efficient EM method, initialized using a convex quadratic approximation of the energy. Comparative experiments show that the proposed approach matches or improves the state of the art on several standard datasets.

1. Introduction

The objective of image segmentation is to divide a picture into $K \geq 2$ regions that are deemed meaningful according to some objective criterion, homogeneity in some feature space or separability in some other one for example. Segmentation in the absence of any supervisory information remains a daunting challenge. On the other hand, when supervisory information is available, in the form of labelled training data (full images or, in interactive settings, smaller groups of pixels), accurate segmentations can be achieved (e.g., [1]).The aim of *cosegmentation* methods is to simultaneously divide a set of images assumed to contain instances of K different object classes into regions corresponding to these classes. Note that in this context, an "object" may refer to what is usually called a "thing" (a car, a cow, etc.) but might also be a texture (grass, rocks), or other "stuff" (a building, a forest) [2]. Strong supervision with hand-labelled data is typically not available in this setting. On the other hand, the presence of common object classes in multiple images provides a weak form of supervision that can be exploited by discriminative algorithms. Cosegmentation methods capable of handling large numbers of images and classes could play a key role in the development of effective automated object discovery techniques and part-based approaches to object detection for example. Unfortunately, most existing algorithms have only been demonstrated in rather restricted settings, involving only a pair of images at a time [3, 4], and/or only two *foreground* and *background* classes [5, 6, 7].

Kim et al. [8] have recently proposed the first method (to the best of our knowledge) explicitly aimed at handling multiple object classes and images. They maximize the overall temperature of image sites associated with a heat diffusion process and the position of sources corresponding to the different object classes. They use a greedy procedure guaranteed to achieve a local minimum within a fixed factor of the global optimum thanks to submodularity properties of the diffusion process (see [8] for details). We present in this paper an effective energy-based alternative that combines a spectral-clustering term [9] with a discriminative one [5], and can be optimized using an efficient expectation-minimization (EM) algorithm. Our energy function is not convex and, like [8], we can only hope to find a local minimum. Fortunately, a satisfactory initialization can be obtained by constructing a convex quadratic relaxation closely related to the cost function proposed in the two-class case by Joulin et al. [5].

The proposed approach has been implemented and tested on several datasets including video sequences. It easily handles multiple object classes and input images, and compares favorably to [8] and a simple multi-class extension of [5] in a comparative evaluation on two standard benchmarks. Furthermore, unlike the methods proposed by Kim et al. [8] and Joulin et al. [5], ours admits a probabilistic interpretation, with the potential to be easily combined with other components of an end-to-end recognition system. To summarize, the main contributions of this paper are:

[3] WILLOW project-team, Laboratoire d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA UMR 8548.

[4] SIERRA project-team, Laboratoire d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548.

- a simple and flexible energy-based formulation of true multi-class image cosegmentation that admits a probabilistic interpretation;
- a convex quadratic approximation of our energy which generalizes the cost function of [5] to the multi-class setting and affords a satisfactory initialization to the EM process; and
- an efficient algorithm that handles large numbers of input images and matches or improves the state of the art on two standard datasets.

2. Proposed model

Cosegmentation can be thought of as a multi-label pixel classification task. It is modeled in this paper as the minimization over the pixel labels of an energy function that combines local appearance and spatial consistency terms (as in spectral clustering [9]) with class-level discriminative ones (as in discriminative clustering [5, 10]) and an entropy regularizer aimed at balancing the size of the output regions.

Image representation. We assume that we are given a set $\mathcal{I}$ of images, and that each image i is sampled on a (coarse) grid $\mathcal{N}_i$ of N_i pixels. We denote by $N = \sum_{i \in \mathcal{I}} N_i$ the total number of pixels. We associate with each pixel n its color c_n, its position p_n within the corresponding image, and an additional feature $x_n \in \mathcal{X}$, that may be a SIFT vector or color histogram for example. The first two of these features are used to encode the local spatial layout and appearance of each image, and the third one is used to discriminate among different object classes in different images.

Let us denote by K the number of object classes. As is common in the cosegmentation setting, K is assumed in the following to be fixed and known a priori. We denote by y the $N \times K$ matrix such that:

$$y_{nk} = \begin{cases} 1 & \text{if the } n^{th} \text{ pixel is in the } k^{th} \text{ class,} \\ 0 & \text{otherwise.} \end{cases}$$

Given the set $\mathcal{I}$ of images, our goal is thus to find y without any other prior information.

As noted above, the idea of cosegmentation is to divide each image into K visually and spatially consistent regions while maximizing class separability across images. The first problem leads to unsupervised spectral-clustering methods such as *normalized cuts* [9] with little or no sharing of information between different images. The second one leads to multi-class discriminative clustering methods with information shared among images. Following Joulin et al. [5], we propose to combine the two approaches. However, generalizing their two-class (foreground/background) model to the multi-class setting leads to a completely different approach to discriminative clustering. Our overall energy function is the sum of spectral- and discriminative-clustering terms, plus a regularizer enforcing class-size balance. We now detail these three terms.

2.1. Spectral clustering

In cosegmentation algorithms, visual and spatial consistency is usually enforced using binary terms based on total variation [4] or the Laplacian of similarity matrices [5, 8]. While the former work well in interactive segmentation tasks [11], they do not admit the interpretation in terms of graphical spectral clustering of the latter [9]. Since our approach is closely related to a graphical model, we follow Shi and Malik [9], and use a similarity matrix W^i to represent the local interactions between pixels of the same image i. This matrix is based on feature positions p_n and color vectors c_n, which leads to high similarity for nearby pixels with similar colors. Concretely, for any pair (n, m) of pixels in i, W^i_{nm} is given by:

$$W^i_{nm} = \exp(-\lambda_p \|p_n - p_m\|_2^2 - \lambda_c \|c_n - c_m\|^2)$$

if $\|p_n - p_m\|_1 \leq 2$ and 0 otherwise. We fix $\lambda_p = 0.001$ and $\lambda_c = 0.05$ since it has been reported that these values work well in practice [5]. We denote by W the $N \times N$ block-diagonal matrix obtained by putting the blocks $W^i \in \mathbb{R}^{N_i \times N_i}$ on its diagonal, and by $L = I_N - D^{-1/2} W D^{-1/2}$ the *Laplacian* matrix, where I_N is the N-dimensional identity matrix and D the diagonal matrix composed of the row sums of W [9]. Following [5, 9], we thus include the following quadratic term into our objective function:

$$E_B(y) \quad = \quad \frac{\mu}{N} \sum_{i \in \mathcal{I}} \sum_{n,m \in \mathcal{N}_i} \sum_{k=1}^{K} y_{nk} y_{mk} L_{nm}, \qquad (1)$$

where μ is a free parameter. This term encourages an *independent* segmentation of the images into different groups, based solely on local features.

2.2. Discriminative clustering

The goal of discriminative clustering is to find the pixel labels y that minimize the value of a regularized discriminative cost function [10]. More precisely, given some labels y and some feature map $\phi : \mathcal{X} \mapsto \mathbb{R}^d$, a multi-class discriminative classifier finds the optimal parameters $A \in \mathbb{R}^{K \times d}$ and $b \in \mathbb{R}^K$ that minimize

$$E_U(y, A, b) = \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, A\phi(x_n) + b) + \frac{\lambda}{2K} \|A\|_F^2, \quad (2)$$

where $\ell : \mathbb{R}^K \times \mathbb{R}^K \mapsto \mathbb{R}$ is a loss function, y_n is the n-th column of y^T, and $\|A\|_F$ is the Frobenius norm of A. A discriminative-clustering method minimizes the response of the classifier over the set $\mathcal{Y}$ of labels, i.e, it solves:

$$\min_{y \in \{0,1\}^{N \times K},\ y1_K = 1_N} \quad \min_{A \in \mathbb{R}^{K \times d}, b \in \mathbb{R}^K} E_U(y, A, b).$$

Different choices for the loss function ℓ lead to different algorithms. In the two-class case, Joulin et al. [5] use the square loss, which has the advantage of leading to a convex problem that can be solved efficiently, but is not adapted to the multi-class setting (this is related to the *masking problem*, see Section 4.2 in Hastie et al. [12]). In this paper we use instead the soft-max loss function defined as:

$$\ell(y_n, A, b) = -\sum_{k=1}^{K} y_{nk} \log\left(\frac{\exp(a_k^T \phi(x_n) + b_k)}{\sum_{l=1}^{K} \exp(a_l^T \phi(x_n) + b_l)}\right),$$

where a_k^T is the k-th row of A, and b_k the k-th entry of b. This loss is well adapted to the multi-class setting, and it encourages a soft assignement of the pixels to the different classes [12].

Mapping approximation. Using a kernelized version of the soft-max cost function instead of a linear one is attractive since features that may not be linearly separable in $\mathcal{X}$ might easily be separated in $\mathbb{R}^d$ [13]. However, explicitly introducing the kernel matrix κ with entries $\kappa_{nm} = \phi(x_n)^T\phi(x_m)$ in either the primal or dual formulation of the minimization of E_U requires the evaluation of $O(N^2)$ kernel values at each step of the optimization [14], which may be prohibitively expansive. In the case where κ is known but ϕ is not, a common trick is to construct an incomplete Cholesky decomposition [13] of κ—that is, calculate a matrix $\psi \in \mathbb{R}^{N\times d}$ such that $\psi\psi^T \approx \kappa$, then directly use Eq. (2), where $\phi(x_n)$ has been replaced by ψ_n, where ψ_n^T is the n-th row of ψ.

This is the method used in this paper for efficiency. Since our features are histograms, we use the χ^2-kernel defined by

$$\kappa_{nm} = \exp\left(-\lambda_h \sum_{d=1}^{D} \frac{(x_{nd} - x_{md})^2}{x_{nd} + x_{md}}\right),$$

where $\lambda_h > 0$ (in the experiments, we use $\lambda_h = 0.1$). With a slight abuse of notation, we still use $\phi(x_n) = \psi_n$ to denote the approximated mapping in the rest of this presentation.

2.3. Cluster size balancing

A classical problem with spectral- and discriminative-clustering methods is that assigning the same labels to all the pixels leads to perfect separation. A common solution is to add constraints on the number of elements per class [10, 15]. Despite good results, this solution introduces extra parameters and is hard to interpret. Another solution is to encourage the proportion of points per class and per image to be close to uniform. An appropriate penalty term for achieving this is the entropy of the proportions of points per image and per class:

$$H(y) = -\sum_{i\in\mathcal{I}}\sum_{k=1}^{K}\left(\frac{1}{N}\sum_{n\in\mathcal{N}_i} y_{nk}\right)\log\left(\frac{1}{N}\sum_{n\in\mathcal{N}_i} y_{nk}\right). \quad (3)$$

As shown later, there is a natural interpretation that allows us to set the parameter in front of this term to 1.

Weakly supervised segmentation. Cosegmentation can be seen as a "very weakly" supervised form of segmentation, where one knows that K object classes occur in the images, but not which ones of the K do occur in a given image. Indeed, our entropy term encourages (but does not force) every class to occur in every image. Our framework is easily extended to *weakly supervised segmentation*, where tags are attached to each image i, specifying the set K_i of object classes appearing in it: This simply requires replacing the sum over indices k varying from 1 to K in Eq. (3) by a sum over indices k in K_i. For any pixel n in image i, this naturally encourages y_{nk} to be zero for any k nor in K_i.

2.4. Probabilistic interpretation

Combining the three terms defined by Eqs. (1)–(3) we finally obtain the following optimization problem:

$$\min_{\substack{y\in\{0,1\}^{N\times K},\\ y1_K=1_N}} \left[\min_{\substack{A\in\mathbb{R}^{d\times K},\\ b\in\mathbb{R}^K}} E_U(y, A, b)\right] + E_B(y) - H(y). \quad (4)$$

Let us show that the labels y can be seen as latent variables in a directed graphical model [16]. First, for each pixel n, we introduce a variable t_n in $\{0,1\}^{|\mathcal{I}|}$ indicating to which image n belongs, as well as a variable z_n in $\{1,\dots,M\}$ giving for each pixel n some *observable information*, e.g., some information about its true label or its relation with other pixels. The resulting directed graphical model ($x \rightarrow y \rightarrow z \leftarrow t$) defines the label y as a latent variable of the observable information z given x. Given some pixel n, this model induces an "explain away" phenomenon: the label y_n and the variable t_n compete to explain the observable information z_n. This model can be seen as an extension of topic models [17, 18] where the labels y represent *topics* which explain the *document* z given the *words* x, independently of the *group of documents* t from which z has been taken. More precisely, we suppose a bilinear model:

$$P(z_{nm} = 1 \mid t_{ni} = 1,\ y_{nk} = 1) \quad = \quad y_{nk} G_m^{ik} t_{ni},$$

where $\sum_{m=1}^{N} G_m^{ik} = 1$, and we show in the supplementary material that the problem defined by Eq. (4) is equivalent to the mean-field variational approximation of the following (regularized) negative conditional log-likelihood of $Y = (y_1,\dots,y_N)$ given $X = (x_1,\dots,x_N)$ and $T = (t_1,\dots,t_n)$ for our model:

$$\min_{\substack{A\in\mathbb{R}^{d\times K}, b\in\mathbb{R}^K,\\ G\in\mathbb{R}^{N\times K|\mathcal{I}|},\\ G^T 1_N = 1,\ G\geq 0}} -\frac{1}{N}\sum_{n=1}^{N}\log\left(p(y_n \mid x_n,\ t_n)\right) + \frac{\lambda}{2K}\|A\|_2^2.$$

The introduction of the variable z makes our model suitable for a a semi-supervised setting where z would encode "must-link" and "must-not-link" constraints between pixels. This may prove particularly useful when superpixels are used, since it is equivalent to adding "must-link" constraints between pixels belonging to the same superpixel (in this case, M is the total number of superpixels).

3. Optimization

We now present a non-convex relaxation of our combinatorial problem, which leads to an optimization scheme based on an expectation-maximization (EM) procedure, which can be initialized by efficiently solving a convex optimization problem closely related to [5].

3.1. EM algorithm

We use a continuous relaxation of our combinatorial problem, replacing the set of possible y values by the convex set $\mathcal{Y} = \{y \in [0,1]^{N \times K} \mid y 1_K = 1_N\}$. In this setting, y_{nk} can be interpreted as the probability for the n-th point to be in the k-th class. Our cost function is a difference of convex functions, which can be optimized by either *difference-of-convex* (DC) programming [19] or a block-coordinate descent procedure. We choose the latter, and since our energy is closely related to a probabilistic model, dub it an EM procedure with a slight abuse of notation.

M-step. For some given value of y, minimizing $E_U(y, A, b)$ in terms of (A, b) is a (convex) softmax regression problem which can be solved efficiently by a quasi-Newton method such as L-BFGS [20].

E-step. For given A and b, the cost function of Eq. (4) is convex in $y \in \mathcal{Y}$, and can thus be minimized with a simple projected gradient descent method on $\mathcal{Y}$. This first-order optimization method is slower than the second-order one used in the M-step, and it is the bottleneck of our algorithm, leading us to use superpixels for improved efficiency.

Superpixels. We oversegment every image i into $\mathcal{S}_i$ superpixels. For a given image i, this is equivalent to forcing every pixel n in $\mathcal{N}_i$ in a superpixel s to have the same label $y_n = y_s$. Denoting by $|s|$ the number of pixels contained in a superpixel s, each term of our cost function depending directly on y is reduced to:

$$\left\{ \begin{array}{lcl} E_U(y) & = & \frac{1}{N} \sum_{s \in \mathcal{S}} y_s (A\Phi_s + |s| b), \\ E_B(y) & = & \frac{\mu}{2} \sum_{i \in \mathcal{I}} \sum_{s,t \in \mathcal{S}_i^2} y_{sk} y_{tk} \Lambda_{st}, \end{array} \right.$$

where $E_U(y)$ is the part of $E_U(y, A, b)$ depending on y, $\Phi(s) = \sum_{n \in s} \phi(x_n)$, and $\Lambda_{st} = \sum_{n \in s} \sum_{m \in t} L_{nt}$. The entropy has the form:

$$H(y) = -\sum_{i \in \mathcal{I}} \sum_{k=1}^{K} \left(\frac{1}{N} \sum_{s \in \mathcal{S}_i} |s| y_{sk} \right) \log \left(\frac{1}{N} \sum_{s \in \mathcal{S}_i} |s| y_{sk} \right).$$

Since the problem defined by Eq. (4) is not jointly convex in (A, b) and y, a reasonable initial guess is required. In the next section, we propose a convex approximation of our cost function that can be used to compute such a guess. Moreover we show that this approximation is closely related to the the cost function proposed by Joulin et al. [5]. This allows us to use a modified version of their algorithm to initialize ours.

3.2. Quadratic relaxation

Cosegmentation is caracterized by the lack of prior information on the classes present in the images. A reasonable initial guess for our model parameters is thus to assume a uniform distribution y_n^0 of the classes over each pixel n, and to predict a pixel's class using a linear model whose parameters are independent of the corresponding feature value, which is easily shown to be equivalent to

$$\ell(y_n^0, 0) = \sum_{k=1}^{K} \frac{1}{K} \log(K).$$

We thus propose to approximate our cost function by its second-order Taylor expansion around y^0 (see the supplementary material for the calculation):

$$J(y) = \frac{K}{2} \left[\mathrm{tr}(yy^T C) + \frac{2\mu}{NK} \mathrm{tr}(yy^T L) - \frac{1}{N} \mathrm{tr}(yy^T \Pi_I) \right], \quad (5)$$

where $\Pi_I = I_N - \Lambda$, and Λ is the $N \times N$ block diagonal matrix where there is a block equal to $\frac{1}{N_i} 1_{N_i} 1_{N_i}^T$ for each image i. Note that the projection matrix Π_I centers the data for each image independently. Finally, the matrix C in Eq. (5) is equal to:

$$C = \frac{1}{N} \Pi_N (I - \Phi(N\lambda I_K + \Phi^T \Pi_N \Phi)^{-1} \Phi^T) \Pi_N,$$

where the projection matrix $\Pi_N = I - \frac{1}{N} 1_N 1_N^T$ centers the data *across all images*. Note that C is closely related to the solution of the ridge regression (or Tikhonov regularization) of y over Φ [5].

The first two terms in Eq. (5) add up to the cost function of Joulin et al. [5] (up to a multiplicative constant). The last term is a non-convex quadratic penalization encouraging a uniform distribution over classes on each image. We replace it (during initialization only) by linear constraints that force the pixels in any class k to represent at most 90% of the pixels in each image i, and at least 10% of the pixels in all other images:

$$\sum_{n \in \mathcal{N}_i} y_{nk} \leq 0.9 N_i \; ; \; \sum_{j \in \mathcal{I} \backslash i} \sum_{n \in \mathcal{N}_j} y_{nk} \geq 0.1(N - N_i).$$

These constraints generalize those in [5] to the multi-class case, and using them has the added benefit of allowing us to use a slightly modified version of their publicly available software.[1] However, the output of this code is the $N \times N$ matrix $Y = yy^T$ and not y, thus a rounding step is necessery to initialize our algorithm. The standard approach to this kind of problem is to use either *k-means* or a Gaussian mixture model (GMM) over the eigenvectors associated with the K highest eigenvalues [21] for this purpose.

[1] http://www.di.ens.fr/ joulin/

images	class	Ours	[8]	[5]	[7]
30	Bike	**43.3**	29.9	42.3	42.8
30	Bird	**47.7**	29.9	33.2	-
30	Car	**59.7**	37.1	59.0	52.5
24	Cat	**31.9**	24.4	30.1	5.6
30	Chair	**39.6**	28.7	37.6	39.4
30	Cow	**52.7**	33.5	45.0	26.1
26	Dog	**41.8**	33.0	41.3	-
30	Face	**70.0**	33.2	66.2	40.8
30	Flower	**51.9**	40.2	50.9	-
30	House	51.0	32.2	50.5	**66.4**
30	Plane	21.6	25.1	21.7	**33.4**
30	Sheep	**66.3**	60.8	60.4	45.7
30	Sign	**58.9**	43.2	55.2	-
30	Tree	**67.0**	61.2	60.0	55.9
	Average	**50.2**	36.6	46.7	40.9

Table 1. Binary classification results on MSRC. Best results in bold.

Practical issues. Initializing our algorithm with the convex approximation proposed in this section usually leads to good results, but sometimes fails completely, due to the masking problem mentioned earlier. Therefore, we also start our EM procedure with five random intializations. We compare the final values of our cost function obtained from these initializations, and pick the solution associated with the lowest value as our result. An effective rounding procedure is also a key to good performance. Thus, we perform both the k-means and GMM rounding procedures, run one M-step for each of the corresponding initializations, and run the rest of the algorithm with the one yielding the lowest value of the cost function.

4. Implementation and results

4.1. Experimental set-up

We use the watershed algorithm [22] to find superpixels. The rest of our algorithm is coded in MATLAB. Since a good initialization is crucial, we use a modified version of [5] to initialize our method as explained in Section 3.2. The complexity of our algorithm is $O(NK)$, and its running time (*including* [5]) typically varies from 30mn to one hour for 30 images, depending on the number of superpixels (this could be improved using a C implementation and exploiting the fact that parts of our algorithm are easily parellelized).

We present qualitative multi-class cosegmentation results on various datasets in the rest of this section. We also present quantitative comparisons with Kim et al. [8][2], Mukherjee et al. [7] and Joulin et al. [5] on two standard benchmarks, MSRC-v2[3] and iCoseg [23].[4] We use the publicly available versions of [5, 8] and set their free parameters so as to maximize their performance for the sake of fairness. Likewise, we set the free parameter μ of our algorithm by trying $\mu = 10^k$ for $k \in \{0, \dots, 4\}$, and keeping the value leading to the best performance (taking $\mu = 0.1$ works well in all our experiments in practice).

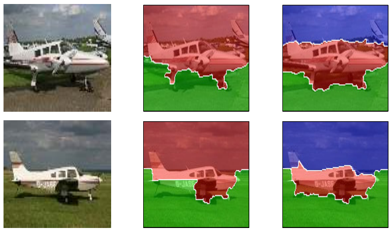

Figure 2. This figure shows how increasing the number of classes leads to a better segmentation. Columns 2 to 3 respectively show results for $K = 2$ and $K = 3$ (best seen in color).

The images in iCoseg only have two labels, and MSRC is not well suited to a multi-class evaluation because of its "clutter" class that does not correspond to a well-defined visual category. We have thus used the main "object" category for each MSRC image as foreground, and the rest of the pixels as background, and limited our quantitative evaluation to the binary case. Segmentation performance is measured by the *intersection-over-union* score that is standard in PASCAL challenges and defined as $\max_k \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \frac{GT_i \cap R_i^k}{GT_i \cup R_i^k}$, where GT_i is the ground truth and R_i^k the region associated with the k-th class in the image i.

4.2. MSRC two-class experiments

Qualitative results obtained on the MSRC-v2 database with two classes are shown in Figure 1. Table 1 gives a quantitative comparison with [5, 8, 7].[5] Note that the algorithm proposed in [7] fails to converge on 5 out of 14 classes. Our algorithm achieves the best performace for 12 out of 14 object classes. We use SIFT for discriminative clustering here because of the high appearance variability of MSRC.

This experiment calls for some additional comments: First, it is interesting to note that our method works best for faces, despite the high background variability compared to sheep or cow for example. Second, for classes with very high variability (e.g., cat, dog, or chair), the three cosegmentation algorithms perform rather poorly, as expected. Third, it appears that the low performance on the bike class is caused by too-coarse superpixels. Finally, the poor performance of our algorithm on the plane category is mostly due to the fact that the background is (essentially) always the same, and is composed of two kinds of "stuff", i.e., grass and sky, as shown in Figure 2. Therefore, with only two classes, our algorithm simply separates sky+plane from grass, which motivates the need for multi-class cosegmentation as demonstrated in the next section.

4.3. Multi-class experiments

We present in this section our experiments with multiple object categories using the recently released iCoseg

[2] http://www.cs.cmu.edu/∼gunhee/r_seg_submod.html

[3] http://research.microsoft.com/en-us/projects/ObjectClassRecognition/

[4] http://chenlab.ece.cornell.edu/projects/touch-coseg/

[5] There is no error bar since we test on the maximum number of images per class.

dataset	images	class	K	Ours	multiclass Joulin et al. [5]	Kim et al. [8]	Joulin et al. [5]
iCoseg	25	Baseball player	5	**62.2**	53.5	51.1	24.9
	5	Brown bear	3	75.6	**78.5**	40.4	28.8
	15	Elephant	4	**65.5**	51.2	43.5	23.8
	11	Ferrari	4	**65.2**	63.2	60.5	48.8
	33	Football player	5	**51.1**	38.8	38.3	20.8
	7	Kite Panda	2	57.8	58.0	**66.2**	58.0
	17	Monk	2	**77.6**	76.9	71.3	76.9
	11	Panda	3	**55.9**	49.1	39.4	43.5
	11	Skating	2	**64.0**	47.2	51.1	47.2
	18	Stonehedge	3	**86.3**	85.4	64.6	62.3
MSRC	30	Plane	3	**45.8**	39.2	25.2	25.1
	30	Face	3	**70.5**	56.4	33.2	66.2
		Average		**64.8**	58.1	48.7	43.9

Table 2. Results on iCoseg and MSRC using more than two segments. Here, K indicates the number of segments used for our algorithm.

database, along with two MSRC classes. iCoseg provides a setting closely related to video segmentation in the sense that, for a given class, the images are similar to key frames in a video, with similar lighting and background. As in the case of the plane in Figure 2 (first two columns), this makes binary segmentation very difficult (sometimes meaningless) since multiple object classes may be merged into a single one. As shown by Figure 2 (right), adding more classes helps.

The number of meaningful "objects" present in the images varies from one problem to the next, and K must be set by hand. In practice, we have tried values between 2 and 5, and Figure 3 shows that this gives reasonably good results in general. Quantitative results are given in Table 2. Since, as argued earlier, MSRC and iCoseg are not well adapted to benchmarking true multi-class cosegmentation, we report the maximum of the intersection-over-union scores obtained for the K classes against the "object" region in the ground-truth data.

As before, we use SIFT features for the two MSRC classes used in this experiment. Due to little change in illumination, we use instead color histograms for iCoseg, which are in general more appropriate than SIFT ones in this setting.[6] We compare our algorithm with both our multiclass implementation of Joulin et al. [5] and their original implementation (with $K = 2$) using the same features as ours. We also compare our method to Kim et al. [8] with K between 2 and 5, and keep the K value with the best performance.

We obtain the best performance for 10 of the 12 classes, including the MSRC plane category for which our two-class algortihm only obtained 21.6% in our previous experiment. Note that we do not claim that using multiple classes solves the binary cosegmentation problem. Indeed, we do not know which one of the K classes corresponds to the "foreground" object. On the other hand, our experiments suggest that this object is indeed rather well represented by one of the classes in most of our test cases, which may be sufficient to act as a filter in an object discovery setting for example [24].

Of course, our method, like any other, makes mistakes, sometimes giving completely wrong segmentations. Figure 4 shows a few examples.

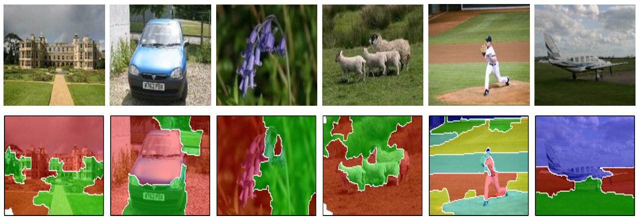

Figure 4. Some failure cases.

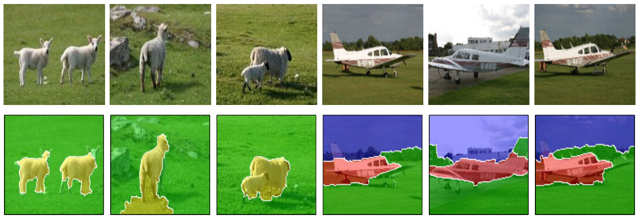

Figure 5. Weakly supervised segmentation results with known tags and SIFT features.

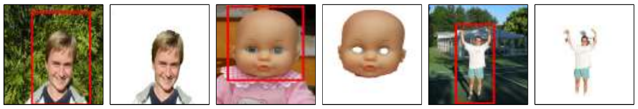

Figure 6. Interactive segmentation results with color histogram features.

4.4. Extensions

Let us close this section with a few proof-of-concept experiments involving simple extensions of our framework.

Weakly supervised segmentation. We start with the case where each image is tagged with the object classes it contains. As explained in Section 2, this can be handled by a

[6] SIFT features lead to better performance in some of the cases (for example, the performance rises to 85.2% for the brown bear class and to 75.9% for pandas), but for a fair comparison we keep the same features for the entire dataset.

simple modification of our entropy-based regularizer. Figure 5 shows qualitative results obtained using 60 sheep and plane images in the MSRC database, labelled with tags from the set {sheep, plane, grass, sky}. The performance is essentially the same as when the two sets of images are segmented separately, but the grass is now identified uniquely in the 60 images.

Interactive segmentation. The weakly supervised version of our method is itself easily generalized to an interactive setting, as in GrabCut [1], where the user defines a bounding box around the object of interest. For us, this simply amounts to picking a foreground or background label for each pixel inside the box, and a background label for all the pixels outside. Figure 6 shows a few qualitative examples obtained using this method. Again, these are just for proof of concept, and we do not claim to match the state of the art obtained by specialized methods developed since the introduction of [1].

Video segmentation. Our experiments with iCoseg suggest that our method is particularly well suited to keyframes from the same video shot, since these are likely to feature the same objects under similar illumination. This is confirmed with our experiments with two short video clips taken from the Hollywood-2 and Grundmann datasets [25, 26]. We pick five key frames from each video and cosegment them using color features without any temporal information such as frame ordering or optical flow. As shown by Figure 7, reasonable segmentations are obtained. In particular, the main characters in each video are identified as separate segments.

5. Conclusion

We have presented a true multi-class framework for image cosegmentation, and shown that it compares favorably with the state of the art. We have also presented preliminary extensions to related problems such as weakly supervised or interactive cosegmentation, and the joint segmentation of video key frames. Next on our agenda are incorporating motion information in the analysis of video clips and using cosegmentation as a front end to an object recognition/detection system.

Acknowledgments. This paper was partially supported by the Agence Nationale de la Recherche (MGA Project) and the European Research Council (SIERRA and VIDEOWORLD Projects).

References

[1] A. Blake, C. Rother, and V. Kolmogorov. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004. 1, 7

[2] D. Forsyth, J. Malik, M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler. Finding pictures of objects in large collections of images. *ORCV II*, 1996. 1

[3] C. Rother, V. Kolmogorov, T. Minka, and A. Blake. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *CVPR*, 2006. 1

[4] S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation revisited: models and optimization. In *ECCV*, 2010. 1, 2

[5] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010. 1, 2, 4, 5, 6

[6] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, 2011. 1

[7] L. Mukherjee, V. Singh, and J. Peng. Scale invariant cosegmentation for image groups. In *CVPR*, 2011. 1, 5

[8] G. Kim, E.P. Xing, L. Fei-Fei, and T. Kanade. Distributed cosegmentation vis submodular optimization on anisotropic diffusion. In *ICCV*, 2011. 1, 2, 5, 6

[9] J. Shi and J. Malik. Normalized cuts and image segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 1997. 1, 2

[10] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *NIPS*, 2005. 2, 3

[11] Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *ICCV*, 2001. 2

[12] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001. 3

[13] J Shawe-Taylor and N Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. 3

[14] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007. 3

[15] F. Bach and Z. Harchaoui. Diffrac : a discriminative and flexible framework for clustering. In *NIPS*, 2007. 3

[16] M. J. Wainwright and M. I. Jordan. *Graphical models, exponential families, and variational inference*, volume 1. Foundations and Trends in Machine Learning, 2008. 3

[17] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent Dirichlet allocation. *JMLR*, 3:2003, 2003. 3

[18] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. In *ICCV*, 2007. 3

[19] A.L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003. 4

[20] D.C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989. 4

[21] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001. 4

[22] A. S. Wright, A. S. Wright, and S. T. Acton. Watershed pyramids for edge detection. In *ICIP*, 1997. 5

[23] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010. 5

[24] B. C. Russell, W. T. Freeman, A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 6

[25] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010. 7

[26] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 7

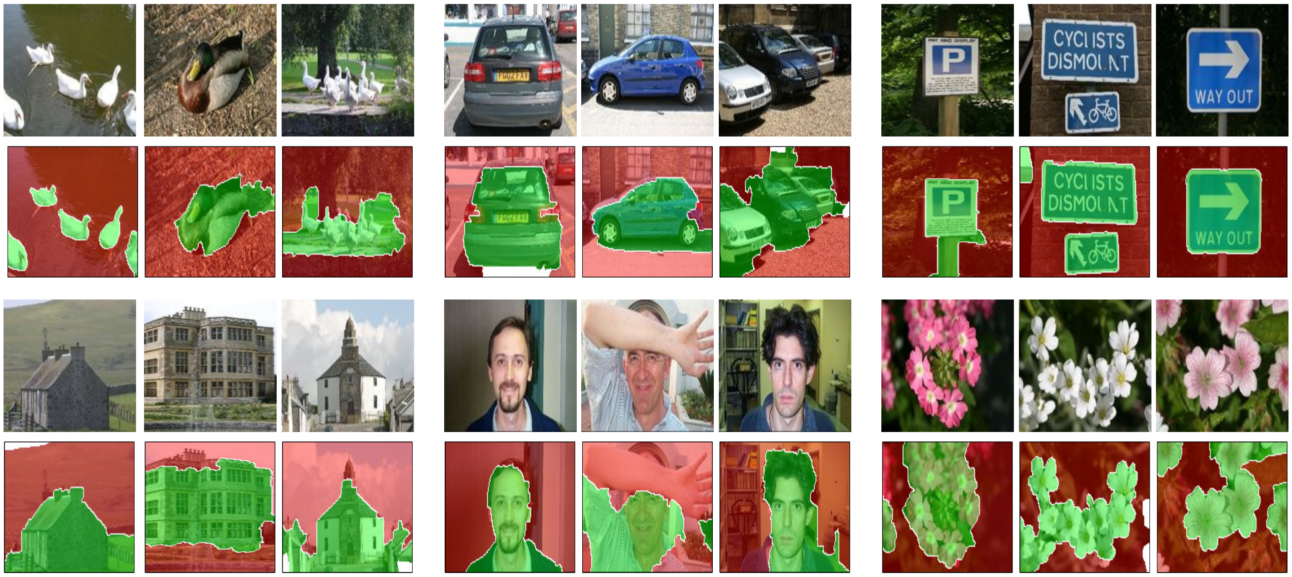

Figure 1. Results on binary classification. There are three set of images by row. The images are taken from MSRC and the features are SIFT.

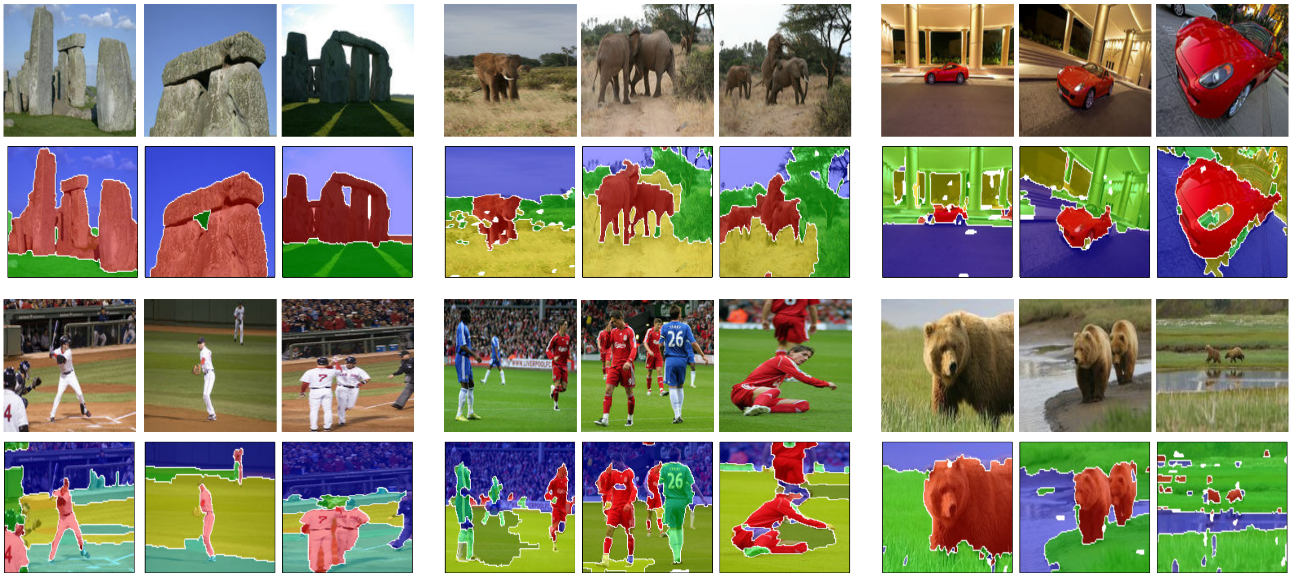

Figure 3. Results for the cosegmentation with multiple classes. There are three experiments by row with respectively. The images are taken from iCoseg and the features are color histograms.

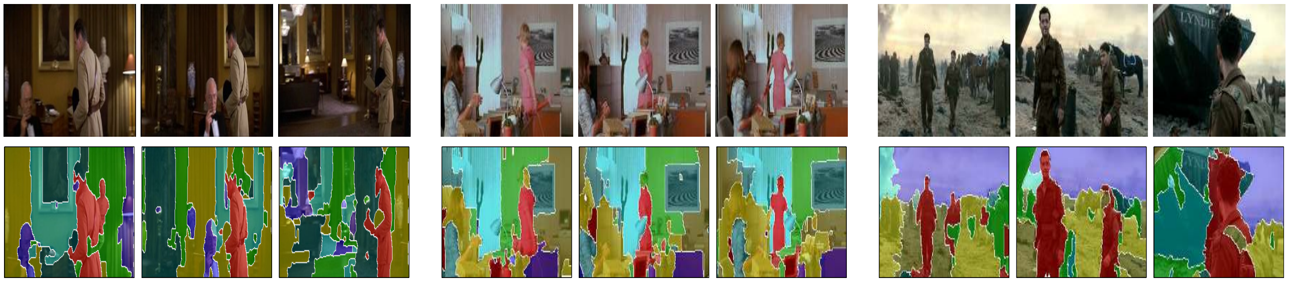

Figure 7. Results on two videos. The first row represent the input images, the second one is the segmentation obtained with our algorithm.