Cosegmentation and Cosketch by Unsupervised Learning

Jifeng Dai[1,2], Ying Nian Wu[2], Jie Zhou[1], and Song-Chun Zhu[2]
[1] Department of Automation, Tsinghua University, China
daijifeng001@gmail.com, jzhou@tsinghua.edu.cn
[2]Department of Statistics, University of California, Los Angeles (UCLA), USA
{ywu, sczhu}@stat.ucla.edu

Abstract

Cosegmentation refers to the problem of segmenting multiple images simultaneously by exploiting the similarities between the foreground and background regions in these images. The key issue in cosegmentation is to align common objects between these images. To address this issue, we propose an unsupervised learning framework for cosegmentation, by coupling cosegmentation with what we call "cosketch". The goal of cosketch is to automatically discover a codebook of deformable shape templates shared by the input images. These shape templates capture distinct image patterns and each template is matched to similar image patches in different images. Thus the cosketch of the images helps to align foreground objects, thereby providing crucial information for cosegmentation. We present a statistical model whose energy function couples cosketch and cosegmentation. We then present an unsupervised learning algorithm that performs cosketch and cosegmentation by energy minimization. Experiments show that our method outperforms state of the art methods for cosegmentation on the challenging MSRC and iCoseg datasets. We also illustrate our method on a new dataset called Coseg-Rep where cosegmentation can be performed within a single image with repetitive patterns.

1. Introduction

Recently, the problem of cosegmentation has attracted considerable attention from the vision community. Cosegmentation refers to the problem of segmenting multiple images into foreground and background simultaneously by aligning similar objects or regions across different images.

To address this alignment problem, we propose an unsupervised learning framework for cosegmentation. The key idea is to couple the task of cosegmentation with what we call "*cosketch*." The goal of cosketch is to learn a codebook of deformable shape templates that are shared by the input images, and to sketch the images by these commonly shared templates. Fig. 1 illustrates the basic idea. A codebook of two shape templates (head and body) are learned from a set of input images of deer that are not a priori aligned or annotated. These shape templates capture distinct and specific image patterns and the same template is matched to similar image patches in different images. Each shape template is associated with a segmentation template to be explained below. The sketch of the input images by these two templates help establish correspondence between different images, and the associated segmentation templates provide crucial top-down information for segmentation.

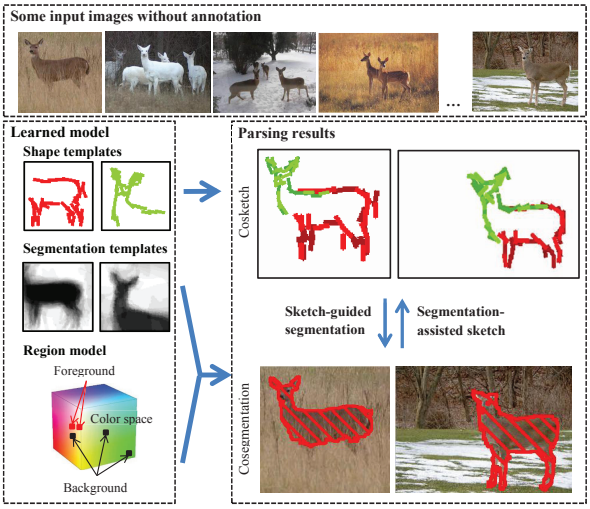

Figure 1. Distinct shape templates are learned (from 19 input images) and are matched to specific image patches in different images. Shape templates are coupled with segmentation templates that provide top-down clues for segmentation.

Model. The learned model consists of the following three components.

(1) *Sketch model*. It seeks to encode the "sketchable" patterns of the input images by a codebook of shape templates. The sketchable patterns include region boundaries as well as non-boundary edges and lines. Each shape tem-

plate is represented by an active basis model [23], which is a generative model with explicit variables for shape deformations and is suitable for unsupervised learning.

(2) *Region model*. It seeks to encode the "non-sketchable" patterns such as region interiors and shapeless patterns such as sky and water etc. Each pixel of an input image is assigned a label indicating which region this pixel belongs to. The region model is defined conditional on the pixel labels. It is in the form of a Markov random field, which models marginal distributions of pixel colors and pairwise similarities between neighboring pixels.

(3) *Coupling*. The sketch model and region model are coupled by associating each shape template with a *segmentation template*, which is in the form of a *probability map* of pixel labels. That is, for each pixel within the bounding box of the shape template, the probability map gives the probability that this pixel belongs to each region. These probability maps provide top-down prior information for pixel labels in the region model. Conversely, the pixel labels obtained from segmentation serve as data for the probability maps, and they provide bottom-up information for inferring sketch representation.

Unsupervised learning algorithm. Fitting the above model by energy minimization leads to a relaxation algorithm that alternates the following two steps.

(I) *Image parsing*: Given the current shape templates, segmentation templates and the parameters for the shape and region models, sketch the images by the shape templates, and segment the images by graph cuts [4].

(II) *Re-learning*: Given the current image sketches and segmentations, re-learn the shape templates, segmentation templates and model parameters.

The image parsing step itself consists of two sub-steps.

(I.1) *Sketch-guided segmentation*. Given the current sketches of the images by the shape templates, segment the images by graph cuts with the associated segmentation templates as prior.

(I.2) *Segmentation-assisted sketch*. Given the current pixel labels of segmentation, sketch the images by matching the shape templates and the associated segmentation templates to the images and their label maps respectively.

Random initialization with no preprocessing. The shape templates and the associated segmentation templates are initialized by learning from randomly cropped image patches, without any sophisticated pre-processing. Relaxation by energy minimization automatically results in alignment and segmentation, while distinct templates are being learned.

Experiments, datasets and performances. We evaluate the proposed method on the **MSRC** [20] and **iCoseg** [3] datasets. Our method achieves higher accuracies than state of the art methods. To further test the proposed method, we collect a new dataset called **Coseg-Rep**, which contains 23 object categories with 572 images. One special category contains 116 images such as tree leaves, where similar shape patterns repeat themselves within the same image. As a result, cosegmentation can be performed on each single image. This dataset will be released with the paper.

2. Related work

Existing methods for cosegmentation can be roughly divided into two classes. The first class of methods employ local features, such as [7–9, 14, 17, 19, 21], where image features such as color histogram, SIFT, Fisher vectors etc. are extracted at all the pixels (or superpixels), and pixels (or superpixels) with similar features are encouraged to share the same segmentation results. One potential problem with the image features is that they may be too local to be distinctive, so they may not provide strong prior information for segmentation. In contrast, the explicit shape templates employed by our method cover much larger area (100×100) and capture much larger and distinctive patterns, so that cosketch by these templates help to establish the correspondence between different images.

The second class of methods, such as [2, 22] and our method, employ explicit models for the sketchable patterns. In [22], the edge model is defined by Gaussian distributions over Canny edge strength transformed by a deformation field. In [2], shape model is in the form of a rigid energy map covering regions determined by salient object detector. Both algorithms are only tested on images with roughly aligned object instances. In contrast, our unsupervised learning method can be effectively applied to non-aligned images where the common object instances can appear at different locations, orientations and scales.

Strongly supervised segmentation is another popular topic in image segmentation, where training images with annotated ground truth are used to train generic segmentation model [5, 11, 15, 18] or to perform segmentation propagation [10]. In [11, 15], template-based models capturing high-level shape cues are trained from the aligned training images. However, unlike our method, these methods do not work with the scenario of cosegmentation where the ground truth annotations are not available.

This work is also related to [1, 13], where repeated sketchable patterns are learned. Unlike our method, they do not deal with the problem of segmentation.

3. Model

For clarity, we first present the simplest form of the model and algorithm. Implementation issues for the general situation will be treated at the end of Section 4.

3.1. Notation and problem definition

Let $\mathbf{I}_m$, $m = 1, ..., M$ be a set of *multiple* input images. Let $\mathcal{D}_m$ be image domain of $\mathbf{I}_m$, i.e., $\mathcal{D}_m$ collects

all the pixels of $\mathbf{I}_m$. For each pixel $x \in D_m$ (x is a two-dimensional coordinate in $\mathcal{D}_m$), let $\delta_m(x)$ be the label of pixel x for image segmentation, so that $\delta_m(x) = 1$ if x belongs to foreground, and $\delta_m(x) = 0$ if x belongs to background. The task of cosementation is to take multiple images $\{\mathbf{I}_m, m = 1, ..., M\}$ as input, and return the label maps $\{(\delta_m(x), x \in D_m), m = 1, ..., M\}$ as output.

In the sketch model, $\mathbf{I}_m(x)$ is assumed to be a grey level intensity. In the region model, $\mathbf{I}_m(x)$ is assumed to be a three-dimensional vector in the color space.

3.2. Sketch model

The sketch model consists of a codebook of shape templates. Each template is represented by an active basis model [6, 23], which is a composition of Gabor wavelets at selected locations, scales and orientations. In Fig. 1, each selected Gabor wavelet is shown by a bar, and these bars illustrate the shape templates. Specifically, let $B_{x,s,\alpha}$ denote a Gabor wavelet (or in general, a basis function) centered at pixel x and tuned to scale s and orientation α. An active basis template is in the form of $\mathbf{B} = (B_{x_i,s,\alpha_i}, i = 1, ..., n)$, where the constituent basis functions are allowed to perturb their locations and orientations while the scale s is fixed.

Preparation: Aligned images and a single template. Let us temporarily assume that $\{\mathbf{I}_m\}$ are defined on the same image domain, i.e., $\mathcal{D}_m = \mathcal{D}$ is the same for $m = 1, ..., M$. Let us also assume that these images are aligned so that objects in these images can be represented by a single shape template with $\mathcal{D}$ being its bounding box (the bounding box of a template in this article is 100×100 pixels). The active basis model then assumes the following form:

$$\mathbf{I}_m = \sum_{i=1}^{n} c_{m,i} B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}} + U_m, \qquad (1)$$

where U_m is the unexplained residue image, $\mathbf{B} = (B_{x_i,s,\alpha_i}, i = 1, ..., n)$ form the nominal template of an active basis model (the number of basis functions n in our work is fixed at 40). $\mathbf{B}_m = (B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}, i = 1, ..., n)$ is the deformed version of the nominal template $\mathbf{B}$ for encoding $\mathbf{I}_m$, where $(\Delta x_{m,i}, \Delta\alpha_{m,i})$ are the perturbations of the location and orientation of the i-th basis function. The perturbations are introduced to account for shape deformations. Both $\Delta x_{m,i}$ and $\Delta\alpha_{m,i}$ are assumed to vary within limited ranges (default setting: $\Delta x_{m,i} \in [-3, 3]$ pixels, and $\Delta\alpha_{m,i} \in \{-1, 0, 1\} \times \pi/16$).

For the convenience of stochastic modeling and for the efficiency of computation, we assume that $(B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}, i = 1, ..., n)$ are orthogonal to each other, so that the coefficient $c_{m,i} = \langle \mathbf{I}_m, B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}} \rangle$ is a deterministic transform extracted from $\mathbf{I}_m$.

For statistical modeling, let $p(\mathbf{I}_m \mid \mathbf{B}_m)$ be the distribution of $\mathbf{I}_m$ given the deformed template $\mathbf{B}_m = (B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}, i = 1, ..., n)$. Let $q(\mathbf{I}_m)$ be a reference distribution, such as the distribution of natural images. The active basis model assumes the following form:

$$\frac{p(\mathbf{I}_m \mid \mathbf{B}_m)}{q(\mathbf{I}_m)} = \prod_{i=1}^{n} \frac{1}{Z(\lambda_i)} \exp\{\lambda_i h(|c_{m,i}|^2)\} \qquad (2)$$

where $(c_{m,i}, i = 1, ..., n)$ are assumed to be independent under both $p(\mathbf{I}_m \mid \mathbf{B}_m)$ and $q(\mathbf{I}_m)$. For $r = |c_{m,i}|^2$ (a Gabor wavelet may consist of a pair of sine and cosine components, so r is the sum of squares of the responses from the two components), $h(r) = \xi[2/(1 + e^{-2r/\xi}) - 1]$, so $h(r) \approx r$ for small r, and $h(r) \to \xi$ as $r \to \infty$ (default setting: $\xi = 6$). $Z(\lambda) = \mathrm{E}_q[\exp\{\lambda h(r)\}]$ is the normalizing constant, which is computed from natural images.

Shared matching pursuit algorithm. This algorithm is used to learn the active basis model from aligned $\{\mathbf{I}_m\}$. At the i-th iteration, the algorithm selects B_{x_i,s,α_i} and estimates the associated λ_i by seeking the maximal increase of the likelihood. Specifically, for each $\mathbf{I}_m$, we initialize the response maps $R_m(x, \alpha) \leftarrow \langle \mathbf{I}_m, B_{x,s,\alpha} \rangle$ for all (x, α). Then in the i-th iteration, we select

$$(x_i, \alpha_i) = \arg\max_{x,\alpha} \sum_{m=1}^{M} \max_{\Delta x,\Delta\alpha} h(|R_m(x+\Delta x, \alpha+\Delta\alpha)|^2), \qquad (3)$$

where $\max_{\Delta x,\Delta\alpha}$ is local maximum pooling within the perturbation range. After that, for each $\mathbf{I}_m$, we infer the perturbations $(\Delta x_{m,i}, \Delta\alpha_{m,i})$ by retrieving the arg-max in the above local maximum pooling, and let the arg-max basis function $B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}$ inhibit those $B_{x,s,\alpha}$ whose squared correlation with $B_{x_i+\Delta x_{m,i},s,\alpha_i+\Delta\alpha_{m,i}}$ exceeds a tolerance value (default tolerance = .1). For such $B_{x,s,\alpha}$, we set the corresponding $R_m(x, \alpha) = 0$. The associated λ_i is estimated by maximum likelihood. We then select the next basis function and repeat this process until n basis functions are selected. See [23] for more details.

Our situation: Non-aligned images and a codebook of templates. Now suppose that $\{\mathbf{I}_m\}$ are not aligned, and we want to encode the sketchable parts of $\{\mathbf{I}_m\}$ by a codebook of templates $\{\mathbf{B}^{(t)}, t = 1, ..., T\}$ (default: $T = 4$). Each template $\mathbf{B}^{(t)} = (B_{x_i^{(t)},s,\alpha_i^{(t)}}, i = 1, ..., n)$ is associated with parameters $\Lambda^{(t)} = (\lambda_i^{(t)}, i = 1, ..., n)$. Define $\Theta_S = (\mathbf{B}^{(t)}, \Lambda^{(t)}, t = 1, ..., T)$ to be the sketch model parameter.

For each $\mathbf{I}_m$, suppose we encode $\mathbf{I}_m$ by K templates which are spatially translated instances of templates in the codebook. For now, let us assume that these K templates do not overlap with each other. The issues of overlap as well as rotation and scaling of the templates will be considered later, which do not add anything conceptually.

For template $\mathbf{B}^{(t)}$, let $\mathbf{B}^{(t)}_X = (B_{X+x_i^{(t)},s,\alpha_i^{(t)}}, i = 1, ..., n)$ be the template obtained by spatially translating $\mathbf{B}^{(t)}$ to X. Suppose $\mathbf{I}_m$ is encoded by $(\mathbf{B}^{(t_{m,k})}_{X_{m,k}}, k = 1, ..., K)$. We define $W^S_m = (\mathbf{B}^{(t_{m,k})}_{X_{m,k}}, k = 1, ..., K)$ to be the sketch representation of $\mathbf{I}_m$. Then the log-likelihood ratio is the sum of the log-likelihood ratios of the K templates,

$$l(\mathbf{I}_m \mid W^S_m) = \sum_{k=1}^{K} l\left(\mathbf{I}_m \mid \mathbf{B}^{(t_{m,k})}_{X_{m,k}}\right), \tag{4}$$

where the log-likelihood ratio or the template matching score of $\mathbf{B}^{(t)}_X$ on $\mathbf{I}_m$ is

$$l\Big(\mathbf{I}_m \mid \mathbf{B}^{(t)}_X\Big) = \sum_{i=1}^{n}\Big[\lambda_i^{(t)} \max_{\Delta x,\Delta\alpha} h\big(|\langle \mathbf{I}_m, B_{X+x_i^{(t)}+\Delta x,s,\alpha_i^{(t)}+\Delta\alpha}\rangle|^2\big) - \log Z\big(\lambda_i^{(t)}\big)\Big]. \tag{5}$$

Energy function for sketch model. We define the energy function of the sketch model to be

$$\mathcal{E}(\mathbf{I}_m \mid W^S_m, \Theta_S) = -l(\mathbf{I}_m \mid W^S_m). \tag{6}$$

3.3. Region model

The region model generates non-sketchable visual patterns, by modeling the marginal distributions of $\mathbf{I}_m(x)$ (here $\mathbf{I}_m(x)$ is a three-dimensional vector in the color space), and the pairwise similarities between neighboring pixels, conditioning on pixel labels for segmentation. The energy function of the region model is in the form of pair-potential Markov random field. It consists of two terms: the unary potential and the pairwise potential.

Unary potential. The unary potential models the marginal distribution of pixel colors conditional on the pixel labels by mixtures of Gaussian distributions. Let $g(v; \mu, \Sigma)$ denote a three-dimensional Gaussian density function with mean μ and variance-covariance matrix Σ, and ρ denote the prior of a Gaussian density function within the mixture model, the unary potential is as

$$\begin{aligned}&\phi_1(\mathbf{I}_m(x)|\delta_m(x))\\ &= -\log\Big[\sum_{c=1}^{C}\rho^{(0)}_{\delta_m(x),c} g(\mathbf{I}_m(x); \mu^{(0)}_{\delta_m(x),c}, \Sigma^{(0)}_{\delta_m(x),c})\\ &+ \sum_{c=1}^{C}\rho^{(m)}_{\delta_m(x),c} g(\mathbf{I}_m(x); \mu^{(m)}_{\delta_m(x),c}, \Sigma^{(m)}_{\delta_m(x),c})\Big],\end{aligned} \tag{7}$$

where $\theta^{(0)}_R = (\rho^{(0)}_{\delta,c}, \mu^{(0)}_{\delta,c}, \Sigma^{(0)}_{\delta,c})$ is a generic color model shared by all input images, $\theta^{(m)}_R = (\rho^{(m)}_{\delta,c}, \mu^{(m)}_{\delta,c}, \Sigma^{(m)}_{\delta,c})$ is an image specific color model. As a commonly used approximation, the sum operation in (7) can be replaced by max operation. The default value of C is set to be 5.

Pairwise potential. If pixels x and y are nearest neighbors as denoted by $x \sim y$, then we want $\mathbf{I}_m(x)$ and $\mathbf{I}_m(y)$ to be different from each other if x and y belong to different regions. The pairwise potential is defined as

$$\begin{aligned}&\phi_2(\mathbf{I}_m(x), \mathbf{I}_m(y)|\delta_m(x), \delta_m(y))\\ &= \mathbf{1}(\delta_m(x) \neq \delta_m(y)) \exp\left[-\frac{\|\mathbf{I}_m(x) - \mathbf{I}_m(y)\|_2^2}{2\sigma^2}\right].\end{aligned} \tag{8}$$

where $\mathbf{1}()$ is the indicator function, $\|\cdot\|_2^2$ denotes the squared ℓ_2 distance between the colors of neighboring pixels, and σ^2 is taken to be the mean squared distance between neighboring pixels.

Energy function for region model. Define $W^R_m = (\delta_m(x), x \in \mathcal{D}_m)$ to be the region representation of $\mathbf{I}_m$. Define $\Theta_R = (\theta^{(0)}_R, \theta^{(m)}_R, \forall m)$ to be the parameters of the region model. The energy function for the region model is

$$\begin{aligned}\mathcal{E}(\mathbf{I}_m|W^R_m, \Theta_R) &= \sum_x \phi_1(\mathbf{I}_m(x)|\delta_m(x))\\ &+ \sum_{x\sim y}\phi_2(\mathbf{I}_m(x), \mathbf{I}_m(y)|\delta_m(x), \delta_m(y)).\end{aligned} \tag{9}$$

3.4. Coupling sketch and region models

The generative model that involves both the sketch model and the region model can be written as: $P(W^S_m, W^R_m)P(\mathbf{I}_m \mid W^S_m, W^R_m)$. The prior model can be factorized into $P(W^S_m, W^R_m) = P(W^S_m)P(W^R_m|W^S_m)$, where $W^R_m = (\delta_m(x), x \in \mathcal{D}_m)$ consists of pixel labels, and $W^S_m = (\mathbf{B}^{(t_{m,k})}_{X_{m,k}}, k = 1, ..., K)$ consists of selected templates. We couple them by modeling $P(W^R_m|W^S_m)$, where the templates provide prior for pixel labels.

Segmentation templates as probability maps. For the codebook of template $\{\mathbf{B}^{(t)}, t = 1, ..., T\}$, we associate a segmentation template with each $\mathbf{B}^{(t)}$. Specifically, let $\mathcal{D}^{(t)}$ be the bounding box of $\mathbf{B}^{(t)}$. We assume that $\mathcal{D}^{(t)}$ is centered at origin. The segmentation template is in the form of a probability map $\mathbf{P}^{(t)}$ defined on $\mathcal{D}^{(t)}$, so that for each $x \in \mathcal{D}^{(t)}$, $\mathbf{P}^{(t)}(x, \delta) = \Pr(\delta(x) = \delta)$, where $\delta(x) = 1$ if x belongs to the foreground, and $\delta(x) = 0$ otherwise.

If we spatially translate $\mathbf{B}^{(t)} = (B_{x_i^{(t)},s,\alpha_i^{(t)}}, \forall i)$ to $\mathbf{B}^{(t)}_X = (B_{X+x_i^{(t)},s,\alpha_i^{(t)}}, \forall i)$, then we also translate the bounding box $\mathcal{D}^{(t)}$ to $\mathcal{D}^{(t)}_X = \{X + x, x \in \mathcal{D}^{(t)}\}$. For each $x \in \mathcal{D}^{(t)}_X$, $\Pr(\delta(x) = \delta) = \mathbf{P}^{(t)}(x - X, \delta)$. Therefore, given $W^S_m = (\mathbf{B}^{(t_{m,k})}_{X_{m,k}}, \forall k)$, we have the prior probabilities of $W^R_m = (\delta_m(x), x \in \mathcal{D}_m)$.

Coupling energy function. Let $\Theta_C = (\mathbf{P}^{(t)}_x(\delta), t = 1, ..., T, x \in \mathcal{D}^{(t)}, \delta \in \{0, 1\})$ be the segmentation tem-

plates, we define the coupling energy

$$\mathcal{E}(W_m^R|W_m^S,\Theta_C) = -\sum_{k=1}^{K}\sum_{x\in\mathcal{D}_{X_{m,k}}^{(t_{m,k})}} \log \mathbf{P}^{(t_{m,k})}(x - X_{m,k}, \delta_m(x)). \quad (10)$$

Combined energy function. Let $W_m = (W_m^R, W_m^S)$, and let $\Theta = (\Theta_R, \Theta_S, \Theta_C)$. The combined energy function is:

$$\mathcal{E}(\mathbf{I}_m, W_m|\Theta) = \gamma\mathcal{E}(\mathbf{I}_m|W_m^S,\Theta_S) + \mathcal{E}(\mathbf{I}_m|W_m^R,\Theta_R) + \mathcal{E}(W_m^R \mid W_m^S,\Theta_C). \quad (11)$$

Here we introduce a weighting parameter γ because the sketch model is a sparse model with n (default: $n = 40$) basis functions, whereas the region model and the coupling model are dense models defined on all the pixels (default: the size of $\mathbf{P}^{(t)}$ is 100×100). The parameter γ is introduced to balance these two terms (default: $\gamma = 100$). One may consider that $\mathcal{E}(\mathbf{I}_m, W_m|\Theta)$ defines a joint probability via the Gibbs distribution: $P(\mathbf{I}_m, W_m \mid \Theta) = \exp\{-\mathcal{E}(\mathbf{I}_m, W_m|\Theta)/\gamma\}/Z(\Theta)$, where $Z(\Theta)$ is the normalizing constant.

4. Learning algorithm

The input of the learning algorithm is $\{\mathbf{I}_m\}$. The output includes $\{W_m = (W_m^S, W_m^R), \forall m\}$ and $\Theta = (\Theta_S, \Theta_R, \Theta_C)$. The cosegmentation results are $\{W_m^R\}$.

The unsupervised learning algorithm seeks to minimize the total energy function $\sum_m \mathcal{E}(\mathbf{I}_m, W_m|\Theta)$ over $\{W_m\}$ and Θ. The algorithm iterates the following two steps. (I) Image parsing: Given Θ, infer W_m for each $\mathbf{I}_m$. (II) Re-learning: Given $\{W_m, \forall m\}$, estimate Θ.

4.1. Image parsing

The image parsing step can be further divided into two sub-steps. (I.1) Sketch-guided segmentation: Given W_m^S, infer W_m^R. (I.2) Segmentation-assisted sketch: Given W_m^R, infer W_m^S. An illustration of the image parsing algorithm is shown in Fig. 2. The issue of overlap between templates will be discussed at the end of this section.

I.1: Sketch-guided segmentation. This step minimizes

$$\begin{aligned}&\mathcal{E}(\mathbf{I}_m|W_m^R,\Theta_R) + \mathcal{E}(W_m^R \mid W_m^S,\Theta_C)\\ &= \Big[\sum_x \phi_1(\mathbf{I}_m(x)|\delta_m(x))\\ &\quad - \sum_{k=1}^{K}\sum_{x\in\mathcal{D}_{X_{m,k}}^{(t_{m,k})}} \log \mathbf{P}^{(t_{m,k})}(x - X_{m,k}, \delta_m(x))\Big] \quad (12)\\ &\quad + \sum_{x\sim y} \phi_2(\mathbf{I}_m(x), \mathbf{I}_m(y)|\delta_m(x), \delta_m(y))\end{aligned}$$

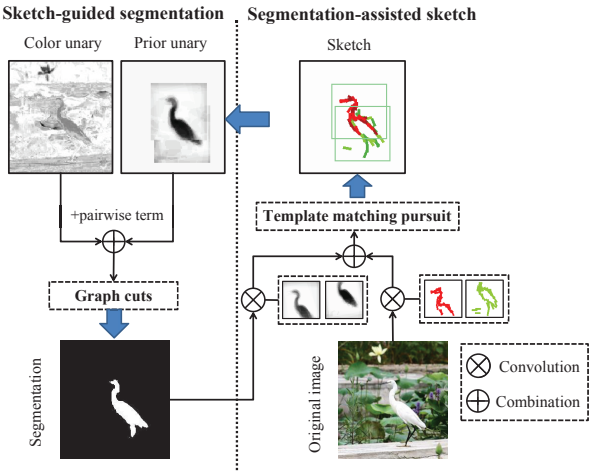

Figure 2. Image parsing by sketch-guided segmentation and segmentation-assisted sketch. Sketch result helps to locate the foreground objects and provides top-down prior information for segmentation. Conversely, segmentation result provides bottom-up information for sketch.

over W_m^R. The energy function is in the form of a unary term and a pairwise term, which satisfies the submodular condition and can be efficiently optimized by graph cuts [4]. The sketch representation generates the prior distribution of pixel labels and adds to the unary term of the energy function of the region model $\mathcal{E}(\mathbf{I}_m|W_m^R,\Theta_R)$.

I.2: Segmentation-assisted sketch. This step minimizes

$$\begin{aligned}&\gamma\mathcal{E}(\mathbf{I}_m|W_m^S,\Theta_S) + \mathcal{E}(W_m^R \mid W_m^S,\Theta_C)\\ &= -\sum_{k=1}^{K}\Big[\gamma l(\mathbf{I}_m \mid \mathbf{B}_{X_{m,k}}^{(t_{m,k})})\\ &\quad + \sum_{x\in\mathcal{D}_{X_{m,k}}^{(t_{m,k})}} \log \mathbf{P}^{(t_{m,k})}(x - X_{m,k}, \delta_m(x))\Big] \quad (13)\end{aligned}$$

over W_m^S. We scan each pair of shape and segmentation templates $(\mathbf{B}^{(t)}, \mathbf{P}^{(t)})$ over $\mathbf{I}_m$ and its label map $(\delta_m(x), x \in \mathcal{D}_m)$ to get the combined template matching score:

$$\mathbf{R}_m^{(t)}(X) = \gamma l(\mathbf{I}_m \mid \mathbf{B}_X^{(t)}) + \sum_{x\in\mathcal{D}_X^{(t)}} \log \mathbf{P}^{(t)}(x - X, \delta_m(x)). \quad (14)$$

Template matching pursuit algorithm. This algorithm sequentially selects templates from the codebook to sketch $\mathbf{I}_m$ based on the maps of the combined score $\mathbf{R}_m^{(t)}(X)$. Specifically, at the k-th iteration, we select the k-th template by finding the global maximum $(X_{m,k}, t_{m,k}) = \arg\max_{X,t} \mathbf{R}_m^{(t)}(X)$. Then we let the selected template $\mathbf{B}_{X_{m,k}}^{(t_{m,k})}$ suppress overlapping templates $\mathbf{B}_X^{(t)}$ by modifying

$\mathbf{R}_m^{(t)}(X) \leftarrow -\infty$. We then select the next template until K templates are selected.

When performing cosegmentation on multiple images, we further require that each template in the codebook can only be used once for each image. So $K = T$. When performing cosegmentation on images with repetitive patterns, we do not impose such requirement, and we choose K adaptively for each image by stopping the template matching pursuit algorithm when all $\mathbf{R}_m^{(t)}(X)$ are less then a prespecified threshold (default threshold = 0).

4.2. Re-learning

This step seeks to minimize the total energy function $\sum_m \mathcal{E}(\mathbf{I}_m, W_m|\Theta)$ over Θ_S, Θ_R and Θ_C given $\{W_m^S\}$ and $\{W_m^R\}$. These three parameters are decoupled so the minimizations can be carried out separately.

II.1: Re-learn shape templates. For each $t = 1, ..., T$, we re-learn $\mathbf{B}^{(t)}$ from all the image patches that are currently covered by $\mathbf{B}^{(t)}$. Specifically, for image $\mathbf{I}$, let $\mathbf{I}(\mathcal{D})$ be the image patch of $\mathbf{I}$ within set $\mathcal{D}$. Then we re-learn $\mathbf{B}^{(t)}$ from the aligned image patches $\{\mathbf{I}_m(\mathcal{D}_{X_{m,k}}^{(t_{m,k})}), t_{m,k} = t, \forall k, m\}$ by the shared matching pursuit algorithm in subsection 3.2.

II.2: Re-learn marginal distributions of regions. For foreground and background, fit the corresponding mixture of Gaussian distributions using the EM algorithm.

II.3: Re-learn segmentation templates. The probability map $\mathbf{P}^{(t)}$ associated with each $\mathbf{B}^{(t)}$ is learned from the pixel labels of all the aligned image patches $\{\mathbf{I}_m(\mathcal{D}_{X_{m,k}}^{(t_{m,k})}), t_{m,k} = t, \forall k, m\}$ explained by $\mathbf{B}^{(t)}$:

$$\mathbf{P}^{(t)}(x, \delta) = \frac{\sum_{m,k} \mathbf{1}(\delta_m(x + X_{m,k}) = \delta)\mathbf{1}(t_{m,k} = t)}{\sum_{m,k} \mathbf{1}(t_{m,k} = t)}. \quad (15)$$

Initialization. For Θ_S, $\mathbf{B}^{(t)}$ and the associated $\Lambda^{(t)}$ are learned from randomly cropped image patches. For Θ_R, the marginal distribution of background is learned from pixels within 10 pixels (default) from the boundary. The marginal distribution of foreground is learned from pixels covered by the aforementioned random patches. The label maps are then initialized by graph cuts. For Θ_C, $\mathbf{P}^{(t)}$ is learned from the label maps of aforementioned random patches.

4.3. Implementation issues

The model and algorithm presented so far are of simplest prototype form, where the templates do not overlap and are only subject to spatial translation. In practical implementations, it is desirable to allow limited overlaps between the selected templates so that we do not miss important structures in the images. It is also desirable to scan the templates over images at multiple resolutions to account for scale variation. In addition, we should allow the templates to undergo rotation and mirror reflection.

Overlap. In the template matching pursuit algorithm, a selected template only inhibits nearby candidate templates with significant overlapping, instead of all overlapping templates. In sketch-guided segmentation, the prior probability of a pixel covered by multiple overlapping segmentation templates is determined by the one with the highest template matching score.

Resolution. In template matching pursuit, we scan $(\mathbf{B}^{(t)}, \mathbf{P}^{(t)})$ over multiple resolutions of $\mathbf{I}_m$ and its label map $(\delta_m(x), x \in \mathcal{D}_m)$ (default: we use three resolutions, which are .8, 1, 1.2 relative to the original image). After that, we map the selected shape and segmentation templates back to the original or the highest resolution, and perform inhibition and image segmentation at this resolution.

In addition, we also allow the templates to rotate (default range: $\{-2, -1, 0, 1, 2\} \times \pi/16$) and to mirror reflect.

5. Experiments

5.1. Cosegmentation on MSRC and iCoseg

The MSRC [20] and iCoseg [3] datasets are widely used by previous work to evaluate co-segmentation performace. In both datasets, instances are of varying appearances, locations, deformations and in cluttered backgrounds. There have been different evaluation protocols employed by different cosegmentation algorithms. Here for clarity and fair comparison, we use all the images of the major object categories in both datasets to avoid bias, and compare with the unsupervised cosegmentaion algorithms without interactive input or additional annotated training images. As for evaluation criterion, we follow the evaluation protocols employed by two recent state-of-the-art methods applied to the two datasets respectively.

For experiments on the MSRC dataset, we use all the images in 14 well defined main object categories, which is the same as in [8]. The pixels corresponding to main objects in each image are deemed as foreground, while the rest pixels are treated as background. Segmentation performance is measured by the intersection-of-union score following [8], which is defined as $\frac{1}{M}\sum_{m=1}^{M} \frac{GT_m \cap R_m}{GT_m \cup R_m}$, where GT_m is the ground truth and R_m is the segmented region of foreground. The results of the proposed approach, Joulin et al. [8], Kim et al. [9], Mukherjee et al. [14] and Joulin et al. [7] are presented in Table 1, in which the results for [7–9, 14] are taken from Table 1 in [8]. The results show that our proposed approach surpasses the other methods in 13 out of 14 categories. And it achieves an average accuracy of 63.0%, which is higher than existing methods by a clear margin.

The iCoseg dataset [3] contains 643 images separated into 38 object categories (e.g. kites, pyramids, hot balloons etc.). Experiments are conducted on all the images of the 38 object categories. Segmentation accuracy is measured by the ratio of correctly labeled pixels of foreground and back-

Table 1. Intersection-over-union scores of the proposed approach and the methods in [7–9, 14] on the MSRC dataset. The results of [7–9, 14] are taken from Table 1 in [8].

Images	Class	Ours	[8]	[9]	[14]	[7]
30	Bike	**51.1**	43.3	29.9	42.8	42.3
30	Bird	**51.2**	47.7	29.9	-	33.2
30	Car	**63.7**	59.7	37.1	52.5	59.0
24	Cat	**61.0**	31.9	24.4	5.6	30.1
30	Chair	**56.1**	39.6	28.7	39.4	37.6
30	Cow	**69.9**	52.7	33.5	26.1	45.0
26	Dog	**63.8**	41.8	33.0	-	41.3
30	Face	55.6	**70.0**	33.2	40.8	66.2
30	Flower	**68.8**	51.9	40.2	-	50.9
30	House	**70.8**	51.0	32.2	66.4	50.5
30	Plane	**46.5**	21.6	25.1	33.4	21.7
30	Sheep	**75.2**	66.3	60.8	45.7	60.4
30	Sign	**73.3**	58.9	43.2	-	55.2
30	Tree	**74.3**	67.0	61.2	55.9	60.0
	Average	**63.0**	50.2	36.6	40.9	46.7

Table 2. Correctly labeled pixel ratios of the proposed approach and the methods in [7, 19, 21] on the iCoseg dataset. The results of [7, 19, 21] are taken from Table 1 in [19]. The method in [21] utilizes an additional annotated dataset for training.

	Ours	[19]	[7]	[21]
Average	**89.5**	83.9	78.9	85.3

ground with respect to the total number of pixels, following the criterion in [19]. The average accuracies of the proposed approach, two recent unsupervised methods in [19] and [7] are presented in Table 2. We also reported the performance of the method in [21], which trained model parameters on an additional annotated dataset. The results of [7, 19, 21] are taken from Table 1 in [19]. The experiment results show that the proposed approach achieves an average accuracy of 89.5%, which is 5.6%, 10.6% and 4.2% higher than the methods in [19], [7] and [21] respectively.

Figure 3 shows some learned models and the corresponding parsing results on the MSRC and iCoseg datasets. It can be seen that our proposed approach can effectively perform cosegmentation and cosketch despite that the object instances in the images are of varying appearances, locations, deformations and in cluttered backgrounds.

5.2. Cosegmentation on Coseg-Rep

To further test our method, we collected a new dataset called Coseg-Rep, which has 23 object categories with 572 images. [1] Among them, 22 categories are different species of animals and flowers, and each category has 9 to 49 images. More important, there is a special category called "repetitive", which contains 116 natural images where similar shape patterns repeat themselves within the same image, such as tree leaves and grapes etc. Segmentation of a single image with repetitive patterns is an important step for applications like automatic leaves recognition [12]. Cosegmentation results of our proposed approach are presented in Table 3. The mean accuracies are 67.4% and 90.2% when evaluated by the intersection-of-union score and the correctly labeled pixel ratio respectively. Fig. 4 shows the learning procedure on a single image with repetitive patterns. Meaningful templates and satisfactory parsing results can be obtained although the algorithm starts from random initialization. As a comparison, our method gives more accurate segmentation result than a Grabut [16] baseline method where the bounding box is set to be 10 pixels away from the boundary. Fig. 3 presents more parsing results on the Coseg-Rep dataset and some failure examples.

[1]The dataset, code and a demo can be downloaded from http://www.stat.ucla.edu/~jifeng.dai/research/CosegmentationCosketch.html.

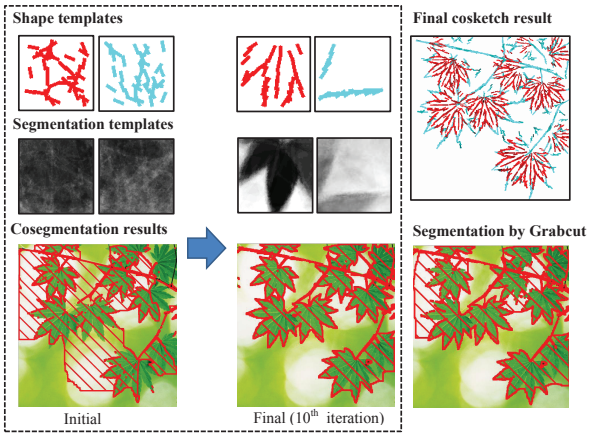

Figure 4. Learned templates and corresponding parsing results in the initial and final iterations of the proposed approach on a single image with repetitive patterns. More accurate segmentation is achieved than the Grabcut [16] baseline.

6. Conclusion

This paper makes the following contributions. (1) We propose a principled model-based unsupervised learning framework for cosegmentation and cosketch. (2) Shape templates and segmentation templates are automatically learned from non-aligned images without ground-truth annotation. (3) We create a new dataset Coseg-Rep for cosegmentation. A special category of the dataset contains natural images with repetitive patterns.

Acknowledgments. The authors thank for the research grants: NSF DMS 1310391, NSF CNS 1028381, ONR MURI N00014-10-1-0933, NSFC 61225008, NSFC 61020106004, MOEC 20120002110033 and China Scholarship Council.

References

[1] N. Ahuja and S. Todorovic. Extracting texels in 2.1D natural textures. In *ICCV*, 2007. 2

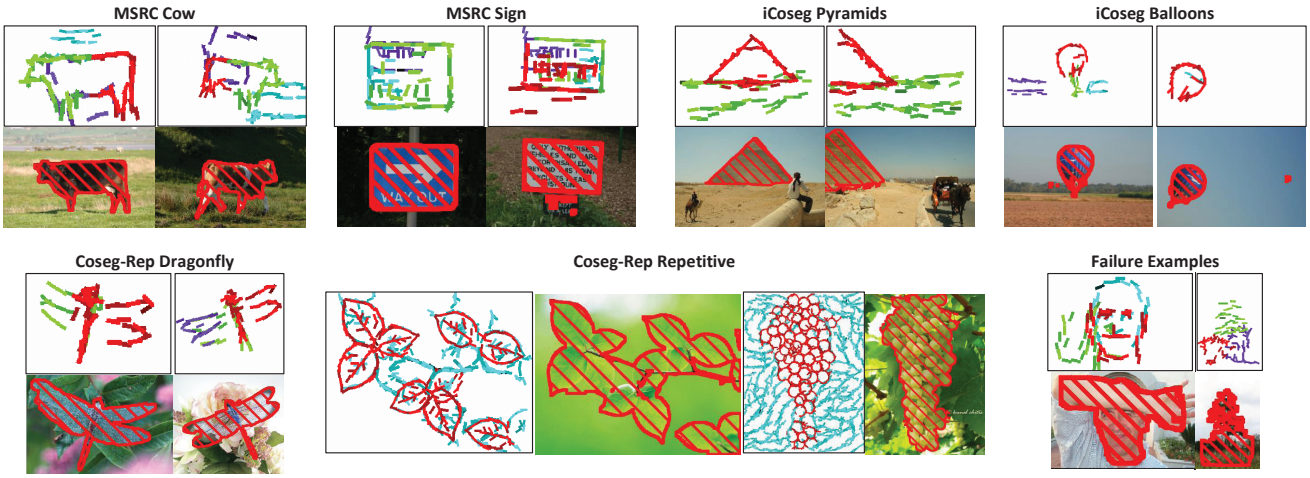

Figure 3. Some cosketch and cosegmentation examples in the MSRC, iCoseg and Coseg-Rep datasets.

Table 3. Intersection-over-union scores (Acc1) and correctly labeled pixel ratios (Acc2) of the proposed approach on the coseg-Rep dataset.

Class	**Repetitive**	Blueflagiris	Camel	Cormorant	Cranesbill	Deer	Desertrose	Dragonfly	Egret	Firepink	Fleabane	Forgetmenot	Frog	Geranium	Ostrich	Pearblossom	Plegon	Seagull	Seastar	Silenecolorata	Snowowl	Whitecampion	Wildbeast	Average
Images	116	10	24	14	18	19	49	14	20	15	19	47	20	33	22	23	19	14	9	15	20	18	14	
Acc1	75.4	89.0	64.1	49.3	84.2	45.0	88.0	38.0	46.3	90.2	88.8	86.7	48.4	89.7	60.5	77.7	42.7	46.4	63.1	83.5	35.5	73.9	83.9	67.4
Acc2	86.2	96.7	89.4	87.6	94.3	83.7	95.3	84.8	92.6	98.0	95.7	94.3	84.5	97.1	91.8	91.3	81.7	87.5	90.2	95.9	69.0	92.9	95.0	90.2

[2] B. Alexe, T. Deselaers, and V. Ferrari. Classcut for unsupervised class segmentation. In *ECCV*, 2010. 2

[3] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. iCoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010. 2, 6

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001. 2, 5

[5] D. Cremers, T. Kohlberger, and C. Schnörr. Nonlinear shape statistics in Mumford-Shah based segmentation. In *ECCV*, 2002. 2

[6] Y. Hong, Z. Si, W. Hu, S.-C. Zhu, and Y. N. Wu. Unsupervised learning of compositional sparse code for natural image representation. *Q. Appl. Math.*, in press. 3

[7] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010. 2, 6, 7

[8] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, 2012. 2, 6, 7

[9] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*, 2011. 2, 6, 7

[10] D. Kuettel, M. Guillaumin, and V. Ferrari. Segmentation propagation in imagenet. In *ECCV*, 2012. 2

[11] M. Kumar, P. H. Torr, and A. Zisserman. Objcut: Efficient segmentation using top-down and bottom-up cues. *PAMI*, 32(3):530–545, 2010. 2

[12] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *ECCV*, 2012. 7

[13] L. Lin, X. Liu, and S.-C. Zhu. Layered graph matching with composite cluster sampling. *PAMI*, 32(8):1426–1442, 2010. 2

[14] L. Mukherjee, V. Singh, and J. Peng. Scale invariant cosegmentation for image groups. In *CVPR*, 2011. 2, 6, 7

[15] B. Packer, S. Gould, and D. Koller. A unified contour-pixel model for figure-ground segmentation. In *ECCV*, 2010. 2

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

[17] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into MRFs. In *CVPR*, 2006. 2

[18] M. Rousson and D. Cremers. Efficient kernel density estimation of shape and intensity priors for level set segmentation. In *MICCAI*, 2005. 2

[19] J. C. Rubio, J. Serrat, A. López, and N. Paragios. Unsupervised co-segmentation through region matching. In *CVPR*, 2012. 2, 7

[20] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006. 2, 6

[21] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, 2011. 2, 7

[22] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *ICCV*, 2005. 2

[23] Y. N. Wu, Z. Si, H. Gong, and S.-C. Zhu. Learning active basis model for object detection and recognition. *IJCV*, 90(2):198–235, 2010. 2, 3