EURASIP Journal on Image and Video Processing

SpringerOpen

A hierarchical graph model for object cosegmentation

EURASIP Journal on Image and Video Processing 2013, **2013**:11 doi:10.1186/1687-5281-2013-11

Yanli Li (liyanli725@gmail.com)
Zhong Zhou (zz@vrlab.buaa.edu.cn)
Wei Wu (wuwei@vrlab.buaa.edu.cn)

ISSN 1687-5281

Article type Research

Submission date 9 June 2012

Acceptance date 2 February 2013

Publication date 26 February 2013

Article URL http://jivp.eurasipjournals.com/content/2013/1/11

A hierarchical graph model for object cosegmentation

Yanli Li[1]
Email: liyanli725@gmail.com

Zhong Zhou*[1]
*Corresponding author
Email: zz@vrlab.buaa.edu.cn

Wei Wu[1]
Email: wuwei@vrlab.buaa.edu.cn

[1]State Key Laboratory of Virtual Reality Technology & Systems, Beihang University, Beijing, China

Abstract

Given a set of images containing similar objects, cosegmentation is a task of jointly segmenting the objects from the set of images, which has received increasing interests recently. To solve this problem, we present a novel method based on a hierarchical graph. The vertices of the hierarchical graph involve pixels, superpixels and heat sources, and cosegmentation is performed as iterative object refinement in the three levels. With the inter-image connection in the heat source level and the intra-image connection in the superpixel level, we progressively update the object likelihoods by transferring message across images via belief propagation, diffusing heat energy within individual image via random walks, and refining the foreground objects in the pixel level via guided filtering. Besides, a histogram based saliency detection scheme is employed for initialization. We demonstrate experimental evaluations with state-of-the-art methods over several public datasets. The results verify that our method achieves better segmentation quality as well as higher efficiency.

Keywords

Cosegmentation, Hierarchical graph, Heat Source, Saliency Detection, Belief Propagation, Random Walks, Guided Filtering

1 Introduction

The term "cosegmentation" is first introduced by Rother et al. [1] in 2006, referring to the problem of simultaneously segmenting "similar" foreground objects in a set of images. The definition of "similar" commonly indicates the constraint that the distribution of some appearance cues such as color and texture in each image has to be similar. Cosegmentation has many

potential applications. It can be used for summarizing personal photo album, guiding multiple images' editing, boosting unsupervised object recognition, improving content based image retrieval and so on.

Since the introduction of the problem, various methods have been presented. One type of methods handles the problem of multi-class cosegmentation, while others focus on binary cosegmentation. In this article, we are interested in binary cosegmentation and observe that for most applications of binary cosegmentation several criteria should be followed: (1) automation, i.e., it is executed without user interactions; (2) scalability, i.e., it can be applied to hundreds of images instead of two images or small sized image sets; (3) focusing on "object" instead of "stuff". Here the "object" refers to "foreground things" such as a person or a bird, while "stuff" refers to "background regions" such as road or sky; (4) high segmentation accuracy; (5) low running time. According to these criteria, existing methods have some limitations. For example, the iCoseg system presented by Batra et al. [2] can obtain highly accurate results, but requires user input. The methods reviewed by Vicente et al. [3] all focus on cosegmenting two images. The recently presented CoSand [4] only extracts similar large regions, thus it often omits the small foreground objects in the images. Methods based on topic discovery like [5–7] all take superpixels as computation nodes, and hence they suffer from detail loss because superpixels tend to merge foreground regions with the backgrounds. Some unsupervised object segmentation methods [8–11] extract objects from multiple images via iteratively learning class models and segmenting objects in pixel level, while they are time-consuming because the employed optimization schemes like graphcut [12] and belief propagation [13] are inefficient with a large number of pixel nodes.

In this article, we try to meet these criteria by extracting the foreground objects with a three-level hierarchical graph model. As shown in Figure 1, the graph model is composed of the pixel, superpixel and heat source levels, in which superpixels are grouping units of pixels obtained by an over-segmentation method [14] and heat sources are the representative superpixels obtained by a bottom-up agglomerative clustering scheme. The term "heat source" is introduced in random walks [15], representing heat energy convergence points. Here, we adopt it to describe message transferring among images and heat energy diffusion within individual image. The iterative object refinement is operated at the three levels with different optimization schemes. The heat source level utilizes belief propagation [13] for message transferring. In the superpixel level, random walks [15] is employed for heat energy diffusion. In the pixel level, we refine the foreground objects within each image via guided filtering [16]. By doing so, the foreground objects are gradually extracted. Besides, we employ a histogram based saliency detection method [17] for initializing the object likelihoods.

Figure 1 An illustration of the hierarchical graph model for cosegmentation. The graph model is composed of the pixel, superpixel and heat source levels. The cosegmentation method is performed by message transferring among images in the heat source level, heat energy diffusion in the superpixel level and local refinement in the pixel level.

It is no doubt that our method is automatic and has the following advantages. (1) It is scalable. Since the superpixel and pixel levels both treat each image separately, and the heat source level's integration only operates on limited heat sources, this method has high parallelization

capacity and can be easily applied to large scale image collection. (2) It focuses on "object" instead of "stuff". This is because our method is initialized by saliency detection, which can filter out background stuff. (3) It is computationally more efficient. Compared with methods [8,9,18] which perform message transferring among images using a large number of superpixels or pixels, our method uses a small number of heat sources and thus significantly reduce computation time. (4) It can preserve object boundaries. This method finally refines object segmentation in the pixel level, and hence avoids the problem of detail loss existing in other superpixel based methods.

The remainder of this article is organized as follows. After summarizing the related study in Section 2, we present the hierarchical graph model in Section 3. The stages of object refinement along the model, including foreground initialization, local object refinement, message transferring and heat energy diffusion are described in Section 4. Experimental results are demonstrated in Section 5, and we conclude the article in the last section.

2 Related study

Basically, the solutions to cosegmentation can be roughly classified into two categories: clustering based methods [5–7, 19] and labeling based methods [3, 8–11, 18]. The former tries to partition nodes (pixels or superpixels) in the images into distinct, semantically coherent clusters, while the latter aims at assigning each node with a unique label.

2.1 Clustering based methods

Under the assumption that similar objects often recur in multiple images, clustering based methods employ clustering models to discover such frequent regions. The well-known clustering models include topic discovery models like probabilistic latent semantic analysis (PLSA) [20], and geometry based models like normalized cuts (NCut) [21]. Motivated by the success of topic discovery in text analysis, Russell et al. [5] first adopt PLSA to address the cosegmentation problem. Later, Cao et al. [6] and Zhao et al. [7] both present spatially coherent topic models to encode the spatial relationship of image patches which is ignored by the traditional topic models. Combining NCut and supervised classification technique, Joulin et al. [19] utilize a discriminative clustering scheme to tackle the cosegmentation problem. For speeding up, all clustering based methods take superpixels as computation nodes. The major limitation of these methods is the lower segmentation accuracy caused by the over-segmentation methods.

2.2 Labeling based methods

Considering the Markov property in the images, labeling based methods formulate cosegmentation as a Markov random field (MRF) energy minimization problem. Over the past decade, methods that use graphcut [12] to minimize MRF energy have become the standard for figure-ground separation.

One technique is to minimize an energy function that is a combination of a pairwise MRF energy and a histogram matching term. The histogram matching terms such as $L1$ norm model [1], $L2$ norm model [22] and "reward" model [23] force foreground histograms between a pair of images to be similar. Vicente et al. [3] review these models and make a comparison. Yet these methods are limited to two images. Another technique, also called unsupervised object segmentation such as LOCUS [8], ClassCut [9], Arora et al. [10] and Chen et al. [11], performs object cosegmentation by iteratively learning the object geometric models and segmenting the foreground objects. The initialization stages of these methods play an important role for energy minimization. For example, LOCUS [8] takes the pre-trained mask and edge probability maps as the initial object models, ClassCut [9] uses a general object detector [24] to locate objects. However, these methods are limited to segmenting objects with similar geometric shape. In contrast, the recently proposed cosegmentation method—BiCos [18] is more general and can be applicable for any non-rigid objects. BiCos [18] operates at the two levels: the bottom level treats each image separately and uses graphcut [12] to refine foreground objects in pixel level, whereas the top level takes superpixels as computation units and employs a discriminative classification to propagate information among images.

Our method falls into the last category. The main idea is to combine multiple schemes along a three-level hierarchical graph to refine foreground objects successively. In contrast to other labeling based methods [3, 8–11, 18], this method has the following characteristics: (1) utilization of heat sources for message propagation among images, which can significantly reduce computation time; (2) a saliency detection based initialization, which can remove the impact of background stuff; (3) instead of using graphcut [12] to refine objects in the pixel level, we introduce guided filtering [16] for local refinement. In experiments, we compare our method quantitatively and qualitatively with other state-of-the-art methods over several public datasets. As a outcome, our method achieves better segmentation quality as well as lower computation time.

3 The hierarchical graph model

3.1 Problem formulation

Given a set of images containing objects of the same class, $\mathcal{I} = \{I_k, k = 1, \ldots, K\}$, the goal of cosegmentation is to simultaneously extract the foreground objects. We formulate this problem as a binary labeling: $\mathcal{L} = \{L_k, k = 1, \ldots, K\}$, which assigns each pixel x in the image I_k with a label $L_k(x)$. $L_k(x) = 0$ indicates x belongs to the background, whereas $L_k(x) = 1$ to the foreground. The best labeling follows maximum a posteriori estimation, i.e., $\mathcal{L}^* = \arg\max_{\mathcal{L}} p(\mathcal{L}|\mathcal{I})$. Based on the Bayesian perspective, $p(\mathcal{L}|\mathcal{I}) \propto p(\mathcal{L})p(\mathcal{I}|\mathcal{L})$, where $p(\mathcal{L})$ is the labeling prior and $p(\mathcal{I}|\mathcal{L})$ is the observation likelihood. Under the assumption that the prior follows uniform distribution and the observation likelihood is pair-wise dependent among images, the posteriori can be rewritten as:

$$p(\mathcal{L}|\mathcal{I}) \propto \prod_k p(I_k|L_k) \prod_{(k_1,k_2)} p(I_{k_1}, I_{k_2}|L_{k_1}, L_{k_2}) \tag{1}$$

The corresponding energy function (i.e., $E(x) = -\log p(x)$) is:

$$E(\mathcal{L}|\mathcal{I}) = \sum_{k} E_d(I_k|L_k) + \sum_{(k_1,k_2)} E_s(I_{k_1}, I_{k_2}|L_{k_1}, L_{k_2}) \tag{2}$$

The energy function combines the unary terms $E_d(\cdot)$ and the pairwise terms $E_s(\cdot,\cdot)$. In our study, the unary term is composed of two parts:

$$E_d(I_k|L_k) = E_{d_1}(I_k|L_k) + E_{d_2}(I_k|L_k, \theta_k), \tag{3}$$

where $E_{d_1}(I_k|L_k)$ is derived from saliency detection, and $E_{d_2}(I_k|L_k, \theta_k)$ is inferred under the guide of an inherent object model. θ_k is the latent parameter set for the object model of I_k.

The pairwise term can be considered as a smooth term, which penalizes the inconsistent labeling among images. Ideally, this term should be formulated in the pixel level. For computational efficiency, we define it in the heat source level using appearance information (see Equation (8)). Minimizing the above energy with respect to all discrete labels is intractable. Instead, we relax the labels firstly, i.e., let $L_k(x) \in [0, 1]$ be the object likelihood, and iteratively update them along a hierarchical graph model, finally obtain the segmentation results by rounding.

3.2 The hierarchical graph and our method

As shown in Figure 1, the graph model is composed of three types of nodes: pixels, superpixels and heat sources. For each image, superpixels are the clustering units of coherent pixels, and heat sources are the representative superpixels located in the centers of the clustering regions formed by coherent superpixels. In our implementation, the superpixels are extracted by an over-segmentation method—Turbopixels [14]. The generation of heat sources will be described in detail in Section 4.2.

Based on the graph model, our method successively updates the object likelihoods by the following iteration: (1) estimating the latent parameters and refining object segmentation, (2) transferring message among images and diffusing heat energy within individual image. Specifically, we first obtain the object likelihoods in each image with saliency detection [17], and then estimate the latent parameters to update the object likelihoods. The likelihoods of the heat sources are further updated among images via message transferring which is fulfilled by belief propagation [13], and diffused to other superpixels using random walks [15] within individual image. Now the likelihoods can be considered as input for further iteration. In the following sections, we denote the updated object likelihoods at different stages by $L_k^{*,t}, t = 0, \ldots, 3, k = 1, \ldots, K$. To summarize the cosegmentation method presented in this article, we provide a high level overview of the method pipeline as follows.

- **Input:** a set of images containing objects of the same class $\mathcal{I} = \{I_k, k = 1, \ldots, K\}$
- **Output:** the cosegmentation results with the form of binary labeling $\mathcal{L}^* = \{L_k^*, k = 1, \ldots, K\}$

Step 1. Initialization (Section 4.1)

a) partition each image I_k into a set of superpixels S_k and extract heat sources Z_k.

b) obtain the initial object likelihoods $L_k^{*,0}$ via saliency detection [17].

c) estimate the latent parameter set θ_k.

d) acquire the updated object likelihoods $L_k^{*,1}$ via guided filtering [16].

Step 2. Global message transferring (Section 4.2)

Optimize the energy function defined in Equation (6) via belief propagation [13] to provide the updated object likelihoods $L^{*,2}(Z)$ for all heat sources.

Step 3. Local heat energy diffusion (Section 4.3)

For each image I_k, the object likelihoods of the heat sources $L_k^{*,2}(Z_k)$ are diffused to other superpixels $U_k = S_k - Z_k$ via random walks [15], obtaining $L_k^{*,2}(U_k)$.

Step 4. Local object refinement (Section 4.1)

a) let $L_k^{*,3} = (L_k^{*,0} + L_k^{*,1} + L_k^{*,2})/3$.

b) re-estimate the latent parameter set θ_k.

c) acquire the updated object likelihoods $L_k^{*,1}$ via guided filtering [16].

Step 5. Repeat Step 2, 3, and 4 until convergence. The final labeling L_k^* is obtained by binarizing $L_k^{*,3}$.

4 Hierarchical graph based object cosegmentation

4.1 Initialization and local refinement

One major visual characteristic of objects is that they often stand out as saliency [24]. Based on this characteristic, we apply saliency detection to initially detect foreground regions in each image. Over various of saliency detection methods, we choose a recently proposed histogram based method [17] for its efficiency and effectiveness. Figure 2b demonstrates the saliency detection result of Figure 2a. We define the initial object likelihoods $L_k^{*,0}$ as the saliency likelihoods.

Figure 2 Saliency detection based model initialization. (a) The input image, **(b)** the saliency detection result, **(c)** the segmentation result built on GMM, and **(d)** the segmentation result obtained after guided filtering.

The segmentation results obtained by thresholding saliency likelihoods often contain holes and ambiguous boundaries. Motivated by the interactive segmentation methods, e.g., GrabCut [25], we utilize the inherent color Gaussian mixture model (GMM) in the image to update the object likelihoods. Two GMMs, one for the foreground and another for the background, are estimated in RGB color space. Each GMM is taken to be a full-covariance Gaussian mixture with M components. The GMM parameters are defined as: $\theta_k = \{\theta_k^J | J \in \{B, F\}\}$, in which $\theta_k^J = \{\theta_{m,k}^J | m = 1, \ldots, M\}$, $\theta_{m,k}^J = (\mu_{m,k}^J, \Sigma_{m,k}^J, \omega_{m,k}^J)$. $(\mu_{m,k}^F, \Sigma_{m,k}^F, \omega_{m,k}^F)$ are the mean, covariance and weighting values for the foreground components, and $(\mu_{m,k}^B, \Sigma_{m,k}^B, \omega_{m,k}^B)$ for the background

components. The GMM parameters are estimated from the initial likelihoods as follows: (1) given two thresholds T_1 and T_2, satisfying $0 < T_1 < T_2 < 1$, we label the pixels with $L_k^{*,0}(x) > T_1$ as foreground, whereas $L_k^{*,0}(x) < T_2$ as background; (2) the colors of the foreground and background regions are clustered into M components using K-Means [26], respectively; (3) for each component, we statistically acquire its parameters $\theta_{m,k}^J$. The object likelihoods built on the GMMs are given by:

$$p(I_k(x)|\theta_k^J) = \max_m(p(I_k(x)|\theta_{m,k}^J)) \tag{4}$$

$$p(I_k(x)|\theta_{m,k}^J) = \omega_{m,k}^J \exp(-\|I_k(x) - \mu_{m,k}^J\|/\Sigma_{m,k}^J)/\sqrt{|\Sigma_{m,k}^J|} \tag{5}$$

Segmenting objects by directly thresholding the updated object likelihoods will result in noises, as shown in Figure 2c. We use guided filtering [16] to remove noises. The main idea of guided filtering [16] is that, given the filter input p, the filter output q is locally linear to the guidance map I, $q_i = a_x I_i + b_x$, $\forall i \in w_x$, where w_x is a window with radius r centered at the pixel x. By minimizing the difference between the filter input p and the filter output q, i.e., $Err(a_x, b_x) = \Sigma_{i \in w_x}((p_i - q_i)^2 + \epsilon a_x^2)$, we can obtain a_x, b_x and the filter output q.

Based on guided filtering [16], we perform local refinement with three steps: (1) obtaining the foreground likelihood map $L_{k,F} = \{p(I_k(x)|\theta_k^F)\}$ and the background likelihood map $L_{k,B} = \{p(I_k(x)|\theta_k^B)\}$; (2) taking the grayscale image of I_k as the guidance map, the two likelihood maps are filtered, respectively (denoting the filter outputs as $\hat{L}_{k,F}$ and $\hat{L}_{k,B}$); (3) defining the refined object likelihoods as $L_k^{*,1} = \hat{L}_{k,F}/(\hat{L}_{k,F} + \hat{L}_{k,B})$. Figure 2d shows the refinement result of Figure 2c. As can be seen, the guided filtering based scheme can significantly improve segmentation quality.

4.2 Global message transferring

Due to the diversity of realistic scenes, saliency based object segmentation sometimes fails to extract objects of the same class (see Figure 3c). The segmentation quality can be further boosted by sharing appearance similarity among images. Unlike other cosegmentation methods [8,9,18] which propagate the distributions of visual appearance in the pixel or superpixel level, we perform message propagation in the heat source level to reduce computation time.

Figure 3 The segmentation results obtained before and after message transferring. (a) The input images, **(b)** the saliency detection results, **(c)** the segmentation results obtained in the initial stage, and **(d)** the segmentation results obtained after message transferring.

As stated in Section 3, heat sources are the representative superpixels located in the centers of the clustering regions formed by coherent superpixels. The regions are formed by a bottom-up agglomerative clustering scheme. Specifically, given an image I, we first partition it into a collection of superpixels via Turbopixels [14] (see Figure 4b, in which superpixels are encircled with red boundaries). Then we build an intra-image graph $G_S =< S, Y_S >$, where $S = \{s_i\}$ is the superpixel set and $Y_S = \{(s_i, s_j)\}$ is the edge set connecting all pairs of adjacent superpixels. The edge weight is defined by Gaussian similarity between the normalized mean RGB color of

the nodes, i.e., $w(s_i, s_j) = \exp(-\|I(s_i) - I(s_j)\|^2)/\sigma_s$, where σ_s is a variance constant. Based on the graph G_S, we use a greedy scheme to merge nodes one by one. Each time, we select the edge with the maximum weight value and merge its two nodes. This step is repeated until all nodes are merged into N regions. The central superpixel of each region is chosen as a heat source. Figure 4c demonstrates the clustering regions overlaid by the heat sources, in which the regions are encircled with green boundaries and the heat sources are colored in blue.

Figure 4 An example of extracting superpixels and heat sources from an input image. **(a)** The input image, **(b)** the superpixels extracted by Turbopixels [14] are encircled with red boundaries, and **(c)** the regions extracted by an agglomerative clustering scheme are encircled with green boundaries, and the extracted heat sources are colored in blue.

For message transferring among images, we construct an inter-image graph $G_Z =< Z, Y_Z >$. G_Z is an undirected complete graph, where $Z = \{z_i | z_i \in Z_k, k = 1, \ldots, K\}$ includes all heat sources from the input images, $Y_Z = \{(z_i, z_j)\}$ connects all pairs of heat sources. We update the object likelihoods of the heat sources by minimizing a standard MRF energy function that is the sum of unary terms $E_1(\cdot)$ and pairwise terms $E_2(\cdot, \cdot)$:

$$E(L(Z)) = \sum_{z_i \in Z} E_1(z_i) + \lambda \sum_{(z_i, z_j) \in Y_Z} E_2(z_i, z_j), \tag{6}$$

where λ is the weighting value balancing the trade off between the unary terms and the pairwise terms.

The unary term $E_1(\cdot)$ imposes individual penalties for assigning any likelihood $L(z_i)$ to the heat source z_i. We rely on the object likelihoods $L^{*,1}$ acquired in the previous stage to define this term:

$$E_1(z_i) = \left| L(z_i) - \left(\sum_{x \in z_i} L^{*,1}(x)/|z_i| \right) \right| \tag{7}$$

The pairwise term $E_2(\cdot, \cdot)$ defines to what extent adjacent heat sources should agree. It often depends on local observation. In our study, the pairwise potential takes the form:

$$E_2(z_i, z_j) = w(z_i, z_j)|L(z_i) - L(z_j)| \tag{8}$$

where $w(z_i, z_j)$ is the edge weight, defined as $w(z_i, z_j) = \exp(-\|f(z_i) - f(z_j)\|^2)/\sigma_z$, σ_z is a variance constant. $f(z)$ is a nine-dimensional descriptor for the heat source z, including three-dimensional mean Lab color feature, four-dimensional mean texture feature[a] and two-dimensional mean position feature. This definition suggests that the larger the weight for the edge, the more similar the labels for its two nodes.

We utilize belief propagation [13] to optimize the energy function in several bounds. The main idea of belief propagation is to iteratively update a set of message maps between neighboring nodes. The message maps that are denoted by $\{m^t_{z_i \to z_j}(L(z_j)), t = 1, \ldots, T\}$ represent the transferred message from one node to another at each iteration. In our study, the message maps are

initially set to zero and updated as follows:

$$m^t_{z_i \to z_j}(L(z_j)) = \min_{L(z_i)} \left(E_1(z_i) + \lambda E_2(z_i, z_j) + \sum_{z_k \in Z/z_j} m^{t-1}_{z_k \to z_i}(L(z_i)) \right) \tag{9}$$

Finally, a belief vector is computed for each node, $b_{z_i}(L(z_i)) = E_1(z_i) + \sum_{z_j \in Z} m^T_{z_j \to z_i}(L(z_i))$, and the updated object likelihoods are expressed as: $L^{*,2}(z_i) = b_{z_i}(0)/(b_{z_i}(0) + b_{z_i}(1))$.

4.3 Local heat energy diffusion

After global message transferring, the object likelihoods for heat sources preserve appearance similarity among images. We further diffuse them to other superpixels. As illustrated in the middle level of Figure 1, this is performed by heat energy diffusion within individual image. The heat energy diffusion can be imagined in the following situation: putting some heat sources in a metal plate, the heat energy will diffuse to other points as time goes by, finally each point will have a stable temperature. How to calculate such steady-state temperatures? This is a well-known Dirichlet energy minimization problem:

$$u^* = \arg\min_u(E(u)) = \arg\min_u \frac{1}{2} \int_{u \in \Omega} |\nabla u|^2 d\Omega \tag{10}$$

Grady [15] states the similar problem in discrete space with the term "random walks". Based on a graph $G_X =< X, Y_X >$, where $X = \{x_i\}$ is the node set and $Y_X = \{(x_i, x_j)\}$ is the set of node pairs, the Dirichlet energy function takes the form:

$$E(u(X)) = \frac{1}{2} \sum_{(x_i, x_j) \in Y_X} w(x_i, x_j)(u(x_i) - u(x_j))^2, \tag{11}$$

where $w(x_i, x_j)$ is the edge weight for the adjacent node pair (x_i, x_j).

In our study, the random walks works on the graph $G^S_k =< S_k, Y^S_k >$ for the image I_k, where $S_k = \{s_i\}$ is the superpixel set and $Y^S_k = \{(s_i, s_j)\}$ is the edge set connecting all pairs of adjacent superpixels. The corresponding energy function is:

$$E(L(S_k)) = \frac{1}{2} \sum_{(s_i, s_j) \in Y^S_k} w(s_i, s_j)(L(s_i) - L(s_j))^2 = \frac{1}{2} L(S_k)^T Q L(S_k) \tag{12}$$

where $Q = D - A$ is the Laplacian matrix, in which $A = \{w(s_i, s_j)\}$ is the edge weight matrix, and D is a diagonal matrix with the entities $D(s_i, s_i) = \sum_j w(s_i, s_j)$.

We divide the node set S_k into two parts: the heat sources Z_k and the superpixels $U_k = S_k - Z_k$. The energy function can be rewritten as:

$$E(L(S_k)) = [L(Z_k)^T, L(U_k)^T] \begin{bmatrix} Q_{Z_k} & B \\ B^T & Q_{U_k} \end{bmatrix} \begin{bmatrix} L(Z_k) \\ L(U_k) \end{bmatrix}, \tag{13}$$

where Q_{Z_k} and Q_{U_k} correspond to the Laplacian matrix for the node set Z_k and U_k, respectively.

Minimizing $E(L(S_k))$ is equal to differentiating $E(L(S_k))$ with respect to $L(U_k)$ and yields: $L(U_k) = -B^T L(Z_k)/Q_{U_k}$. $L(Z_k)$ are the object likelihoods acquired in the previous stage, i.e., $L(Z_k) = L^{*,2}(Z_k)$. The diffused object likelihoods for U_k are obtained by: $L^{*,2}(U_k) = -B^T L^{*,2}(Z_k)/Q_{U_k}$. The nonsingularity of Q_{U_k} guarantees that the solution exists and is unique.

For each pixel x, its object likelihood $L^{*,2}(x)$ is assigned as the object likelihood of the superpixel it belongs to. Taking $L_k^{*,3}(x) = (L_k^{*,0}(x) + L_k^{*,1}(x) + L_k^{*,2}(x))/3$ as input, we further invoke local refinement (see Section 4.1) to optimize object segmentation. Figure 3 demonstrates the segmentation results obtained before and after heat energy diffusion. As can be seen, although the saliency based initialization stage sometimes fails to extract the foreground objects, the stages of message transferring and heat energy diffusion can boost segmentation quality via sharing visual similarity of objects among images.

5 Experimental results

We apply our hierarchical graph based cosegmentation method to five public datasets with varying scenario and difficulty, including Weizmann horses[b], Caltech-4[c], Oxford flowers[d], UCSD birds[e], and CMU iCoseg[f]. All images of these datasets have ground truth masks, which allows us to evaluate segmentation performance quantitatively.

5.1 Datasets and implementation details

5.1.1 Weizmann horses

The Weizmann horses dataset has 324 images, in which each image depicts a different instance of the horse class. All horses pose in their side view and face to the same direction. Generally speaking, the horses preserve fixed geometric models and occupy most parts of the images.

5.1.2 Caltech-4

The Caltech-4 dataset includes four categories: airplane, car, face, and motorbike. We omit the grayscale car and use the other three categories for evaluation. This is a large-scale dataset, in which both the airplane and motorbike categories contains 800 images, and the face category contains 435 images. Similar to the Weizmann horses dataset, each image of Caltech-4 only depicts one object and the object occupy most parts of the image.

5.1.3 *Oxford flowers*

The Oxford flowers dataset has 17 different flower species with 80 images per category. Each image contains a finite number of repeating subjects. Some flowers like sunflower occupy most parts of the images, while others like lily of the valley scatter in the images.

5.1.4 *UCSD birds*

The UCSD birds dataset consists of 200 bird categories and 6033 images in total. This is a challenging dataset, where the birds appear in their natural habitat, change considerably in terms of viewpoint and illumination, and even in some cases only a part of the bird is visible.

5.1.5 *CMU iCoseg*

The CMU iCoseg dataset was introduced in [2]. It contains 643 images divided into 38 groups which are collected in various real situations such as soccer players in a field, airshows in the sky, a brown bear around a river. Omitting the background stuffs, each group contains one or several foreground objects of the same class.

With these datasets, we are interested in two evaluations: (1) unsupervised object segmentation over the Weizmann horses and Caltech-4 datasets where each image captures only one object and the objects typically preserve fixed orientation and well-defined geometric shape; (2) object cosegmentation on the Oxford flowers, UCSD birds and CMU iCoseg datasets where each image contains one or several objects that appear in their natural habitat. The first evaluation is performed to quantitatively compare our method with several traditional unsupervised object segmentation methods [8–10] which are only applicable in this setting. The second evaluation tests how well our method works with real world data.

5.1.6 *Implementation details*

In the initialization stage, we partition each image into 1000 or less superpixels, and extract about $N = 50$ heat sources from these superpixels. The other parameters are set as: the GMM component number $M = 5$, the thresholds $T_1 = 0.38$, $T_2 = 0.52$, the guided filtering's parameters $r = 7$, $\epsilon = 0.04$, the variances $\sigma_s = 0.004$, $\sigma_z = 0.08$, and the weighting value $\lambda = 0.5$. All experiments are performed on a computer with 2.9 GHz CPU and 2 GB RAM.

5.2 Evaluation on Weizmann horses and Caltech-4

Here we compare our method over the Weizmann horses and Caltech-4 datasets with four related methods, including LOCUS [8], ClassCut [9], Arora et al. [10] and BiCos [18]. LOCUS [8], ClassCut [9], and Arora et al. [10] all take advantage of the objects' inherent geometric models to jointly extract the foreground objects. In contrast, our method and BiCos [18] make no assumption about the foreground objects' geometric shape. Given a ground truth mask, the segmentation accuracy is measured by the ratio of correctly labeled pixels with respect to

the total number of pixels. According to the performance reported in their articles, Table 1 summarizes the segmentation accuracies over the four classes.

Table 1 The average segmentation accuracies obtained with LOCUS [8], ClassCut [9], Arora et al. [10], BiCos [18] and our method over the Weizmann horses and Caltech-4 datasets

Method	Weizmann horses	Caltech airplane	Caltech face	Caltech motorbike
LOCUS [8]	**0.931**	-	-	-
ClassCut [9]	0.862	0.888	0.890	**0.903**
Arora et al. [10]	-	0.931	**0.924**	0.831
BiCos [18]	0.900	0.932	0.911	0.822
Our method	0.884	**0.943**	0.921	0.878

The values in bold indicate the best results.

As can be seen, LOCUS [8], ClassCut [9] and Arora et al. [10] achieve better performance on the horse, motorbike and face categories, respectively. The reason is that the geometric models employed in those methods can strongly separate the foreground and background regions. Yet BiCos [18] and our method can still achieve competitive performance even without geometric models. Our method outperforms BiCos [18] on the airplane, face and motorbike categories, while BiCos [18] performs better on the horse category.

5.3 Evaluations on Oxford flowers, UCSD birds and CMU iCoseg

As baselines, three state-of-the-art methods (Joulin et al. [19], CoSand [4], and ClassCut [9]) are evaluated using their implementations with the default parameter settings. Joulin et al. [19] is a clustering based method, which takes superpixels as basic units and utilizes discriminative clustering to find common objects. CoSand [4] takes the large coherent, appearance similar regions among images as the foreground objects. ClassCut [9] is an energy iteration based method, which first obtains object bounding boxes by [24], and then builds a common class model with color, shape and position cues, finally extracts foreground objects via iteratively optimizing an MRF energy function and updating the class model.

The segmentation accuracy is defined as the proportion of pixels correctly classified as foreground or background by comparing the segmentation results with the ground truth. We take the form: $F\_\text{Measure} = 2 * \text{pre} * \text{rec}/(\text{pre} + \text{rec})$, where pre is defined as the ratio of true positive pixels (i.e., the pixels labeled as foreground actually belong to foreground) to all labeled foreground pixels, and rec is defined as the ratio of true positive pixels to ground truth pixels. The average segmentation accuracies across all images are shown in Table 2. Several examples from the Oxford flowers, UCSD birds and CMU iCoseg datasets can be seen in Figure 5.

Table 2 The segmentation performance of CoSand [4], ClassCut [9], Joulin et al. [19] and our method over the Oxford flowers, UCSD birds and CMU iCoseg datasets

Method	Oxford flowers		UCSD birds		CMU iCoseg	
	Accuracy	Time(s)	Accuracy	Time(s)	Accuracy	Time(s)
CoSand [4]	0.68	39.21	0.42	37.50	0.52	23.90
ClassCut [9]	0.72	95.96	0.32	93.71	0.51	78.43
Joulin et al. [19]	0.70	33.07	0.35	19.44	0.43	19.19
Our method (initial)	0.67	-	0.52	-	0.64	-
Our method (final)	**0.84**	**24.14**	**0.68**	**13.11**	**0.74**	**11.19**

The values in bold indicate the best results.

Figure 5 Segmentation comparison with ClassCut [9], Joulin et al. [19] and CoSand [4] on the Oxford flowers, UCSD birds and CMU iCoseg datasets. The regions in white indicate the foreground objects, while the regions in black stand for the background. **(a)** The input images, **(b)** ClassCut [9]'s results, **(c)** Joulin et al. [19]'s results, **(d)** CoSand [4]'s results, and **(e)** our method's results.

5.3.1 Overall performance

As illustrated in Table 2 and Figure 5, our method outperforms the three methods in terms of segmentation accuracy as well as computation time. The method of Joulin et al. [19] takes superpixels as basic units, thus the objects' boundaries are not clearly delineated as some superpixels merge foreground and background regions together. CoSand [4] only focuses on extracting the large coherent regions, it performs poorly for the figure-ground separation task. For example, it only extracts the black regions in the panda image set, failing to detect the white regions as foreground objects. ClassCut [9] can extract most of foreground regions, while it tends to omit some fragile regions like the petals in the Oxford flowers dataset. This is because the over-segmentation method it adopted has merged the boundaries with backgrounds. In contrast, our method can extract the whole foreground object accurately, no matter it is composed of one or several appearance distributions. We attribute this to the initialization scheme and the appearance sharing among images.

The benefit of segmenting all images together has been qualitatively shown in Figure 3. In Table 2, we quantitatively compare the segmentation accuracies obtained before and after sharing appearance similarity among images, obtaining that the accuracies are improved from 0.67, 0.52, 0.64 to 0.84, 0.68, 0.74 for the Oxford flowers, UCSD birds and CMU iCoseg datasets, respectively. Figure 6 compares some segmentation results obtained in the initialization and last stages. We can observe that most errors induced in the initialization stage are rectified finally.

Figure 6 Segmentation results obtained before and after sharing appearance similarity. The white regions denote the foreground objects, while the black regions stand for the background. **(a)** The input images, **(b)** the segmentation results obtained in the initial stage, and **(c)** the segmentation results obtained in the final stage.

5.3.2 Initialization performance

One contribution of our method is applying saliency detection with guided filtering to initially obtain foreground regions. To verify this stage's effectiveness, we compare it with other initialization schemes, including GrabCut [25] used in BiCos [18], the large coherence regions presented in CoSand [4] and the initialization stage of ClassCut [9]. Since the initialization stages are all performed in still images, we randomly select 100 images from the three datasets for comparison.

In BiCos [18], GrabCut [25] estimates the foreground regions by optimizing a MRF energy function with the foreground and background color models. The foreground model is estimated with a bounding box in the center (50 % of the image size) and the background model is estimated from the rest. In CoSand [4], the foreground region comes from K-way segmentation. As suggested in the article, the number of segments K ranges from two to eight and the highest accuracies are reported. In ClassCut [9], a class model with shape, location and color cues is initialized by an object detector [24], and the foreground regions are estimated by optimizing a MRF energy function with the class model.

Table 3 shows the average segmentation accuracies as well as computation time for different initialization schemes. As can be seen, our initialization scheme achieves best performance for the UCSD birds and CMU iCoseg datasets, while GrabCut [25] reports higher accuracy than ours for the Oxford flowers dataset. We believe that this is due to the characteristics of the dataset, where the objects tend to be centered in the image and have a good contrast with the backgrounds. Under such constraint situation, the class models can be accurately estimated by GrabCut [25]. In contrast, the UCSD birds and CMU iCoseg datasets are more general, which verifies that our method is more flexible to be applied to real situations. Besides, our initialization scheme is significantly faster than those competitors.

Table 3 The segmentation performance obtained by the initial stages of BiCos [18], CoSand [4], ClassCut [9] and our method over the Oxford flowers, UCSD birds and CMU iCoseg datasets

	Oxford flowers		UCSD birds		CMU iCoseg	
Method	**Accuracy**	**Time(s)**	**Accuracy**	**Time(s)**	**Accuracy**	**Time(s)**
BiCos [18]	**0.72**	14.06	0.48	8.40	0.61	7.27
CoSand [4]	0.63	20.00	0.32	12.00	0.43	10.00
ClassCut [9]	0.57	23.32	0.42	18.00	0.31	11.40
Our method	0.67	**2.70**	**0.52**	**1.55**	**0.64**	**1.32**

The values in bold indicate the best results.

5.3.3 Running time

One advantage of our method is its efficiency. Table 2 compares the running time of our methods with others. To further learn about how the time is cost in the whole process, we analyze each step's performance on the Oxford flowers, UCSD birds and CMU iCoseg datasets. As shown in Table 4, most of the time is spent on extracting superpixels, while the main stages in the article, including saliency detection, local refinement, global message transferring and heat energy diffusion cost only 8.01 s in total for the Oxford flowers dataset, 4.92 s for the UCSD birds dataset and 4.32 s for the CMU iCoseg dataset.

Table 4 The running time cost by each stage of our method over the Oxford flowers, UCSD birds and CMU iCoseg datasets

Dataset	Superpixel extraction	Heat source extraction	Saliency detection	Local refinement	Heat energy transfer and diffusion	Total time (s)
Oxford flowers	16.13	0.28	0.24	2.28	5.21	24.14
UCSD birds	8.19	0.14	0.18	1.47	3.13	13.11
CMU iCoseg	6.87	0.12	0.19	1.30	2.71	11.19

5.4 Failure cases

Our method works under an assumption that the interested objects should stand out as saliency. Yet such an assumption may not hold in some cases, which can be demonstrated over the MOMI dataset[g]. Figure 7 illustrates some failure cases of our method for the images from the MOMI dataset. As illustrated, although the "kfc", "lego", and "pringles" regions recur in the image sets, they are not too distinct with other regions to be detected as saliency. Our method fails to separate them from the backgrounds under such cases.

Figure 7 Failure cases. (a) The input images, **(b)** the segmentation results, and **(c)** the ground truth.

6 Conclusion

In this article, we present an iterative energy minimization method along a hierarchical graph for object cosegmentation. Starting from initialization by saliency detection, the method alternates via updating the latent parameters, refining object segmentation and propagating appearance distribution among images. Experiments demonstrate its superiority over start-of-the-art methods in aspects of accuracy and computation time. We attribute this to the combination of saliency detection, guided filtering and heat sources.

Still there are several issues remained to be explored. Currently, our method works under the assumption that the input images contain the common foreground objects. It is worth exploring a more general case that the input image set is composed of several groups where each group contains the common foreground objects. In addition, considering the parallelization capacity of our method, the system can be redesigned for implementation in parallel graphic hardware.

Endnotes

[a]http://www.robots.ox.ac.uk/~vgg/research/texclass/.
[b]http://www.msri.org/people/members/eranb/.
[c]http://www.vision.caltech.edu/archive.html.
[d]http://www.robots.ox.ac.uk/~vgg/data/flowers/.
[e]http://www.vision.caltech.edu/visipedia/CUB-200.html.

[f]http://chenlab.ece.cornell.edu/projects/touch-coseg/.
[g]http://imp.iis.sinica.edu.tw/ivclab/research/coseg/.

Competing interests

The authors declare that they have no competing interests.

Acknowledgments

This study was supported by the National 973 Program of China under Grant No. 2009CB320805, the Natural Science Foundation of China under Grant No. 61170188, and Fundamental Research Funds for the Central Universities of China.

References

1. C Rother, V Kolmogorov, T Minka, A Blake, in *IEEE Conference on Computer Vision and Pattern Recognition*, Cosegmentation of image pairs by histogram matching, vol.1, (Washington, 2006), pp. 993–1000

2. D Batra, A Kowdle, D Parikh, in *IEEE Conference on Computer Vision and Pattern Recognition*, iCoseg: interactive co-segmentation with intelligent scribble guidance, vol. 1, (San Francisco, 2010), pp. 3169–3176

3. S Vicente, V Kolmogorov, C Rother, in *European Conference on Computer Vision*, Cosegmentation revisited: models and optimization, vol. 2, (Heraklion, 2010), pp. 465–479

4. G Kim, EP Xing, L Fei-Fei, T Kanade, in *IEEE International Conference on Computer Vision*, Distributed cosegmentation via submodular optimization on anisotropic diffusion, vol. 1, (Barcelona, 2011), pp. 169–176

5. B Russell, A Efros, J Sivic, W Freeman, A Zisserman, in *IEEE Conference on Computer Vision and Pattern Recognition*, Using multiple segmentations to discover objects and their extent in image collections, vol. 2, (New York, 2006), pp. 1605–1614

6. L Cao, L Fei-Fei, in *IEEE International Conference on Computer Vision*, Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes, vol. 1, (Rio de Janeiro, 2007), pp. 1–8

7. B Zhao, L Fei-Fei, EP Xing, in *European Conference on Computer Vision*, Image segmentation with topic random field, vol. 5, (Heraklion, 2010), pp. 785–798

8. J Winn, N Jojic, in *IEEE International Conference on Computer Vision*, LOCUS—learning object classes with unsupervised segmentation, vol. 1, (Beijing, 2005), pp. 756–763

9. B Alexe, T Deselaers, V Ferrari, in *European Conference on Computer Vision*, ClassCut for unsupervised class segmentation, vol. 5, (Heraklion, 2010), pp. 380–393

10. H Arora, N Loeff, DA Forsyth, N Ahuja, in *IEEE Conference on Computer Vision and Pattern Recognition*, Unsupervised segmentation of objects using efficient learning, vol. 1, (Minneapolis, 2007), pp. 1–7

11. Y Chen, L Zhu, A Yuille, H Zhang, in *IEEE Conference on Computer Vision and Pattern Recognition*, Unsupervised learning of probabilistic object models (POMs) for object classification, segmentation and recognition, vol. 1, (Anchorage, 2008), pp. 1–8

12. V Kolmogorov, R Zabih, What energy functions can be minimized via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **2**(26), 147–159 (2004)

13. P Felzenszwalb, Efficient belief propagation for early vision. Int J. Comput. Vis. **70**, 41–54 (2006)

14. A Levinshtein, A Stere, KN Kutulakos, DJ Fleet, SJ Dickinson, K Siddiqi, TurboPixels: fast superpixels using geometric flows. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 2290–2297 (2009)

15. L Grady, Random walks for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **28**, 1768–1783 (2006)

16. K He, J Sun, X Tang, in *European Conference on Computer Vision*, Guided image filtering, vol. 1, (Heraklion, 2010), pp. 1–14

17. M Cheng, G Zhang, NJ Mitra, X Huang, S Hu, in *IEEE Conference on Computer Vision and Pattern Recognition*, Global contrast based salient region detection, vol. 1, (Colorado Springs, 2011), pp.409-416

18. Y Chai, V Lempitsky, A Zisserman, in *IEEE International Conference on Computer Vision*, BiCoS: a bi-level co-segmentation method for image classification, vol. 1, (Barcelona, 2011), pp. 2579–2586

19. A Joulin, F Bach, J Ponce, in *IEEE Conference on Computer Vision and Pattern Recognition*, Discriminative clustering for image co-segmentation, vol. 1, (San Francisco, 2010), pp. 1943–1950

20. T Hofmann, Unsupervised learning by probabilistic latent semantic analysis. Mach. Learn. **43**, 177–196 (2001)

21. J Shi, J Malik, in *IEEE Conference on Computer Vision and Pattern Recognition*, Normalized cuts and image segmentation, vol. 1, (San Juan, 1997), pp. 731–737

22. L Mukherjee, V Singh, C Dyer, in *IEEE Conference on Computer Vision and Pattern Recognition*, Half-integrality based algorithms for cosegmentation of images, vol. 1, (Miami, 2009), pp. 2028–2035

23. D Hochbaum, V Singh, in *IEEE Conference on Computer Vision*, An efficient algorithm for co-segmentation, vol. 1, (Kyoto, 2009), pp. 269–276

24. B Alexe, T Deselaers, V Ferrari, in *IEEE Conference on Computer Vision and Pattern Recognition*, What is an object, vol. 1, (San Francisco, 2010), pp. 73–80

25. C Rother, V Kolmogorov, A Blake, Grabcut—interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. **23**(3), 309-314 (2004)

26. R Duda, P Hart, D Stork, *Pattern classification*, 2nd edn. (Wiley Press, New York, 2000)

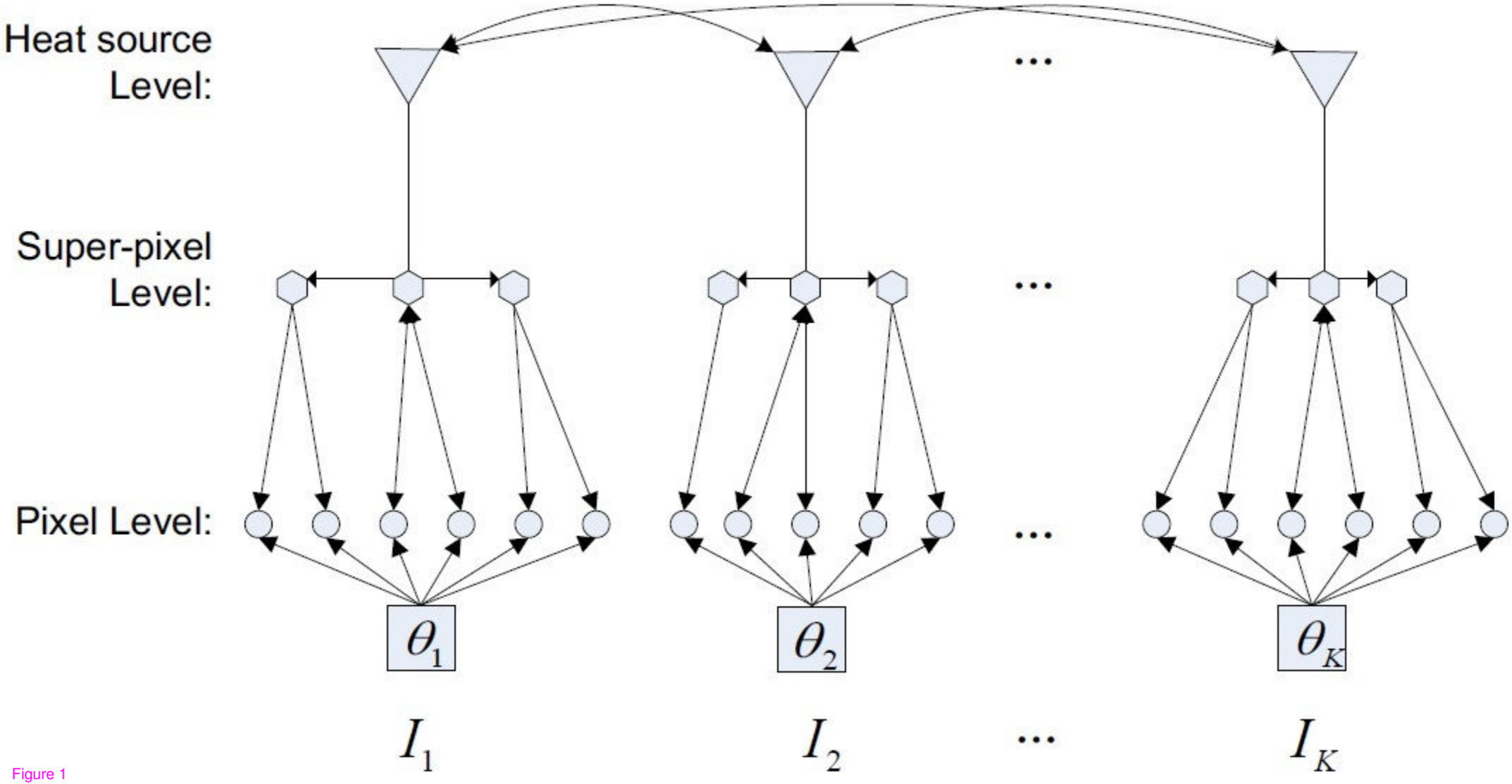

Heat source Level:
Super-pixel Level:
Pixel Level:
θ_1
θ_2
θ_K
I_1
I_2
I_K

Figure 1

(a)

(b)

(c)

(d)

Figure 2

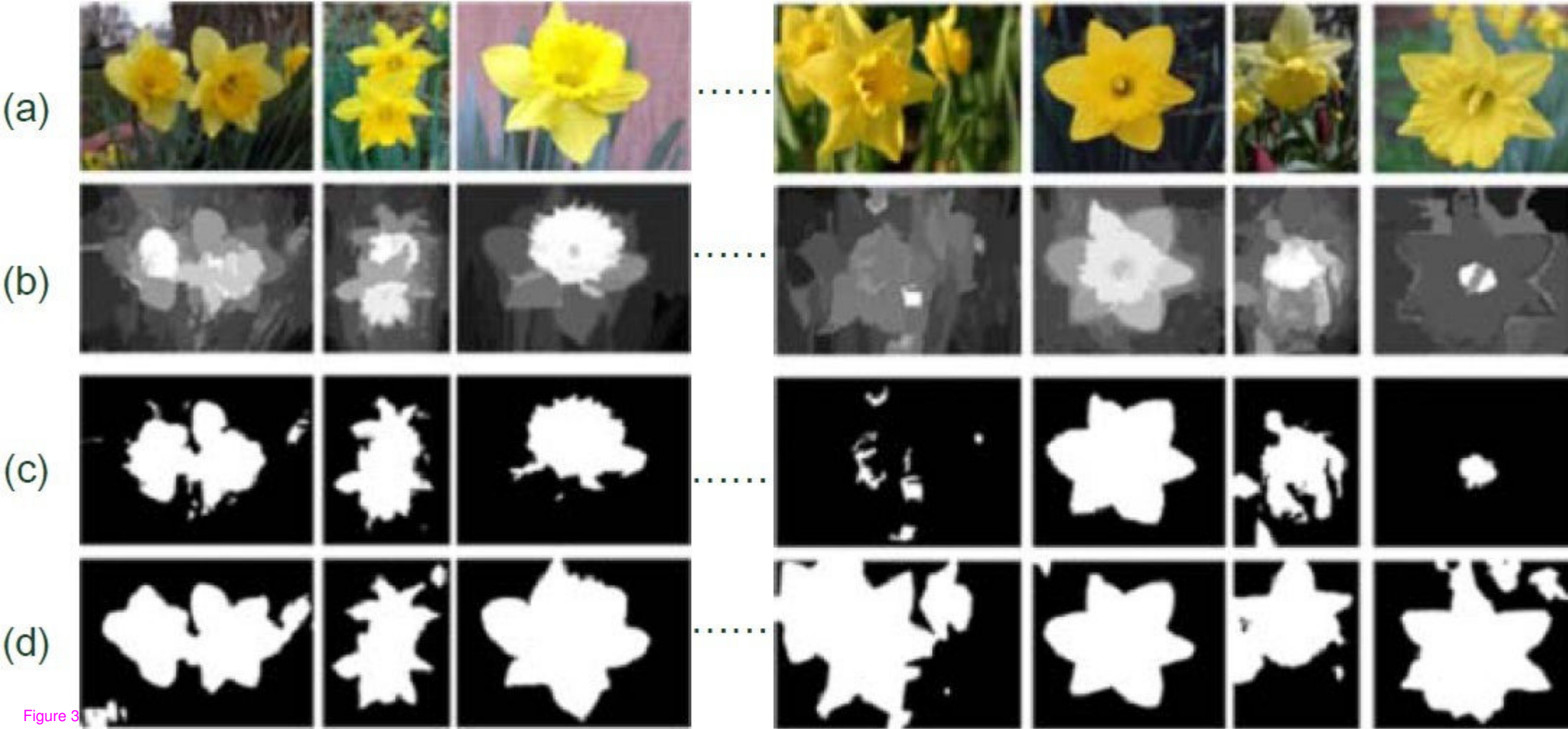
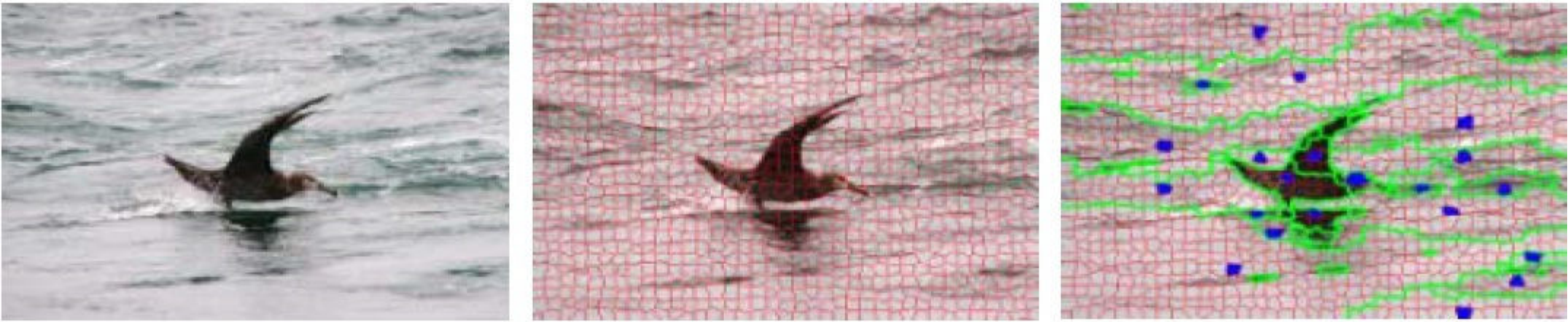


Figure 3



(a) (b) (c)

Figure 4

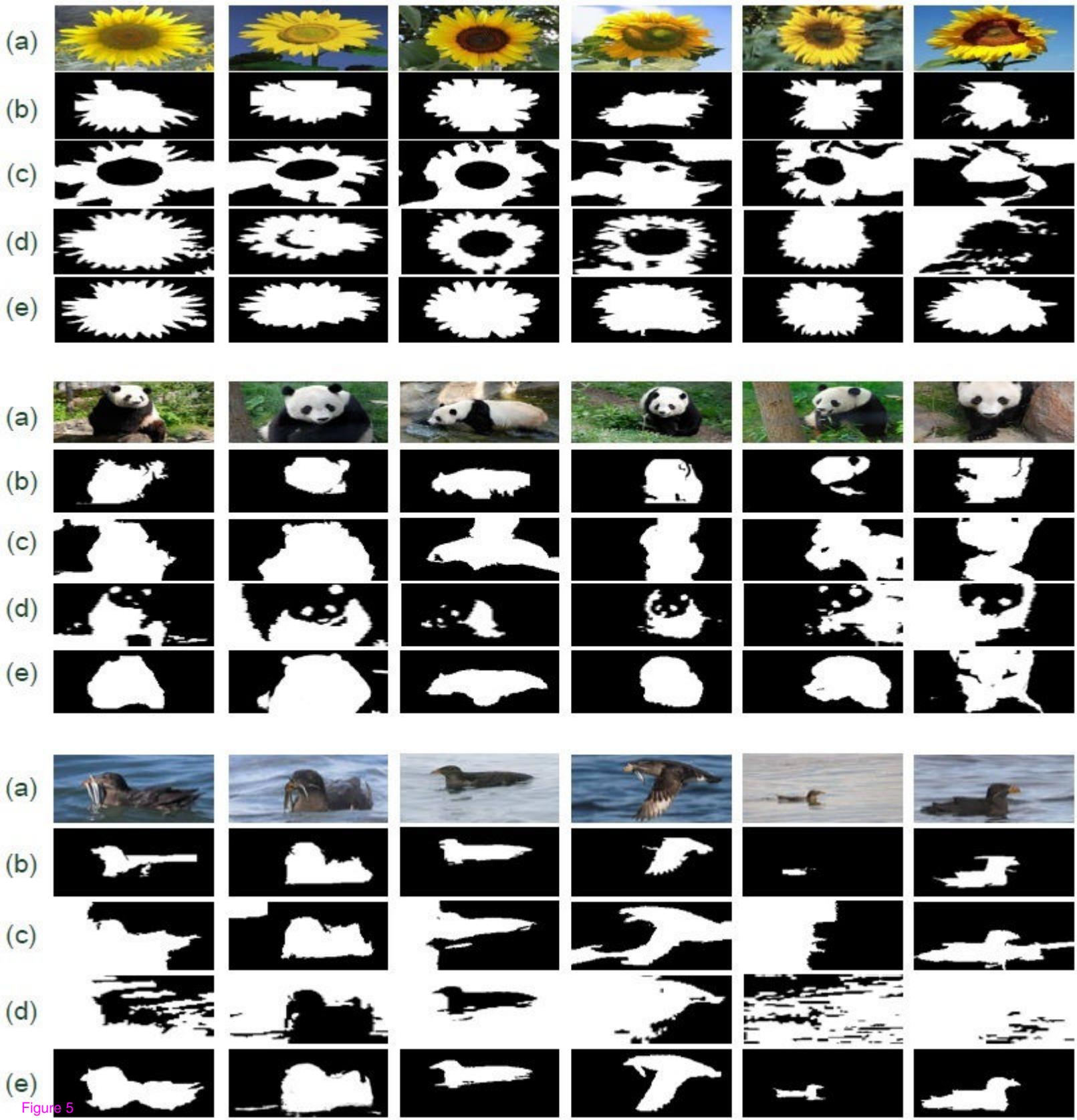


Figure 5

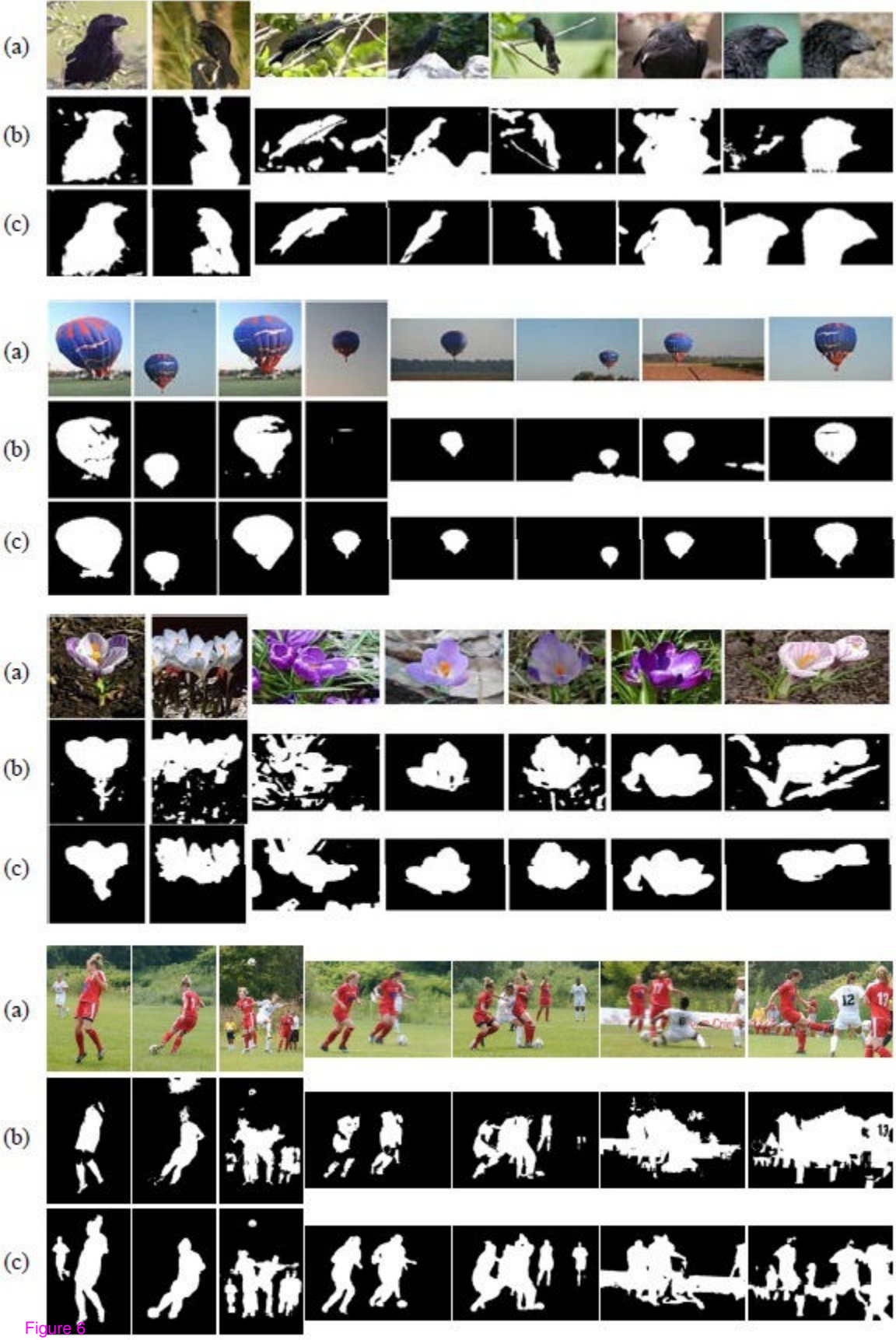


Figure 6

Figure 7