

Distributed Cosegmentation via Submodular Optimization on Anisotropic Diffusion

Gunhee Kim¹ Eric P. Xing¹ Li Fei-Fei² Takeo Kanade¹

¹ School of Computer Science, Carnegie Mellon University

² Computer Science Department, Stanford University

{gunhee, epxing, tk}@cs.cmu.edu feifeili@cs.stanford.edu

Abstract

The saliency of regions or objects in an image can be significantly boosted if they recur in multiple images. Leveraging this idea, cosegmentation jointly segments common regions from multiple images. In this paper, we propose CoSand, a distributed cosegmentation approach for a highly variable large-scale image collection. The segmentation task is modeled by temperature maximization on anisotropic heat diffusion, of which the temperature maximization with finite K heat sources corresponds to a K -way segmentation that maximizes the segmentation confidence of every pixel in an image. We show that our method takes advantage of a strong theoretic property in that the temperature under linear anisotropic diffusion is a submodular function; therefore, a greedy algorithm guarantees at least a constant factor approximation to the optimal solution for temperature maximization. Our theoretic result is successfully applied to scalable cosegmentation as well as diversity ranking and single-image segmentation. We evaluate CoSand on MSRC and ImageNet datasets, and show its competence both in competitive performance over previous work, and in much superior scalability.

1. Introduction

Cosegmentation refers to a procedure that simultaneously segments common regions from multiple images [6, 7, 13, 15]; leveraging an intuition that the saliency of regions or objects in an image can be significantly boosted if they recur in multiple images. Cosegmentation has a wide potential in web-scale applications. For example, it can guide an interactive image editing by suggesting popular regions in the image database [1, 13], or summarize personal photo collections by automatically segmenting highly co-occurring object instances such as persons or dogs [7].

Despite of the promising appeal of cosegmentation, very few algorithms are applicable to web-scale applications,

which require cosegmentation to be not only scalable but also adaptable to heterogeneous images with high variability in content and complexity. In this paper, we address these problems with a new cosegmentation framework, which builds on a solid theoretical ground of submodular optimization, and is readily applicable to large-scale image collection with high variability. Our approach is easily parallelizable; most computations occur independently on individual images, and then an integration step quickly merges all outputs from individual images into a coherent cosegmentation result. We quantitatively show that our approach outperforms state-of-the-arts methods [6, 7] on the MSRC datasets [17]. We also evaluate the scalability of our method on the challenging ImageNet [4]. The magnitude of the dataset sizes in our experiments exceeds those of previous work by an order of magnitude.

The compelling performance and scalability of our approach stem from a novel optimization formulation on the *anisotropic diffusion* (which inspires the name of our algorithm, CoSand, standing for **Co**-Segmentation via **an**isotropic **d**iffusion). The optimization problem underlying CoSand can be summarized in a single sentence as follows; *Given a system under heat diffusion and finite K heat sources, where should one place all the sources in order to maximize the temperature of the system?* In terms of image segmentation, the optimization corresponds to finding the K segment centers that maximize the segmentation confidence of every pixel in the image¹. (e.g. the ideal segmentation is that every pixel has confidence one to be clustered with one of K segment centers). This idea is extended to the cosegmentation problem by constraining the source placements in multiple images to be coupled. This diffusion theoretic optimization framework takes advantage of a strong theoretical property that inspires an efficient computational

¹We use the following terminological correspondences between temperature maximization and image segmentation: *temperature* \equiv *segmentation confidence*, *heat sources* \equiv *segment centers*, *conductance or diffusivity* \equiv *similarity between feature vectors of pixels*.

algorithm. We prove in our paper that, the *temperature*, which is to be optimized in our problem, is a *submodular function* if the system is under linear anisotropic diffusion. A well-known beneficial property of submodular functions is that one can achieve at least a constant factor of the optimal solution by a simple greedy algorithm, which iteratively chooses K locations that maximize marginal temperature gain. Such a greedy solution is particularly promising for cosegmentation tasks on large-scale image collections.

1.1. Relations to Previous work

Submodular optimization: In recent years, submodular optimization has emerged as a useful optimization tool in a variety of machine learning problems such as active learning, structure learning, clustering, and ranking [8, 9]. The submodular function is characterized as a *diminishing return* property that states that, the marginal gain of adding an element to a smaller subset of \mathcal{S} is higher than that of adding it to a larger subset of \mathcal{S} . Some typical submodular functions explored in machine learning include a cut function in a graph and the entropy and the information gain of Gaussian random variables [8].

To the best of our knowledge, our work is the first to address submodular optimization on diffusion in physics².

Cosegmentation: Cosegmentation is the problem of jointly segmenting each of M images into K different regions [6, 7, 11, 13, 15]. Table 1 summarizes the comparison of our work and other unsupervised cosegmentation methods. In summary, our approach is unique in terms of M and K . Most previous work has dealt with binary *figure-ground* segmentation ($K=2$) of small sized image sets (mostly $M=2$ but $M \leq 30$ in [7]). On the other hand, our algorithm is able to perform segmentation of a large-scale dataset with any arbitrary K . We tested with $M \geq 10^3$ images in our experiments, but a more scalable setup is also applicable. The optimization methods for cosegmentation in most previous work, except [7], are based on the graph-cut algorithm. Hence, it is not straightforward and efficient for them to be extended to arbitrary K -way cuts. In theory, the method of [7] can perform cosegmentation with $K > 2$, but it was not evaluated in the paper. On the other hand, our algorithm can attain a constant factor approximation to the optimum with any arbitrary K . The computation time is at worst linear with K .

In addition, our approach also supports the automatic selection of K and robustness against a wrong choice of K . They will be presented in experiments in Section 4, which also reveals that CoSand is compelling in segmentation quality over the state-of-the-art techniques [6, 7] in MSRC [17] and ImageNet [4] datasets.

²*Diffusion* is a heavily overloaded term that is used with different meanings in diverse fields. Here it refers to *diffusion in physics* that is described by a partial differential equation such as heat diffusion or electric current.

Work	Models / Algorithms	M	K
Ours	Diffusion/ Submodularity	$\geq 10^3$	Any
[7]	Discriminative clustering	≤ 30	2
[11]	MRF+ Rank-1 global / Iterative opt.	≤ 20	2
[6]	MRF+Reward global / Graph Cuts	2	2
[13]	MRF+L1 global / Trust Region GC	2	2
[15]	Boykov-Jolly / Dual Decomposition	2	2

Table 1. Comparison with other unsupervised cosegmentation methods. Models and optimization algorithms are summarized. Let M and K denote the number of images and the number of segments, respectively. Most previous work has mainly focused on binary *figure-ground* segmentation of small-sized image sets.

Anisotropic diffusion: The heat diffusion framework that is represented by a partial differential equation has been a successful technique in image processing and computer vision. Notable examples include image segmentation [18], optical flow estimation [2], and image smoothing [16]. In these applications, the temperature corresponds to various objectives, which are the clustering confidence in segmentation, the optical flow in motion analysis, or the RGB value in image smoothing. In this paper, we focus on image segmentation, but our optimization is also easily extendible to those problems such as large-scale edge-preserving image smoothing or layered motion segmentation in video.

1.2. Summary of Contributions

The main contributions of this paper are as follows:

(1) We propose a diffusion-based optimization framework that is applicable to a wide range of computer vision problems. In this paper, we show that our optimization leads to an effective solution to diversity ranking, single-image segmentation, and cosegmentation.

(2) We prove that the *temperature* of a linear anisotropic diffusion system, which corresponds to many important objectives in computer vision tasks, including the cosegmentation score concerned in this paper, is a submodular function. This is a new result that widens the applicability of submodular optimization in computer vision research.

(3) We present CoSand, a distributed cosegmentation exploiting the submodularity of our diffusion-inspired segmental objective. As compared in Table 1, our approach has some unique benefits including compelling performance over previous methods, superior scalability, and a desirable ability of automatically deciding the number of segments.

2. Submodularity and Diffusion

2.1. Optimization on Anisotropic Diffusion

We begin with a general theory of anisotropic diffusion [16]. Let Ω denote the domain of a system and x be a point in $\Omega \in \mathbb{R}^d$ ($x \in \Omega$). Since we are usually interested in discrete systems (*e.g.* images or graphs), let us assume

that Ω is a discrete set of points³. The $u(x, t)$ is the temperature at position x at time t and $D(x)$ is a $d \times d$ positive symmetric tensor called the *diffusion tensor*. The *linearity* of diffusion indicates that D is not a function of u or ∇u . The *anisotropy* means that the flux $-D(x)\nabla u(x, t)$ and the gradient $\nabla u(x, t)$ are not parallel in an image domain. The diffusion equation of such a system is as follows:

$$\frac{\partial u(x, t)}{\partial t} = \text{div}(D(x)\nabla u(x, t)). \quad (1)$$

Our optimization problem is that of maximizing the sum of temperature of the system that is under anisotropic diffusion by choosing the locations of K heat sources. Formally,

$$\begin{aligned} & \max \int_{x \in \Omega} u(x, t) dx \\ & \text{s.t. } \frac{\partial u(x, t)}{\partial t} = \text{div}(D(x)\nabla u(x, t)) \\ & u(g) = 0, u(s) = 1 \text{ for } s \in \mathcal{S} \subset \Omega, |\mathcal{S}| \leq K \end{aligned} \quad (2)$$

where we assume that the temperature of environment (*i.e.* outside of the system Ω) is zero (*i.e.* $u(g) = 0$), and the source temperature is one at any time (*i.e.* $u(s) = 1$)⁴.

For physical analogy, you may imagine a metal plate in open air, and its temperature is to be maximized with K point heat sources. Without loss of generality, we explicitly decompose the heat flux at every point into two parts - a flux within the system and a dissipation flux to out of the system. Let $z(x)$ be a positive scalar diffusivity to the environment at x , and then the dissipation heat loss is $-z(x)(u(x) - u(g))$. If $z(x) = 0$ for $\forall x \in \Omega$, the system is *insulated*. From now on, we assume that $-D(x)\nabla u(x, t)$ solely contributes to the diffusion within the system.

In order to efficiently solve the optimization of Eq.(2) for arbitrary K , we first prove that the temperature under the linear anisotropic diffusion is submodular.

Theorem 1 (Submodularity on Anisotropic Diffusion).

Suppose that the system undergoes *linear anisotropic diffusion*. Let $u(x, t; \mathcal{S})$ be the temperature at position x at time t when identical heat sources are attached to $\mathcal{S}(\subset \Omega)$. Then, the following statements hold for $\forall x \in \Omega, \forall t \in [0, \infty]$.

- (T1) $u(x, t; \emptyset) = 0$
- (T2) $u(x, t; \mathcal{S})$ is *nondecreasing* and *submodular*.

Proof. The proof is shown in the supplementary material.

Let $U(t; \mathcal{S}) = \int_{x \in \Omega} u(x, t; \mathcal{S}) dx$ be the temperature sum of the system at t . Intuitively, $U(t, \mathcal{S})$ is also submodular since it is the sum of submodular functions [8]. Theorem 2 below states that a simple greedy algorithm achieves a near optimal solution for the maximization of a submodular function.

³It is not difficult to obtain the corresponding results of following arguments for the continuous (*i.e.* Ω and t : continuous) and semi-discrete (*i.e.* Ω : discrete, t : continuous) cases [16].

⁴Here we consider only *Dirichlet* boundary conditions.

Theorem 2 ([12]). Let u be a submodular, nondecreasing set function and $u(\emptyset) = 0$. Then, the greedy algorithm finds a set \mathcal{S}_G such that $u(\mathcal{S}_G) \geq C \cdot \max_{|\mathcal{S}| \leq K} u(\mathcal{S})$ where $C = (1 - 1/e) \approx 0.632$.

2.2. Examples: Diversity ranking and clustering

For better understanding of the above diffusion formulation, let us first examine a simple case – diversity ranking in a graph. Diversity ranking [19] aims to re-rank items to reduce redundancy while maintaining their centrality, which is highly relevant to the goal of segmentation. Intuitively, in order to maximize the temperature of the system with limited sources, the sources should be located in the center-of-gravity regions that are densely connected to other elements with high conductivity. Simultaneously, the sources should be sufficiently distant from one another to have a broad and balanced coverage of the system. In the next section, we extend this idea into the cosegmentation problem.

Suppose the following; (1) The system Ω is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. (2) We are interested in the steady state (*i.e.* when $t \rightarrow \infty$), thus we can drop t in our notation. (3) The diffusivity (*i.e.* conductance) is defined by Gaussian similarity between the features of vertices:

$$d_{xy} = \begin{cases} \exp(-\beta \|\mathbf{g}(x) - \mathbf{g}(y)\|^2), & \text{if } (x, y) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $\mathbf{g}(x)$ is the feature vector at node $x \in \mathcal{V}$. (4) The dissipation conductance at a vertex x is constant in time, denoted by z_x . That is, each node x is connected to an environment node g with conductance of z_x . With these assumptions, diffusion reduces to the famous random walk model [5] or Gaussian random fields [20]. The optimization problem in Eq.(2) grounds to a more specific form below⁵:

$$\begin{aligned} & \max \sum_{x \in \mathcal{V}} u(x) \\ & \text{s.t. } u(x) = \frac{1}{a_x} \sum_{(x, y) \in \mathcal{E}} d_{yx} u(y) \text{ for } a_x = \sum_{(x, y) \in \mathcal{E}} d_{yx} + z_x \\ & u(g) = 0, u(s) = 1 \text{ for } s \in \mathcal{S} \subset \mathcal{V}, |\mathcal{S}| \leq K \end{aligned} \quad (4)$$

where a_x is the degree of x . In terms of *random walks*, the optimization of Eq.(4) corresponds to *selecting K nodes as absorbing nodes to maximize the sum of absorbing probabilities of a random walker in a given network \mathcal{G}* . In terms of *linear electric circuits*, the first constraint of Eq.(4) is the *Kirchhoff equation*, and the problem is locating K voltage sources to maximize the electric potential of the circuit.

Since the objective $u(x; \mathcal{S})$ is submodular, we can obtain a near-optimal solution by a greedy algorithm, which starts with an empty \mathcal{S} and iteratively adds the item s_k that maximizes the marginal temperature gain, $U(\mathcal{S}_{k-1} \cup \{s_k\}) -$

⁵Refer to [5, 18] for the derivation from Eq.(2) to Eq.(4).

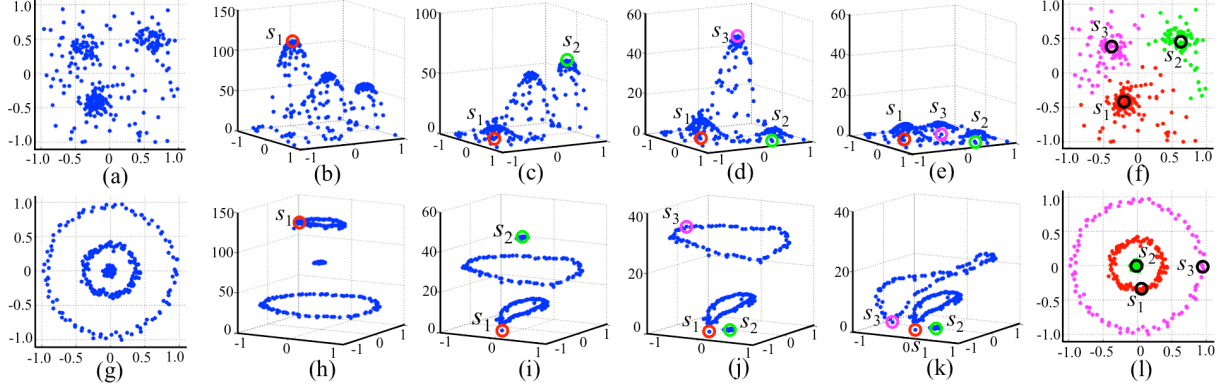


Figure 1. Two toy examples of diversity ranking. The data points are randomly generated from three Gaussian distributions in (a) and three co-centric circles in (g). In (b)-(e) and (h)-(k), the marginal temperature gain of each point $U(\mathcal{S} \cup \{x\}) - U(\mathcal{S})$ is shown along z -axis. $s_k (\in \mathcal{S})$ are iteratively selected by solving Eq.(5). Once a point is selected, the marginal gains of its neighbors largely drop because they already get high temperatures. In (f)(l), final three clusters are shown. The clustering from \mathcal{S} will be discussed in Algorithm 1.

$U(\mathcal{S}_{k-1})$, as shown in Eq.(5). The details of the greedy algorithm will be discussed in Section 3.

$$s_k = \operatorname{argmax}_{x \in \mathcal{V}} U(\mathcal{S}_{k-1} \cup \{x\}) - U(\mathcal{S}_{k-1}) \quad (5)$$

where $U(\mathcal{S}_k) = \sum_{x \in \mathcal{V}} u(x; \mathcal{S}_k)$

The dissipation conductance z is a parameter to control trade-off between centrality and diversity. With a larger z , the heat loss to the environment is larger as well, and only the neighbors within a shorter range of a source will get high temperatures. Hence, a point to be closer to the already ranked set \mathcal{S}_{k-1} is likely to be chosen as a next s_k .

Fig.1 shows two toy examples of diversity ranking and clustering. Here, the location of a point is used as the feature $\mathbf{g}(i)=[x \ y]^T$ to compute the similarity of Eq.(3). Therefore, a closer point pair (i, j) has a larger diffusivity d_{ij} . In the first example of three Gaussian distributions (Fig.1.(a)-(f)), our intuition tells that the center point in the largest blob should be selected as the first item s_1 , and it actually has the highest marginal gain in Fig.1.(b). In the next iteration, since the points near s_1 already have high temperature, the second choice to maximize the marginal gain should be not only distant enough from s_1 (diversity) but also densely linked by other points with high diffusivity d_{ij} (centrality), which is s_2 in Fig.1.(c). In sum, s_k is chosen as the most central but distant enough from already selected items \mathcal{S}_{k-1} .

In the second example of three co-centric circles (Fig.1.(g)-(l)), one interesting behavior is that among the points in each circle, the point at the opposite side of the circle to the selected point has the highest marginal gain. Thus, if the fourth s_4 is chosen in Fig.1.(k), it is the exact opposite of s_3 in the circle. That is, the largest circle in Fig.1.(l) will be divided as two exact half circles with $K=4$.

This algorithm may seem to be similar to the Grasshopper algorithm [19], a greedy algorithm for diversity ranking.

However, the objective function is different, and our main contribution over [19] is that our method is not *ad-hoc*, but a constant-factor approximation based on the submodularity.

3. Large-scale CoSegmentation

In this section, we present our scalable cosegmentation algorithm. Bellow, we begin with the segmentation of a single image to illustrate the basic behavior of the algorithm.

3.1. Segmentation of a Single Image

The segmentation of a single image aims to find K segment centers to maximize the sum of segmentation confidence of every pixel in an image. This can be achieved via the following procedure.

Building the intra-image graph of an image: For faster computational speed, we first extract superpixels from an image as shown Fig.2.(b). Any edge-preserving superpixel methods can be applied, but TurboPixels [10] is used in our implementation. Then we build the intra-image graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{D}_i)$ where the vertex set \mathcal{V}_i is the set of superpixels and the edge set \mathcal{E}_i connects all pairs of adjacent superpixels. Let N_i denote the number of superpixels of an image i . In each superpixel, 3-D CIE Lab color and 4-D texture features⁶ are extracted. The diffusivity \mathcal{D}_i is computed by Gaussian similarity in Eq.(3) on the features of superpixels. The adjacency matrix \mathbf{G}_i of \mathcal{G}_i is a sparse $N_i \times N_i$ matrix, in which the number of nonzero elements of each superpixel is the same with the number of its neighbors. In most cases, it is less than 10.

Construction of evaluation set: In the diversity ranking discussed earlier, we compute the marginal gain at every datapoint to find the maximum (Fig.1). However, this search is inefficient since the actual distinctive regions in

⁶<http://www.robots.ox.ac.uk/~vgg/research/texclass/>.

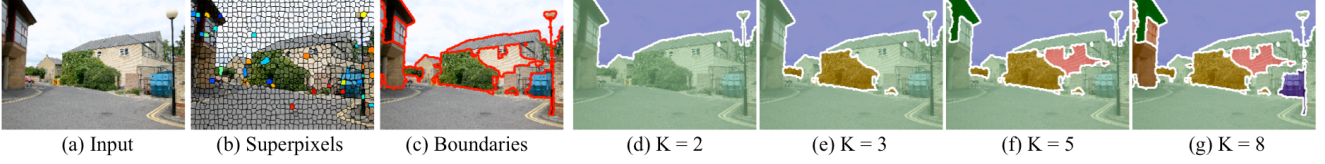


Figure 2. An example of segmenting a single image. (a) An input image. (b) 1000 super-pixels and colored evaluation locations \mathcal{L}_i . (c) Image segmentation with red boundaries. (d)-(g) Color-coded segmentation outputs by ranging K from 2 to 8. As K increases, the following regions are detected in turn: $\{\text{sky, tree, wall (center), roof (left), windows (left), building (left), and trash container}\}$.

an image are usually much fewer than N_i . For example, in Fig.2, there are a lot of *sky* superpixels and there is little difference in the segmentation results no matter which *sky* superpixel is chosen as a segment center. Thus, we first run agglomerative clustering on \mathbf{G}_i to find out the set of evaluation points \mathcal{L}_i . ($|\mathcal{L}_i| \leq 100$ in our experiments). The marginal gain is only computed at \mathcal{L}_i . That is, segment centers are limited to be placed in a subset of \mathcal{L}_i . (i.e. $\mathcal{S}_i \subset \mathcal{L}_i \subset \mathcal{V}_i$ in the third constraint of Eq.(6)). Fig.2.(b) shows an example of \mathcal{L}_i as colored superpixels.

Basic behavior of segmentation: In summary, our segmentation algorithm greedily selects the largest and most coherent regions. As shown in Fig.2.(d), the sky is first chosen with $K=2$. As K increases, the regions of the tree, the house in the center, and the building in the left are chosen in the decreasing order of their sizes and coherence in Fig.2.(d)-(g). This desirable trend comes from the greedy nature of our algorithm. This behavior is quite helpful for automatic selection of K . We can keep increasing K until the detected segment is not significant any more (i.e. temperature increase by adding a new source is not significant any more). As iteration goes, we re-use the previous results of a lower K , which significantly reduce the computation time (e.g. the lazy greedy approach in [9]).

3.2. Scalable Cosegmentation

The input of cosegmentation is an image set \mathcal{I} and the number of segments K . The optimization formulation for cosegmentation in Eq.(6) is an extension of that of the diversity ranking (Eq.(4)).

$$\begin{aligned}
 \max \quad & \sum_{i \in \mathcal{I}} \sum_{x \in \mathcal{V}_i} u_i(x) \\
 \text{s.t.} \quad & u_i(x) = \frac{1}{a_x} \sum_{(x,y) \in \mathcal{E}_i} d_{yx} u_i(y) \quad \text{for } a_x = \sum_{(x,y) \in \mathcal{E}_i} d_{yx} + z_x \\
 & u_i(g) = 0, \quad u_i(s_{ik}) = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} f(\mathbf{g}(s_{ik}), \mathbf{g}(s_{jk})) \\
 & \text{where } s_{ik} \in \mathcal{S}_i \subset \mathcal{L}_i \subset \mathcal{V}_i, |\mathcal{S}_i| \leq K, \quad \text{for } \forall i \in \mathcal{I}.
 \end{aligned} \tag{6}$$

The objective in Eq.(6) is the sum of temperature (i.e. segmentation confidence) of every image in the dataset. Thus, it encourages each image to be segmented as K largest and most coherent regions that are nevertheless

content-wise diverse with respect to one another. In order to enforce inter-image similarity between chosen clusters, the second constraint of Eq.(6) is introduced. The $f(\mathbf{g}(s_{ik}), \mathbf{g}(s_{jk}))$ is an increasing function of the feature affinity between the k -th sources of an image i (s_{ik}) and an image j (s_{jk}). More visually similar the features of s_{ik} and s_{jk} are, a higher value $f(\mathbf{g}(s_{ik}), \mathbf{g}(s_{jk}))$ has. It is intuitive that the system temperature is linear with the source temperature. (e.g. if the source temperature is halved, then the temperatures of all points in the system are halved as well). Hence, the second constraint pushes the k -th source placement of image i to be similar to its corresponding placement in other images of $\mathcal{N}(i)$, which is the image set of i to be jointly cosegmented. If $\mathcal{N}(i) = \mathcal{I} \setminus i$, then each image is cosegmented with respect to all the other images in \mathcal{I} . Meanwhile, the affinity function f controls how strongly the inter-image similarity is imposed. If $f(\mathbf{g}(s_{ik}), \mathbf{g}(s_{jk}))$ is constant, the optimization of Eq.(6) reduces to independent segmentation of each image. Otherwise, if it is a fast increasing function, the inter-image similarity is highly weighted. We use the Gaussian similarity in Eq.(3) for f .

Algorithm 1 presents the greedy algorithm to solve Eq.(6). Note that Algorithm 1 is easily parallelizable. All steps except step 5 can be computed individually in each image. The computation complexity of step 5 is $O(|\mathcal{I}||\mathcal{N}|)^7$.

Once we obtain K source placement \mathcal{S}_i for each image, the segmentation is straightforward. Here we use the method of [5], which is summarized in step 7-8 of Algorithm 1. It first calculates $(N_i - K) \times K$ matrix \mathbf{X} in which $\mathbf{X}(j, k)$ is the probability that a random walker starting at an unselected j -th point (i.e. $x_j \in \mathcal{V}_i \setminus \mathcal{S}_i$) reaches the k -th source points. Then, we cluster the superpixels that share the same source point as the most probable destination.

Fig.3 shows an example of our cosegmentation on three MSRC cow images with $K=4$. Since our algorithm can handle arbitrary K , the brown and black cows and the river in the first image can be detected as individual clusters.

Optimality: The constant factor approximation of our algorithm is guaranteed if the element with the maximum marginal gain is chosen in each round (step 5). In diversity

⁷ In our Matlab implementation, the main independent computation, step 3-4, took about 2 second per image of 1,000 superpixels. Step 5 took about 6-8 minutes for 1000 images with full dependency (i.e. $|\mathcal{I}|=1000$, $|\mathcal{N}|=999$). The other steps took much less than 1 second per image.

Algorithm 1: CoSand Cosegmentation.

Input: (1) Intra-image matrix \mathbf{G}_i for all $I_i \in \mathcal{I}$. (2) Number of segments K . (3) Evaluation set size $|\mathcal{L}|$.
Output: Cluster centers \mathcal{S}_i and segmented images for $I_i \in \mathcal{I}$.

```

1: foreach  $I_i \in \mathcal{I}$  do  $\mathcal{S}_i \leftarrow \emptyset$  end
2: foreach  $I_i \in \mathcal{I}$  do  $\mathcal{L}_i \leftarrow \text{AggloClust}(\mathbf{G}_i, |\mathcal{L}|)$  end
while  $|\mathcal{S}_i| \leq K$  do
  foreach  $I_i \in \mathcal{I}$  do
    foreach  $l_j \in \mathcal{L}_i$  do
      3: Solve  $\mathbf{u} = \mathbf{L}_i \mathbf{u}$  where  $\mathbf{L}_i$  is the Laplacian of  $\mathbf{G}_i$  and  $\mathbf{u}$  is an  $N_i \times 1$  vector with the constraints of  $\mathbf{u}(\{\mathcal{S}_i \cup l_j\}) = 1$  and  $\mathbf{u}(g) = 0$ .
      4: Obtain the gain  $\Delta U_i(l_j) = \|\mathbf{u}\|_1$  ( $l_1$  norm of  $\mathbf{u}$ ).
    end
  end
  5: Solve the energy maximization by belief propagation  $E(l) = \sum_{i \in \mathcal{I}} \Delta U_i(l_i) \left( \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} f(\mathbf{g}(l_i), \mathbf{g}(l_j)) \right)$ .
   $\{s_1, \dots, s_{\mathcal{I}}\} \leftarrow \text{argmax}_{l_1, \dots, l_{\mathcal{I}}} E(l)$ .
  6: foreach  $I_i \in \mathcal{I}$  do  $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup s_i$  end
end
foreach  $I_i \in \mathcal{I}$  do
  7: Compute  $(N_i - K) \times K$  matrix  $\mathbf{X}$  by solving  $\mathbf{L}_u \mathbf{X} = -\mathbf{B}^T \mathbf{I}_s$  where if we let  $\mathcal{X}_i = \mathcal{V}_i \setminus \mathcal{S}_i$ ,  $\mathbf{L}_u = \mathbf{L}_i(\mathcal{X}_i, \mathcal{X}_i)$ ,  $\mathbf{B} = \mathbf{L}_i(\mathcal{S}_i, \mathcal{X}_i)$ , and  $\mathbf{I}_s$  is a  $K \times K$  identity matrix.
  8: A superpixel  $v_j (\in \mathcal{V}_i)$  is clustered  $c_j = \text{argmax}_k \mathbf{X}(j, k)$ .
end

```

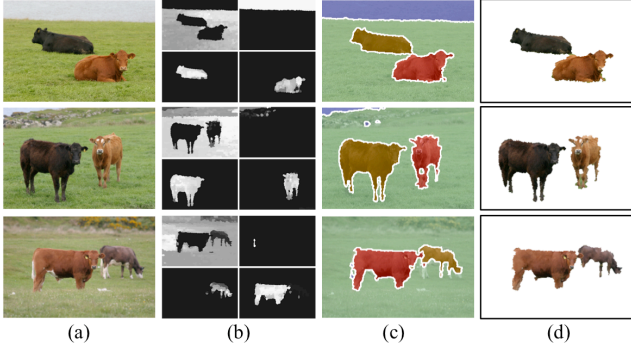


Figure 3. An example of cosegmentation on MSRC cow images ($M=3$, $K=4$). (a) Input images. (b) Likelihood of each segment from white (high) to black (low). (c) Color-coded cosegmentation outputs. (d) The 3rd and 4th segments from input images.

ranking and single-image segmentation, we compute the exact solution for this step. However, we use belief propagation, which is an approximate maximization, for a large-scale cosegmentation with full dependency. In most cases, this relaxed solution is good enough to obtain a high-quality segmentation result.

A more scalable setting: In practice, a large-scale image set is likely to contain various noisy information as well. If heterogeneous images are cosegmented, then the results would be worsen than those of individual image segmentation. Thus, one can first decompose \mathcal{I} into disjoint sets $\mathcal{I} = \mathcal{I}_1 \cup \dots \cup \mathcal{I}_O$ so that each subset \mathcal{I}_o consists of similar images. Then, Algorithm 1 can be applied to each \mathcal{I}_o separately. This decomposition can be done by the proposed

diversity ranking and clustering of Eq.(4) on the similarity graph of \mathcal{I} , which can be constructed by applying Gaussian similarity to image descriptors (e.g. dense SIFT or GIST).

4. Experiments

We evaluate our approach with two different experiments: (1) figure-ground segmentation with a pair of images ($M=2$ and $K=2$), and (2) scalability tests with a large number of images ($M \sim 1000$). The figure-ground tests are performed to quantitatively compare our method with other state-of-the-art cosegmentation techniques that are only applicable in this setting. The scalability tests evaluate how well our algorithm works with real-world data.

Our Matlab toolbox including diversity ranking, single-image segmentation, and cosegmentation, can be found at <http://www.cs.cmu.edu/~gunhee>.

4.1. Evaluation on Figure-ground Cosegmentation

In the figure-ground tests, we use MSRC dataset [17], which provides 30 pixel-wise labeled images per object. Two recent cosegmentation methods, [6] and [7], are compared using their implementation with the default parameter setting⁸. We run [6], [7], and our method on randomly generated 100 pairs in each class.

Unlike the others, the method of [6] requires priori labels of foreground (fg) and background (bg) RGB colors. In order to obtain labels, we first identify the fg and bg regions of each image from the ground truth. Then, we apply K-means to the RGB space of fg and bg pixels to compute three cluster centers each, which are used as labels (i.e. total 6 fg and 6 bg RGB labels in each pair). These labels can be regarded as strong supervision, but they were used because the performance of [6] was highly sensitive to the labels.

Since our method is not designed to aim at figure-ground segmentation, we add an additional step to generate the binary segmentation results. Our approach iteratively chooses large and coherent regions across input images in a bottom-up way. Thus, if the foreground object consists of several distinct regions, it is likely to segment them into multiple regions. For binary segmentation, we first safely cosegment a pair of images with a large K ($K=8$ in our experiments). Then, we apply Normalized cuts to the similarity graph of eight pairs of cosegments to obtain two balanced and discriminative partitions. We observed that our approach showed excellent performance for detecting a moderate number of cosegments but the final figure-ground segmentation accuracy was dependent on this binarization.

Table 2 summarizes the segmentation accuracies on the random test pairs of MSRC dataset. The accuracy is measured by the intersection-over-union metric that is a standard in PASCAL challenges (i.e. For each image, $Ac =$

⁸Codes are available at [6]: <http://www.biostat.wisc.edu/~vsingh/>, [7]: <http://www.di.ens.fr/~joulain/>.

$\frac{GT_i \cap R_i}{GT_i \cup R_i}$). We observed that our method outperformed both [6] and [7] in most objects of the MSRC dataset. Our algorithm was also significantly faster than both competitors; it took less than 10 seconds for a pair of images with a $[320 \times 213]$ dimension, 750 superpixels, and $K=8$.

4.2. Evaluation on Scalable Cosegmentation

For scalability tests, we use ImageNet⁹ [4]. We compute segmentation accuracies by using its bounding box annotations. The bounding boxes may not be a perfect ground truth for segmentation evaluation, but in practice it is difficult to obtain pixel-wise labels for large-scale datasets.

We compare our algorithm to MNcut [3] and the method of [14], which are publicly available¹⁰. As a baseline, the MNcut [3] independently segments each image with $K=2$ and the fg and bg are assigned so that the segmentation accuracy is maximized. For [14], we apply the algorithm several times by changing the number of topics from two to eight, and the best results are reported. Note that most previous cosegmentation methods including [6] and [7] cannot run well with a large number of images. ([7] reported that their algorithm took between 4 and 9 hours for 30 images).

For ImageNet tests, we select 50 synsets that provide bounding box labels. We randomly select up to 1000 images per synset. Since the ImageNet images are too diverse to be jointly cosegmented at once, we first split each synset into 100 disjoint sets $\mathcal{I} = \mathcal{I}_1 \cup \dots \cup \mathcal{I}_{100}$ by our diversity ranking and clustering. Then, our cosegmentation is separately applied into each \mathcal{I}_o . This decomposition is much more favorable to the performance. We tested a single simultaneous cosegmentation with 1,000 images with full dependency, but both accuracy and speed were much worse.

Fig.5.(a) shows an example of synset decomposition. A single synset has several different aspects, which were successfully detected by our diversity ranking and clustering. Table 3 shows the segmentation accuracies for 13 selected synsets. Our algorithm significantly outperformed the two competitors by more than 10%. Our algorithm took 60-70 minutes for 1,000 images on a single machine. Note that this computation time can be significantly reduced by parallelization as discussed in section 3.2.

Fig.4 and Fig.5.(b) show some examples of cosegmentation on the MSRC and ImageNet datasets. We made two interesting observations here: (i) Our method can easily segment multiple instances in the images. (ii) Our algorithm is robust against an incorrect selection of K . In the duck example of the second column of Fig.4, the best choice of K would be four, but a faulty guess with $K=8$ did little harm. The four significant segments are successfully detected (*e.g.*

Class (%)	Our method	Hochbaum et al.[6]	Julin et al.[7]
Aeroplane	37.6 \pm 10.6	25.6 \pm 9.9	26.5 \pm 7.9
Bike	68.4 \pm 12.6	66.8 \pm 13.9	58.4 \pm 11.6
Bird	57.0 \pm 18.2	30.4 \pm 19.3	50.3 \pm 19.2
Car	57.7 \pm 9.4	55.8 \pm 16.6	52.5 \pm 13.3
Cat	73.1 \pm 12.2	75.9 \pm 16.9	65.6 \pm 13.9
Chair	64.4 \pm 12.6	62.2 \pm 21.8	61.6 \pm 15.4
Cow	66.1 \pm 18.5	72.4 \pm 11.9	67.3 \pm 11.9
Dog	55.5 \pm 3.9	47.7 \pm 18.9	48.3 \pm 22.9
Face	78.5 \pm 11.4	72.1 \pm 18.4	60.9 \pm 12.0
Flowers	75.6 \pm 2.2	70.0 \pm 14.44	71.6 \pm 16.4
Sheep	69.2 \pm 16.6	43.7 \pm 19.3	70.5 \pm 16.1
Sign	68.7 \pm 12.9	58.8 \pm 17.9	64.1 \pm 17.5
Tree	67.6 \pm 1.1	60.2 \pm 13.0	60.8 \pm 13.1

Table 2. Accuracies of figure-ground segmentation tests for 100 random pairs of images per object from the MSRC dataset.

Class (in %)	Our method	MNcut [3]	LDA [14]
Barn spider	48.6 \pm 24.1	35.3 \pm 13.0	32.4 \pm 10.0
Hognose snake	55.3 \pm 22.0	47.2 \pm 17.0	44.7 \pm 17.1
Coral	79.3 \pm 20.1	66.4 \pm 22.0	52.6 \pm 14.7
St Bernard	68.2 \pm 21.3	50.5 \pm 13.7	45.7 \pm 12.3
Basenji	58.8 \pm 23.1	46.3 \pm 15.8	42.2 \pm 14.9
Tabby cat	67.2 \pm 22.1	51.3 \pm 16.6	49.6 \pm 14.6
Jaguar	67.8 \pm 21.0	50.2 \pm 14.7	49.4 \pm 14.5
Lion	63.6 \pm 22.4	50.7 \pm 17.7	47.6 \pm 16.8
Starfish	50.2 \pm 25.9	41.6 \pm 18.7	40.1 \pm 16.4
Polecat	58.3 \pm 21.5	47.6 \pm 15.7	44.7 \pm 13.4
Badger	51.6 \pm 24.6	43.0 \pm 17.9	41.3 \pm 16.3
Orangutan	61.3 \pm 26.0	49.5 \pm 19.8	48.0 \pm 18.3
Guenon monkey	58.8 \pm 24.8	47.8 \pm 16.9	46.4 \pm 16.2

Table 3. Accuracies of scalable cosegmentation tests for 13 selected synsets from the ImageNet dataset.

three ducks and grass) and the other four overestimated segments were trivially selected as tiny dots.

5. Conclusion

In this paper, we proved that the temperature of the system under linear anisotropic diffusion is submodular. Based on this finding, we design a constant-factor greedy solution to temperature maximization with limited sources. Our theoretic results were successfully applied to diversity ranking, single-image segmentation, and scalable cosegmentation.

Acknowledgement This work is supported by MURI N00014-07-1-0747, NSF DBI 0640543, AFOSR FA 9550-10-1024, and ONR N000140910758.

References

- [1] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. iCoseg: Interactive Co-segmentation with Intelligent Scribble Guidance. In *CVPR*, 2010. 1
- [2] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods. *IJCV*, 61(3):211–231, 2005. 2

⁹<http://www.image-net.org/challenges/LSVRC/2010>.

¹⁰Codes are available at [3]: <http://www.seas.upenn.edu/~timothee>, [14]: http://www.cs.washington.edu/homes/bcr/projects/mult_seg_discovery/



Figure 4. Four cosegmentation examples on the MSRC dataset. (a) Pairs of input images. (b) Our cosegmentation results with $K=8$. The cosegmented pairs are presented by the same colors. Some segments are too small to be shown. (c) Figure-ground segmentation results that are induced from the eight pairs of cosegments.

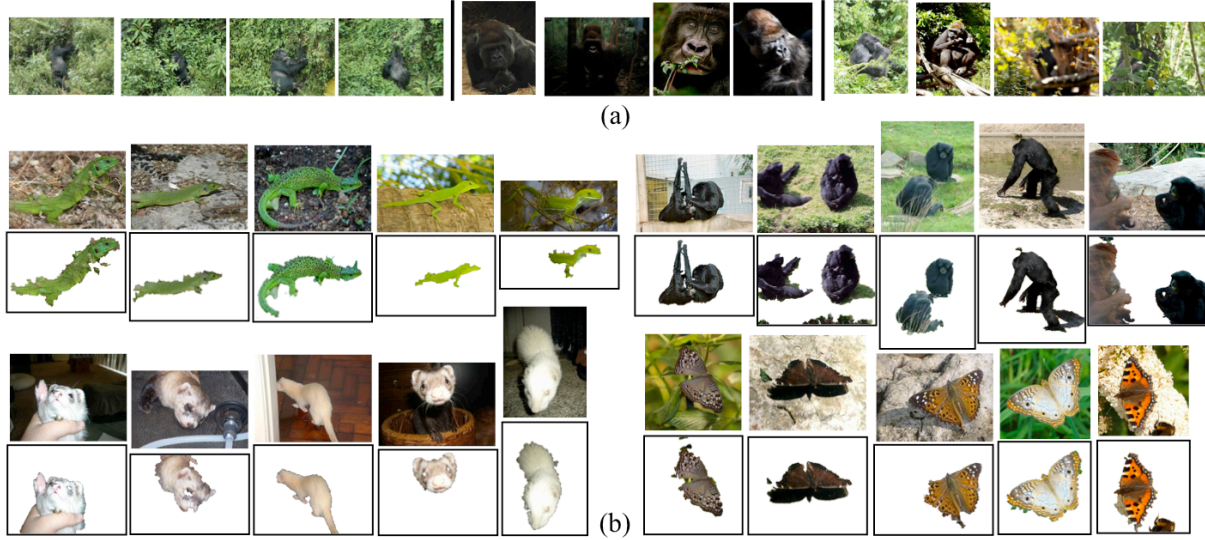


Figure 5. Examples of scalable cosegmentation on the ImageNet dataset. (a) Decomposition of the *Gorilla* Synset by the proposed diversity ranking and clustering. Three cluster centers and their three closest images are shown. (b) Examples of cosegmentation on *green lizard*, *siamang*, *ferret*, and *nymphalid butterfly*. In each set, 20~60 images are simultaneously cosegmented and five selected images are shown.

- [3] T. Cour, F. Benezit, and J. Shi. Spectral Segmentation with Multiscale Graph Decomposition. In *CVPR*, 2005. 7
- [4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 1, 2, 7
- [5] L. Grady. Random Walks for Image Segmentation. *IEEE PAMI*, 28:1768–1783, 2006. 3, 5
- [6] D. S. Hochbaum and V. Singh. An Efficient Algorithm for Cosegmentation. In *ICCV*, 2009. 1, 2, 6, 7
- [7] A. Joulin, F. Bach, and J. Ponce. Discriminative Clustering for Image co-segmentation. In *CVPR*, 2010. 1, 2, 6, 7
- [8] A. Krause and C. Guestrin. Beyond Convexity: Submodularity in Machine Learning. In *ICML Tutorials*, 2008. 2, 3
- [9] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective Outbreak Detection in Networks. In *ACM KDD*, 2007. 2, 5
- [10] A. Levinstein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. TurboPixels: Fast Superpixels Using Geometric Flows. *IEEE PAMI*, 31(12):2290–2297, 2009. 4
- [11] L. Mukherjee, V. Singh, and J. Peng. Scale Invariant Cosegmentation for Image Groups. In *CVPR*, 2011. 2
- [12] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An Analysis of Approximations for Maximizing Submodular Set Functions. *Math. Prog.*, 14:265–294, 1978. 3
- [13] 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. 1, 2
- [14] B. Russell, A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In *CVPR*, 2006. 7
- [15] S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation Revisited: Modes and Optimization. In *ECCV*, 2010. 1, 2
- [16] J. Weickert. *Anisotropic Diffusion in Image Processing*. ECI Series, Teubner-Verlag, 1998. 2, 3
- [17] J. Winn, A. Criminisi, and T. Minka. Object Categorization by Learned Universal Visual Dictionary. In *ICCV*, 2005. 1, 2, 6
- [18] J. Zhang, J. Zheng, and J. Cai. A Diffusion Approach to Seeded Image Segmentation. In *CVPR*, 2010. 2, 3
- [19] X. Zhu, A. B. Goldberg, J. V. Gael, and D. Andrzejewski. Improving Diversity in Ranking using Absorbing Random Walks. In *HLT-NAACL*, 2007. 3, 4
- [20] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *ICML*, 2003. 3