Unsupervised Object Segmentation with a Hybrid Graph Model (HGM)

Guangcan Liu, Zhouchen Lin, *Senior Member*, *IEEE*, Yong Yu, and Xiaoou Tang, *Fellow*, *IEEE*

Abstract—In this work, we address the problem of performing class-specific unsupervised object segmentation, i.e., automatic segmentation *without* annotated training images. Object segmentation can be regarded as a special data clustering problem where both class-specific information and local texture/color similarities have to be considered. To this end, we propose a hybrid graph model (HGM) that can make effective use of both symmetric and asymmetric relationship among samples. The vertices of a hybrid graph represent the samples and are connected by directed edges and/or undirected ones, which represent the asymmetric and/or symmetric relationship between them, respectively. When applied to object segmentation, vertices are superpixels, the asymmetric relationship is the conditional dependence of occurrence, and the symmetric relationship is the color/texture similarity. By combining the Markov chain formed by the directed subgraph and the minimal cut of the undirected subgraph, the object boundaries can be determined for each image. Using the HGM, we can conveniently achieve simultaneous segmentation and recognition by integrating both top-down and bottom-up information into a unified process. Experiments on 42 object classes (9,415 images in total) show promising results.

Index Terms—Segmentation, graph-theoretic methods, spectral clustering.

1 Introduction

Class-specific (or category-level) object segmentation is one of the fundamental problems in computer vision. Its goal is to segment an image into regions, with each region solely containing object(s) of a class. As object segmentation requires that each segmented region be a semantic object, it is much more challenging than traditional image segmentation [2], [3], [4], [5], [6] (we shall call this oversegmentation instead throughout this paper), which only requires that each region is a homogeneous texture. To achieve object segmentation, object classes should be represented appropriately. As the variance of object shape and color/texture within an object class can be large, it is very difficult to obtain class-specific features that can describe the object class accurately. In this regard, object segmentation is a difficult problem. However, object segmentation is feasible due to the recent development of recognition and oversegmentation techniques in computer vision. On the one hand, recently established shape models, such as the implicit shape model [7] and the sparse representation model [8], provide us with effective ways to extract the approximate shape priors of an object class. On the other hand, current oversegmentation techniques [2], [3], [4], [6] can well handle the low-level image features and produce reliable texture segmentation results. This greatly reduces the difficulty of object segmentation because we only have to consider how to group the obtained subregions. So, accurate object segmentation is possible if we can combine both class-specific (high-level) and low-level priors effectively. To achieve this, in this paper we introduce a novel spectral method called the hybrid graph model (HGM).[1]

According to whether the learning of object class priors requires human interaction or not, the existing object segmentation algorithms are broken into two categories: supervised and unsupervised. Supervised algorithms require either manually segmented masks in training images [7], [9], [10], the specification of shape templates [9], [11], [12], [13], [14], or other kinds of priors (e.g., object part configuration [15] or class fragments [16]). These algorithms may be applicable to a particular object class [13], a range of objects [10], [12], or object classes [7], [9], [11], [14], [15], [16], [17] provided that the class dependent priors are available. However, as a practical object recognition system needs to handle a large number of classes of objects and most classes may require many training samples due to significant intraclass shape and appearance variances, it is important that the learning does not involve any human interaction. This makes unsupervised algorithms more appealing. There has been sparse research in this direction. Borenstein and Ullman [18] used the overlap between automatically extracted object parts (or fragments) to determine the foreground and the background. As individual parts are considered independently, the approach is prone to wrongly judge background clutters as foreground parts. Winn and Jojic [19] combined all images together to find a consistent segmentation based on the assumption that the

- *G. Liu and Y. Yu are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, P.R. China. E-mail: roth@sjtu.edu.cn, yyu@apex.sjtu.edu.cn.*
- *Z. Lin is with the Visual Computing Group, Microsoft Research Asia, Beijing 100080, P.R. China. E-mail: zhoulin@microsoft.com.*
- *X. Tang is with the Multimedia Lab, Department of Information Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong SAR, P.R. China. E-mail: xtang@ie.cuhk.edu.hk.*

Manuscript received 27 July 2007; revised 3 July 2008; accepted 20 Jan. 2009; published online 10 Feb. 2009.
Recommended for acceptance by R. Zabih.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-07-0458.
Digital Object Identifier no. 10.1109/TPAMI.2009.40.

1. The preliminary version of this paper [1] was accepted by ICCV 2007.

Fig. 1. Our HGM-based single-class object segmentation. (a) Inputs: A set of images each consisting of objects (foreground) of a class and different backgrounds. (b) Outputs: Regions solely containing objects of the class. The whole process is fully automatic.

object shape and color distribution pattern are consistent within class and that the color/texture variability within a single object of a class is limited. Moreover, each image should only contain *one* object of the class. Rother et al. [20] showed that it is possible to use only two images to segment their common parts simultaneously. They required the common parts to have similar shape and color/texture. Russell et al. [21] segmented images in multiple ways and then borrowed techniques from document analysis to discover multiple object classes. Their assumption was that some regions in some of the segmentations are correct for each object. As segmentation precedes class discovery, it is usually hard to have accurate segmentation. Due to the limitations of these existing methods, we aim at proposing a novel unsupervised algorithm that can produce more accurate object boundaries, where the assumption on the variance of object shape and color/texture is much weaker and images can contain multiple objects.

To ensure robustness, we follow the doctrine that object segmentation should be handled in parallel to object recognition [7], [9], [14], [15], [19], [22] as they are strongly coupled problems. So, the top-down (for recognition) and the bottom-up information (for segmentation) should be utilized simultaneously. Although no annotated training images are available, as long as there are enough images, the common patterns of the object class will appear frequently and the effect of the background will fade out as it is much less structured compared to the objects. So, our target is to segment a large number of images *simultaneously* (Fig. 1). As we will not assume small intraclass shape variance (e.g., Fig. 1a), unlike [7], [9], [14], we do not expect that there will be a global shape prior for recognition. Therefore, we adopt *local* shape priors based on the work of Agarwal and Roth [8]. We first extract the object parts using an interest point detector [23]. The concurrence of object parts and the *weak* spatial relationship among them form our shape priors. The local shape priors provide very *weak* top-down constraints on the object shape, as the object parts are only sparsely distributed across the objects, and very often they also reside in the background. So, it is unlikely to obtain accurate segmentation by using such priors *only*. On the other hand, we break the images into superpixels [24] and group homogeneous superpixels into relatively large subregions [6]. Although a subregion may not exactly correspond to an object, it is nonetheless homogeneous in texture. Such trustworthy oversegmentation results provide good basis for object segmentation. For example, Cao and Fei-Fei [25] suggested to further group such subregions into objects. In contrast, we view the oversegmentation results as

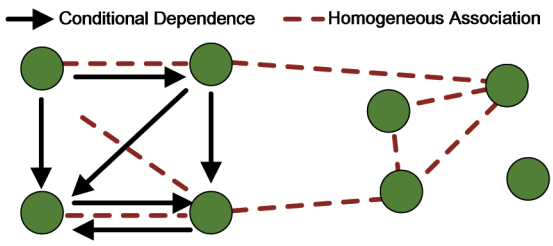


Fig. 2. An illustration of the hybrid graph. A vertex denotes a data sample (superpixel). A directed edge represents the relation of conditional dependence between a pair of samples, while an undirected edge represents the relation of homogeneous association. Between each pair of vertices, there are at most three edges: two directed edges and one undirected edge. In some scenarios, it is possible that some vertices are isolated.

trustworthy bottom-up constraints on the object appearance. Both top-down and bottom-up constraints can be represented by our HGM naturally.

The vertices of a hybrid graph (Fig. 2) represent the samples, e.g., superpixels of an image. The vertices are connected by directed edges and/or undirected ones. A directed edge represents the dependence between the vertices that it connects (which is asymmetric), while an undirected edge represents the similarity between the vertices (which is symmetric). The directed and the undirected subgraphs represent our *weak* top-down and *trustworthy* bottom-up priors, respectively. Then, an image is segmented by classifying the vertices of the hybrid graph into two clusters, with one cluster being the foreground containing object(s) of the class and another being the background. The classification is based on computing a score vector that assigns each vertex its "probability" of belonging to the underlying class. The score vector is the solution to an optimization problem that combines a random walk on the directed subgraph and a minimal cut (Min-Cut) of the undirected subgraph. The optimization problem can also be deduced from the well-established *manifold regularization* [26] framework, where the random walk defines the loss function and the Min-Cut defines the regularization function.

Compared to the previous unsupervised object segmentation algorithms [18], [19], [20], [21], the main advantages of our HGM-based method include:

- Larger variation in shape (including position, size, pose and profile) is allowed within a class.
- Larger variation in color/texture is allowed not only within class but also within object.
- Multiple objects of the same class are allowed in each image.
- More accurate output of object boundaries.
- It is fully automatic for single-class object segmentation.

The remainder of this paper is organized as follows: Section 2 introduces the general formulation of an HGM for two-class clustering. Section 3 details our HGM-based single-class object segmentation approach. Section 4 shows the experimental results. Section 5 extends our HGM to address multiclass clustering and presents the results on multiclass object segmentation. Finally, Section 6 concludes our paper.

2 The Hybrid Graph Model for Two-Class Clustering

In this section, we present the HGM for two-class clustering in an abstract sense. The extension to multiclass clustering is deferred until Section 5. Let the set of n data samples (a sample denotes a superpixel when applying HGM to object segmentation) be $V = \{v_1, \ldots, v_n\}$, with both symmetric and asymmetric relationships among them. We want to classify the samples into two classes $\mathcal{C}$ and $\bar{\mathcal{C}}$. Instead of directly assigning definite class labels, we aim at calculating a score vector $\overrightarrow{\pi} = (\pi_1, \ldots, \pi_n)^T$ and a threshold value t, where π_i is the "probability" of v_i belonging to $\mathcal{C}$. With an appropriate t, $\mathcal{C}$ can be determined as the set $V_+ = \{v_i \mid \pi_i \geq t\}$ and $\bar{\mathcal{C}}$ consists of the remaining samples $V_- = \{v_i \mid \pi_i < t\}$.

2.1 Representing Data by a Hybrid Graph

In a hybrid graph, the symmetric relationship is represented by undirected edges, while the asymmetric relationship is represented by directed edges. The asymmetric relationship is usually the conditional dependence between the samples (which incorporates weak class-specific information), while the symmetric relationship often measures the homogeneity (e.g., similarity) among the samples of a class. As a result, a hybrid graph incorporates two matrices that are associated to the directed subgraph and the undirected subgraph, respectively:

1. *Conditional Dependence Matrix P:*

   $$P = [p_{ij}]_{n\times n},$$

   where p_{ij} measures the conditional dependence of v_j on v_i. This matrix is usually asymmetric. In our object segmentation task, it represents the shape configuration between superpixels, where the shape priors are first acquired from the concurrence of object parts and then transformed to the superpixels.
2. *Homogeneous Association Matrix A:*

   $$A = [a_{ij}]_{n\times n},$$

   where a_{ij} measures the homogeneity between v_i and v_j. This matrix is symmetric. In our problem, it represents the color/texture similarity and the spatial adjacency among superpixels, which mainly come from the oversegmentation results.

Therefore, in a hybrid graph (Fig. 2) there are at most three edges between each pair of vertices: Two are directed and one is undirected. The weights assigned to directed edges and undirected ones correspond to matrix P and matrix A, respectively. So, it is convenient to denote the hybrid graph as $G = (V; P, A)$.

2.2 Computing the Score Vector

As mentioned before, π_i is the "probability" of sample v_i belonging to the class $\mathcal{C}$. Let us consider the directed subgraph and undirected subgraph one by one.

From the directed subgraph, as the samples are interdependent, the probability of sample v_i should depend on the probabilities of samples that point to it. So, the interdependence among the samples naturally forms a Markov Chain. Ideally, like PageRank [27], this results in a stationary distribution $\overrightarrow{\pi}$:

$$P^T\overrightarrow{\pi} = \overrightarrow{\pi}, \tag{1}$$

which is also the solution to

$$\min_{\overrightarrow{\pi}} \|P^T\overrightarrow{\pi} - \overrightarrow{\pi}\|^2. \tag{2}$$

On the other hand, from the undirected subgraph, if two entities v_i and v_j are strongly associated, they are more likely to belong to the same class. So, the score vector should minimize the cut cost

$$\sum_{i,j} a_{ij}(\pi_i - \pi_j)^2. \tag{3}$$

Putting the above two criteria together, we have an optimization problem to calculate the score vector $\overrightarrow{\pi}$:

$$\min_{\overrightarrow{\pi}} \mathcal{E}(G, \overrightarrow{\pi}), \text{ subject to } \overrightarrow{\pi}^T\overrightarrow{\pi} = 1, \tag{4}$$

where

$$\mathcal{E}(G, \overrightarrow{\pi}) = \|P^T\overrightarrow{\pi} - \overrightarrow{\pi}\|^2 + \alpha\sum_{i,j} a_{ij}(\pi_i - \pi_j)^2, \tag{5}$$

$G = (V; P, A)$ is the hybrid graph, and α is a positive parameter used to balance the effects of the two criteria. In our experiments, we fix $\alpha = 1$. The solution to problem (4) is the eigenvector associated to the minimum eigenvalue of the following matrix:

$$M(G) = (I - P)(I - P^T) + \alpha L_A, \tag{6}$$

where L_A is the Laplacian matrix of the undirected subgraph: $L_A = D_A - A$ with $D_A = \text{diag}\{\sum_{j=1}^n a_{1j}, \ldots, \sum_{j=1}^n a_{nj}\}$, and I is the identity matrix. Note that, if $\overrightarrow{\pi}$ is the optimal solution to (4), so is $-\overrightarrow{\pi}$. However, as π_i is the "probability" of sample v_i belonging to $\mathcal{C}$, we have to choose the $\overrightarrow{\pi}$ such that $\sum_{i=1}^n \pi_i \geq 0$.

2.3 Determining the Threshold

With the score vector $\overrightarrow{\pi}$, the samples can be classified by using a threshold t. It is natural that this threshold t be chosen as the mean of $\overrightarrow{\pi}$. However, some computed values in $\overrightarrow{\pi}$ may be negative.[2] Such negative scores make the threshold underestimated. So, we further estimate the threshold t by the geometric mean of $\overrightarrow{\pi}$. So, a reasonable estimate of t is $t_l \leq t \leq t_h$, where

$$t_l = \frac{1}{n}\sum_{i=1}^n \pi_i \quad \text{and} \quad t_h = \sqrt{\frac{1}{n}\sum_{i=1}^n \pi_i^2}.$$

As we also want the classes to be compact and isolated from each other, we borrow the normalized cut (N-Cut) [5] of the undirected subgraph as the criterion to help determine the final threshold t:

2. The negativity problem is not critical for our purpose because, actually, it is the relative order of the scores that matters. For ease of optimization, we do not enforce nonnegativity of the score vector $\overrightarrow{\pi}$ in (4).

$$t = \arg\min_{t\in[t_l,t_h]} NCut(V_+, V_-) = \arg\min_{t\in[t_l,t_h]} \frac{cut(V_+, V_-)}{assoc(V_+, V)} + \frac{cut(V_+, V_-)}{assoc(V_-, V)}, \quad (7)$$

where

$$cut(V_+, V_-) = \sum_{v_i\in V_+, v_j\in V_-} a_{ij},\ assoc(V_+, V) = \sum_{v_i\in V_+, v_j\in V} a_{ij},$$
$$\text{and } assoc(V_-, V) = \sum_{v_i\in V_-, v_j\in V} a_{ij}.$$

The optimal threshold can be found by testing t with $\overrightarrow{\pi}$. So, solving the above optimization problem just takes $O(n)$ operations, where n is the number of samples. We have found that this configuration works well in our experiments.

2.4 Discussions

2.4.1 Connection to Manifold Regularization

Although HGM is essentially a heuristic approach to assign "probabilities" of samples belonging to the target class $\mathcal{C}$, it can be fit into the well-established manifold regularization [26] framework. Let us consider the problem of learning the conditional distribution $Pr(y \mid x)$ from data samples. There is an unknown probability distribution $Pr(x, y)$ on $X \times Y$, where X is the input sample space and $Y = \{1, 0\}$ with $Y = 1$ if a sample belongs to $\mathcal{C}$ and $Y = 0$ if otherwise. Labeled samples are (x_i, y_i) pairs drawn from $Pr(y \mid x)$. Unlabeled samples are simply $x_i \in X$ drawn from the marginal distribution $Pr_X(x)$ of $Pr(x, y)$. To learn the conditional distribution $Pr(y \mid x)$ with few or even no labeled samples, the prior knowledge of the marginal distribution $Pr_X(x)$ can be exploited:

$$Pr(y \mid x) = \arg\min_{Pr(y|x)} \sum_{i=1}^{n} \mathcal{L}(y_{0i}, Pr(y \mid x_i)) + \alpha \mathcal{R}_{Pr_X}(Pr(y \mid x)),$$

where $\overrightarrow{y_0} = [y_{01}, \ldots, y_{0n}]^T$ is a label vector that is possibly known a priori and $\mathcal{L}$ is some loss function, such as the squared loss $(y_{0i} - Pr(y \mid x_i))^2$ for MSE and the hinge loss function $\max[0, 1 - y_{0i} Pr(y \mid x_i)]$ for SVM. The regularization term $\mathcal{R}_{Pr_X}(Pr(y \mid x))$ is defined according to the connection between the marginal and the conditional distributions. It is often assumed that if two points x_1 and x_2 are close to each other with respect to the intrinsic geometry prescribed by $Pr_X(x)$, then the conditional distributions $Pr(y \mid x_1)$ and $Pr(y \mid x_2)$ are similar. So, the regularization term can be defined as

$$\mathcal{R}_{Pr_X}(Pr(y \mid x)) = \sum_{i,j=1}^{n} a_{ij}(Pr(y \mid x_i) - Pr(y \mid x_j))^2,$$

where a_{ij} measures the similarity between x_i and x_j.

There are two kinds of methods to learn the conditional distribution: inductive and transductive. Inductive methods learn a function $f(x_i)$ to fit $Pr(y \mid x_i)$, while transductive methods learn a label vector directly. HGM is a kind of transductive method. It learns a label vector $\overrightarrow{\pi}$, where $\pi_i = Pr(y = 1 \mid x_i)$. Considering the squared loss function and integrating all the above ingredients, the label vector can be computed by minimizing

$$\overrightarrow{\pi} = \arg\min_{\overrightarrow{\pi}} \sum_{i=1}^{n} (y_{0i} - \pi_i)^2 + \alpha \sum_{i,j=1}^{n} a_{ij}(\pi_i - \pi_j)^2.$$

The definition of the loss function usually needs some labeled data. For example, Zhou et al. [28] used partially labeled samples and defined $y_{0k} = 0$ if x_k is unlabeled. In HGM, it could be viewed that the labels come from the random walk on the directed graph

$$y_{0i} = \hat{Pr}(y = 1 \mid x_i) \approx \sum_{j=1}^{n} Pr(y = 1 \mid x_j) Pr(x_i \mid x_j) = \sum_{j=1}^{n} \pi_j p_{ji}.$$

It is easy to see that, in this way,

$$\sum_{i=1}^{n} (y_{0i} - \pi_i)^2 \approx \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \pi_j p_{ji} - \pi_i \right)^2 = \|P^T \overrightarrow{\pi} - \overrightarrow{\pi}\|^2.$$

Therefore, (5) is naturally deduced.

So, our HGM is a kind of the regularization approach [26]. A key difference is that in HGM there is no labeled sample and we use a conditional dependence matrix P to incorporate the class-specific priors instead. It is worth noting that the Laplacian matrix L_A in (6) can be replaced by the normalized Laplacian (i.e., N-Cut of the undirected subgraph). However, we have experimentally found that using normalized Laplacian tends to cut off the high-curvature details of objects (e.g., the legs of horses). In comparison, the original Laplacian matrix always produces promising results. So, we adopt (3) to measure the cut cost.

2.4.2 Connection to Affinity Matrix Learning

Spectral methods that cluster data samples using the eigenvectors and eigenvalues of a similarity matrix have been regarded as a powerful technique for segmentation. As demonstrated in previous research, such as [5], these methods are able to produce impressive image segmentation results using simple low-level image features. However, it is difficult to combine various features by defining the edge weights of the graph. Shi et al. [29], [30] therefore interpreted spectral methods in the Markov random walk framework and turned instead to learn a directed graph G that encodes various image features, i.e., to learn the weights matrix of the graph. This technique is known as affinity matrix learning.

Our HGM can also be regarded as an extension of affinity matrix learning for object segmentation. At first, two affinity matrices P and A are constructed or learned, then the segmentation is obtained by performing eigenvector decomposition of $(I - P)(I - P^T) + \alpha L_A$, which could be viewed as the affinity matrix of HGM. However, there is a key difference: HGM uses a directed subgraph and an undirected subgraph to represent the top-down priors specific to an object class and the bottom-up priors within single images, respectively, while affinity matrix learning uses a directed graph only.

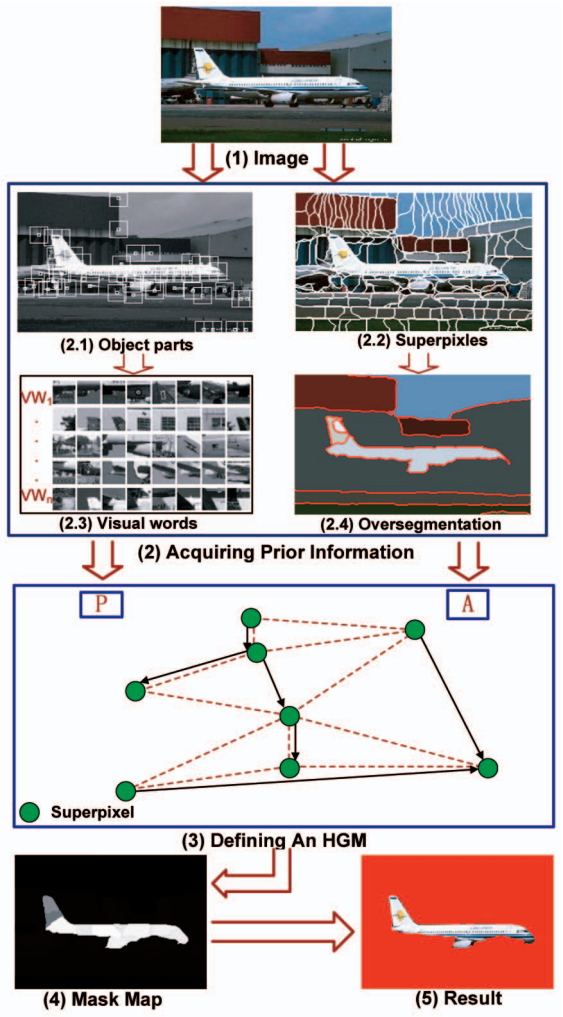


Fig. 3. Illustration of HGM-based object segmentation. Given an image (1), a mask map (4) has to be learned. To this end, we obtain object parts (2.1) using the Harris interest point detector and group the pixels into superpixels (2.2). Then, we further cluster object parts and superpixels into visual words (2.3) and midlevel oversegmentation (2.4), respectively. Next, we incorporate the acquired priors into an HGM by defining the conditional dependence matrix P according to shape priors and the homogenous association matrix A according to the color/texture priors. With the mask map (4) computed from the HGM, the image can be easily segmented (5).

3 HGM-Based Single-Class Object Segmentation

Our HGM-based object segmentation algorithm is outlined in Fig. 3. Given an image (Fig. 3(1)), we aim at learning a mask map (Fig. 3(4)) by incorporating the priors of shapes (Fig. 3(2.3)) and color/texture (Fig. 3(2.4)) into an HGM (Fig. 3(3)). Then, the mask map can be computed by the method in Section 2.2, and finally, based on the mask map, the image can be segmented (Fig. 3(5)) by thresholding the mask (Section 2.3). In the following, we describe details of each step.

3.1 Acquiring Prior Information

We first resize all images to about the same size, with the longer side being 320 pixels. Then, the remaining preprocessing procedure mainly aims at acquiring the prior information of the object class.

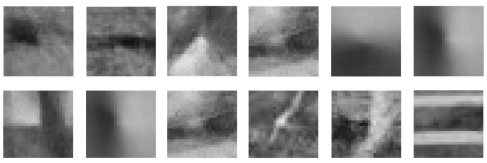


Fig. 4. Some examples of the visual words from the Horse image data set [31]. We find 245 visual words by clustering 4,025 object parts detected from 78 horse images. Not all of the visual words, e.g., the last visual word, correspond to semantic object parts.

3.1.1 Acquiring Local Shape Priors

Our local shape priors consist of visual words [21] and the spatial distances between them. A visual word is the center of a cluster of local windows that have similar appearance. It represents the whole cluster and is a feature of local appearance of an object class (e.g., the tyres of cars). The aforementioned "object part" is an instance of the cluster that a visual word represents.

Building the codebook. We follow the methods in [7], [8]. First, a number of images are randomly chosen from all provided images and are tentatively converted to gray scale. These images are considered as "special" self-training images for extracting the shape priors of the class. Second, object parts with rich textures are detected by extracting windows of size 25×25 around the points detected with the Harris interest point detector [23] (Fig. 3(2.1)). Third, all detected parts are clustered into several clusters by agglomerative clustering [7] (Fig. 3(2.3)). The similarity measure is the Normalized Gray-Scale Correlation (NGC) [7] and the threshold is set to be 0.3 in all our experiments. Then, all the cluster centers form the visual words that describe the local appearances of the images. Fig. 4 shows some visual words. Note that the visual words may *not* belong to the object class as this process does not (and could not) tell whether the object parts belong to the object region or not. For example, there are about 24 percent object parts that have *no* intersection with the foreground. Finally, the codebook consists of all the visual words. It can be refined by HGM for higher accuracy (see the Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/101109/TPAMI.2009.40).

Building the spatial relation table. As we are to address larger shape variation, unlike deformable templates [9], [11], [12], [13], [14] and implicit shape model [7], we can only assume very weak shape configurations. We hence only consider the spatial distance between visual words. By iterating over all selected images and matching visual words to all detected object parts using NGC, we have a table of the spatial relation between pairs of visual words:

$$[vw_i, vw_j, d_{ij} \sim N(\mu_{ij}, \sigma_{ij})], \tag{8}$$

where vw_i and vw_j are two visual words and $N(\mu_{ij}, \sigma_{ij})$ is a Gaussian that fits the distribution of the spatial distance d_{ij} between object parts matched to vw_i and vw_j. Fig. 5 shows some examples of the visual words pairs in the spatial relation table. Unlike [8], which also considered direction between object parts, we ignore the direction because we allow arbitrary object orientation.

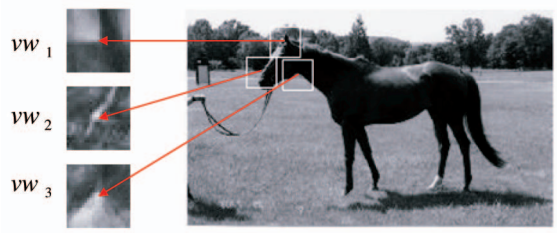


Fig. 5. An example of matching real image patches (object parts) to the visual words. For the three exemplar visual words pairs, their corresponding parameters of the Gaussian models are $(\mu_{12} = 42.08, \sigma_{12} = 17.30)$, $(\mu_{13} = 37.23, \sigma_{13} = 12.75)$, and $(\mu_{23} = 30.93, \sigma_{23} = 13.17)$, respectively.

3.1.2 Acquiring Color/Texture Priors

Color and texture are also features of objects. As object regions should consist of subregions that are homogeneous in color or texture, for computational efficiency, we shall not consider pixel-level segmentation. So, we first oversegment the images into superpixels [24] (Fig. 3(2.2)), then use the midlevel clustering algorithm proposed in [6] to group the superpixels into much larger subregions (Fig. 3(2.4)). Then, the similarity between superpixels can be measured by whether they belong to the same subregions. Using midlevel clustering results as the similarity measure is superior to directly using pairwise similarities, as in [32], because the clustering algorithm in [6] incorporates more information to judge the homogeneity of a subregion.

3.2 Learning Mask Maps via HGM

Given an image, we aim at learning a mask map that gives each superpixel a probability of lying inside object(s). Our basic notion is to integrate all of the priors into a unified framework.

3.2.1 The Hybrid Graph for Object Segmentation

Our hybrid graph $G = (V; P, A)$ for object segmentation (Fig. 3(3)) includes a vertex set V, where superpixels are the vertices, and an edge set E, implied by P and A. The shape priors are encoded in the conditional dependence matrix P of the directed subgraph, while the color/texture priors are embodied by the homogeneous association matrix A of the undirected subgraph. Next, we elaborate the definitions of these two matrices.

3.2.2 Defining Conditional Dependence Matrix P

As a vertex in the HGM denotes a superpixel rather than an object part, we need to transfer the object part based priors to superpixel based. Let $\{S_1, \ldots, S_n\}$ be the n superpixels and $\{O_1, \ldots, O_m\}$ be the m object parts (Fig. 6), then P is computed as follows:

$$P = K_1 \cdot F \cdot K_2,$$

where $K_1 = [k^1_{ij}]_{n\times m}$ and $K_2 = [k^2_{ij}]_{m\times n}$ record the overlap between superpixels and object parts:

$$k^1_{ij} = \frac{|S_i \cap O_j|}{\sum_l |S_i \cap O_l|}, \qquad k^2_{ij} = \frac{|O_i \cap S_j|}{\sum_l |O_i \cap S_l|},$$

where $|\cdot|$ is the area of a region. The matrix F is the conditional dependence matrix for the object parts. Similarly to [33], F is computed as follows:

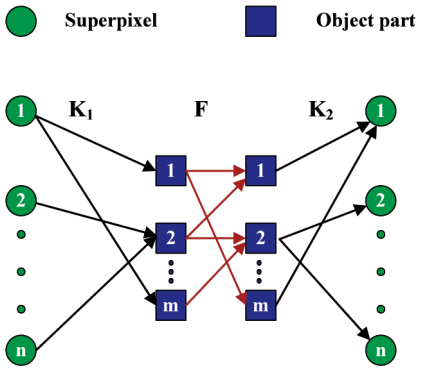


Fig. 6. Illustration of the definition of matrix P. We transfer the priors for object parts (encoded in F) to superpixels using the overlap between superpixels and object parts (encoded in K_1 and K_2).

For an object part O_i observed at location ℓ_i in an image, let $\mathbf{e}_i$ be the event of $[O_i, \ell_i]$ being observed. For an object class $\mathbb{C}$, we define f_{ij} as the conditional dependence of $\mathbf{e}_j$ on $\mathbf{e}_i$, i.e., $f_{ij} = Pr(\mathbf{e}_j \mid \mathbf{e}_i, \mathbb{C})$. With the event $\mathbf{e}_i$ fixed, $\mathbf{e}_j$ is equivalent to a new event $\tilde{\mathbf{e}}_{ij} = [O_i, O_j, d_{ij}]$ that O_j is observed at the location with distance d_{ij} from O_i. Hence,

$$f_{ij} = Pr(\mathbf{e}_j \mid \mathbf{e}_i, \mathbb{C}) \propto Pr(\tilde{\mathbf{e}}_{ij} \mid \mathbb{C}).$$

To compute f_{ij}, we have to estimate $Pr(\tilde{\mathbf{e}}_{ij}|\mathbb{C})$. By matching O_i and O_j to the codebook of the object class $\mathbb{C}$, we obtain a set of interpretations $\mathcal{I}_{ij} = \{I_{i'j'} \mid I_{i'j'}$ is the event that O_i and O_j are matched to the visual words $vw_{i'}$ and $vw_{j'}$, respectively$\}$ (i.e., O_i and O_j are interpreted as the visual words $vw_{i'}$ and $vw_{j'}$, respectively). Then,

$$\begin{aligned} Pr(\tilde{\mathbf{e}}_{ij} \mid \mathbb{C}) &= \sum_{I_{i'j'} \in \mathcal{I}_{ij}} Pr(I_{i'j'} \mid \mathbb{C}) Pr(\tilde{\mathbf{e}}_{ij} \mid I_{i'j'}, \mathbb{C}) \\ &= \sum_{I_{i'j'} \in \mathcal{I}_{ij}} Pr(I_{i'j'} \mid \mathbb{C}) Pr([vw_{i'}, vw_{j'}, d_{ij}] \mid I_{i'j'}, \mathbb{C}), \end{aligned}$$

where $Pr(I_{i'j'} \mid \mathbb{C})$ can be computed as $\frac{1}{|\mathcal{I}_{ij}|}$, assuming the independence on $\mathbb{C}$ and the equal probability of each event, and $Pr([vw_{i'}, vw_{j'}, d_{ij}] \mid I_{i'j'}, \mathbb{C})$ can be computed as

$$\frac{1}{\sqrt{2\pi}\sigma_{i'j'}} \exp\left(-\frac{(d_{ij} - \mu_{i'j'})^2}{2\sigma^2_{i'j'}}\right),$$

due to (8). Finally, normalizing the rows of F gives:

$$f_{ij} = \frac{Pr(\tilde{\mathbf{e}}_{ij} \mid \mathbb{C})}{\sum_k Pr(\tilde{\mathbf{e}}_{ik} \mid \mathbb{C})}.$$

3.2.3 Defining Homogeneous Association Matrix A

We expect the superpixels belonging to the same subregion (Fig. 3(2.4)) and/or being spatially close to each other to have similar scores. Based on this heuristics, we define

$$a_{ij} = \exp\left(-\kappa d^2_{ij}\right) + s_{ij},$$

where

$$s_{ij} = \begin{cases} 1, & \text{if } v_i \text{ and } v_j \text{ are in the same subregion,} \\ 0, & \text{otherwise,} \end{cases}$$

d_{ij} is the spatial distance between superpixels v_i and v_j and, in our experiments, κ is chosen as 0.04. Note that the magnitude of a_{ij} is dominated by s_{ij}, which results from oversegmentation.

Finally, we obtain the segmentation result (Fig. 3(5)) following the procedure in Sections 2.2 and 2.3.

4 Experiments on Single-Class Segmentation

In this section, we present the experimental results and analyze how the components P and A contribute to the final object segmentation.

4.1 Evaluation Metrics

The segmentation accuracy is the most frequently used evaluation metric. It is defined as the following:

$$accuracy = \frac{|CL_F| + |CL_B|}{|Image|}, \tag{9}$$

where CL_F and CL_B are the correctly labeled foreground and background pixels, respectively, and $Image$ is the whole image. However, a disadvantage of "accuracy" is that very bad segmentation can also obtain a promising "accuracy" if the object is small (i.e., $|CL_B| \gg |CL_F|$). So, we also consider the F-Measure, which is a popular metric used in information retrieval community:

$$F_\alpha = \frac{(1+\alpha) \cdot recall \cdot precision}{\alpha \cdot precision + recall}, \tag{10}$$

where α is the parameter to balance recall and precision and

$$recall = \frac{|A \cap B|}{|A|}, \qquad precision = \frac{|A \cap B|}{|B|},$$

where A is the ground truth mask and B is the mask output by segmentation system. The most common choice of α is 2.

4.2 Experimental Results

We apply HGM to 12 public image sets with 3,200 images in total: 10 image sets with 1,300 images are from Corel photo CDs [34], and the other two sets (Airplane and Motorbike) with 1,900 images are from Caltech-101 [35] (Table 1). Each set consists of a number of images each containing objects of the same class in a variety of positions, sizes, poses, and profiles. After tweaking on the Bird image set of Corel,[3] we fix the parameters and apply them to all experiments which are of totally different object classes. Our system automatically outputs the foreground of the images. The numbers of "special" self-training images (Section 3.1.1) for each object class are also listed in Table 1.

Fig. 7 shows some examples of segmentation results. Table 1 shows the evaluation results on all the 12 image sets, where the special self-training images are also used to compute those values because the whole segmentation process is fully automatic. For comparison, we also implement the classic N-Cut [5] algorithm that only uses low-level features for image segmentation. To see the effect of combining the symmetric and the asymmetric relationships in HGM, we also implement an approach that can be easily conceived: a Min-Cut of the directed subgraph (as inspired by [28], for example) plus the Min-Cut of the undirected subgraph. We shall call this merging of the Min-Cuts of both directed and undirected subgraphs the *naive* HGM (nHGM).

TABLE 1
Evaluation Results on 12 Object Classes

object class	# of images		F_2		
	total	*special*	*nHGM*	*N-Cut*	*HGM*
Airplane	1074	200	0.2961	0.3051	0.7609
Antique	100	60	0.5040	0.5268	0.8006
Bus	100	60	0.7200	0.6202	0.7443
Cat	100	60	0.6227	0.5904	0.7609
Dinosaur	100	60	0.4256	0.4404	0.9173
Dog	200	60	0.6070	0.5540	0.6851
Eagle	100	60	0.2434	0.2800	0.7953
Leopard	100	60	0.4224	0.4425	0.6996
Motorbike	826	100	0.6696	0.5898	0.8123
Old Car	200	60	0.6613	0.5830	0.7100
Owl	100	60	0.4707	0.4772	0.8201
Plane	200	60	0.3502	0.3598	0.7906
Average	–	–	**0.4804**	**0.4583**	**0.7743**

"# special images" refers to the number of "special" self-training images for extracting the shape prior (Section 3.1.1).

Comparison results. One can see from Table 1 that the performance of HGM is quite satisfactory, while N-Cut and nHGM cannot perform well on these segmentation tasks. For N-Cut, the main reason is that it does not use any class-specific information that is indispensable for segmenting objects of a class. For nHGM, as the densities of A and P are quite different (see Fig. 8) and the dense part will suppress the sparse one, the top-down priors in P will be ineffective if P is merged with A.

Cao and Fei-Fei [25] recently reported the segmentation accuracy on 28 image classes from the Caltech101 data set. They selected 30 images for each class and their average segmentation accuracy is 0.67. In comparison, HGM segments all the 6,200 color images[4] and gains an average accuracy of 0.74. Fig. 9 shows the details on each class.

We also apply HGM to two object classes (side view of Cars and Horses) that have been used by LOCUS [19].[5] The shape variation within class and/or color/texture variation within objects in these two image sets are smaller than those in the sets we have just presented above. For comparison, we quote results from Borenstein et al. [31], which require 54 hand segmented training data for the Horse image set, and LOCUS [19], which is also an unsupervised object

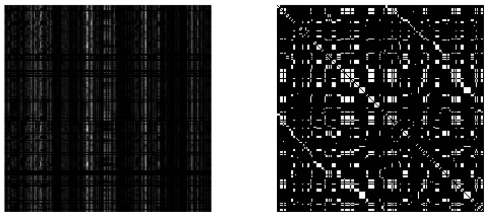
3. Parameter selection is unavoidable, as in those unsupervised algorithms [18], [20], [21], [19].

4. By removing the gray-scale images, there are 6,200 color images in total.

5. As listed in Section 1, we are only aware of four papers on unsupervised object segmentation [18], [20], [21], [19]. However, Rother et al. [20] and Russell et al. [21] actually address slightly different problems from ours. So, we mainly focus on comparing with LOCUS [19], which was claimed to be more accurate than [18].

Fig. 7. Some examples of single-class segmentation results of the 12 object classes. Each row is from the same class. For each group of images: (left) result of nHGM; (middle) result of N-Cut; (right) result of HGM. Note here that HGM can specify the regions that contain object(s) of the desired class, while nHGM and N-Cut can only determine the segmentation boundaries.

segmentation algorithm. For the Horse data set, the number of images is 328. As both baselines only reported results on 200 images and we are unable to know which images they used, we report the average accuracy of first 200 images (the average accuracy across all of 328 images is 0.954). As shown in Table 2, HGM achieves higher segmentation



(a) (b)

Fig. 8. Visualization of matrices P and A. The matrices are constructed in the segmentation task of Fig. 10. The magnitudes of both matrices have been normalized to the range between 0 and 1. (a) The density of matrix P. (b) The density of matrix A.

accuracies than LOCUS. This benefits from the oversegmentation algorithm [6] (Fig. 3(2.4)) that HGM adopts, which preserves boundaries of homogeneous color/texture during its grouping process. On the other hand, the extra s_{ij} defined in (9) encourages HGM to segment images along these boundaries. Notice that HGM segments images *fully* automatically. In contrast, as mentioned in [19], LOCUS requires some effort in choosing some images (without segmentation) to learn a class model. And LOCUS also needs some easy manual work like flipping asymmetric objects to face a consistent direction.[6]

As an unsupervised approach, HGM is also comparable to the state-of-the-art supervised approaches tested on the Horse data set. For examples, Levin and Weiss [22] obtained accuracies over 0.95 as more fragments were learned, and Cour and Shi [17] gained an accuracy of 0.942.

So, HGM is competitive to recently established approaches.

4.3 Analysis

4.3.1 Dissection of HGM

To see how HGM produces promising segmentation results in the case that there exists *large* variation in shape (including position, size, pose, and profile) within an object class and *no* annotated training images are available, we refer to Fig. 10. The task is to segment an airplane from the background clutters. We define the matrix P and A by the procedures in Sections 3.2.2 and 3.2.3, respectively. As the object is relatively small and there exist heavy background clutters, it is hard to get accurate segmentation by using the local shape priors incorporated by matrix P only (Fig. 10b). In order to obtain more accurate object segmentation, we oversegment the image into 27 subregions by using the method in [6] (Fig. 10c). Then, we use the oversegmentation result to define our homogenous association matrix A according to (9). Fig. 10d shows that we can obtain a much better mask map by combing both P and A. To see the individual strengths of P and A, one may refer to Fig. 8 that visualizes their densities. Although matrix A is much denser than matrix P, from Figs. 10b and 10d, one can see that both P and A contribute to the final performance in a cooperative way: The *weak* shape priors encoded in P take the role of finding *parts* of the desired object *approximately*, and the *trustworthy* texture/color priors encoded in A are to help find the whole object by enforcing the superpixels in the same subregion to have the same probability of belonging to the foreground. Fig. 11 shows some examples of the Horse data set. One may notice that there are relatively fewer background noises in the Horse data set (see the second column of Fig. 11). This is because the shape variation within class and/or color/texture variation within objects in this data set are relatively smaller. So, although HGM does not require the priors encoded in P to be accurate, the results will be better if one can obtain more accurate class-specific information.

One may notice that HGM depends on the performance of oversegmentation [6]. So, the basic motivation of HGM is similar to the tradition of performing object segmentation on the basis of oversegmentation. However, instead of directly constructing descriptors for each subregion and selecting some subregions to further group into objects as did in [25], HGM regards the oversegmentation results as the interaction between superpixels: some pairs of superpixels should belong to the same region simultaneously and some should not. And the weak shape priors encoded in P also reflect such interaction. So, HGM can handle size, color, texture, profile, and pose variations better than those methods that focus on the objects *only*.

4.3.2 Impact of Oversegmentation

As the matrix A heavily relies on oversegmentation, it is necessary to investigate the impact of oversegmentation. Let "#subregion" be the number of subregions left in an image after oversegmentation. It can be seen from Fig. 12 that if oversegmentation is less successful, i.e., the number of subregions is large, object segmentation is also less accurate. This testifies to the importance of low-level image features in object segmentation.

5 Extension to Multiclass Clustering

Now we extend our HGM for more general data clustering and apply it to unsupervised multiclass object segmentation, which aims at automatically segmenting a set of images containing object(s) of several classes.

5.1 HGM-Based Data Clustering

Previous clustering algorithms, such as K-means [36], fuzzy C-means [37], and spectral clustering [38] usually have only one distance metric to encode all the information among data. All information, such as the spatial continuity and color/texture similarity in image segmentation [5], has to be merged into a single scalar "distance." This merging requires an artificial definition of the "distance" that combines different categories of quantities. And, as much of the extra information is lost, accurate clustering is often difficult. Recently, there has been some work considering (semisupervised) classification on a directed graph [28], [39], [40] where the relationship between samples is asymmetric, e.g., the hyperlink between Web pages. We find that, using an HGM, we can naturally deal with asymmetric and symmetric relationships at the same time and make better use of different sources of information, rather than awkwardly reducing them into a scalar "distance."

Let $G = (V; P, A)$ be the hybrid graph that represents n samples and the pairwise relationship among them. We want to partition V into K disjoint subsets such that each

6. For comparison, please notice the different facing directions of objects in Fig. 7.

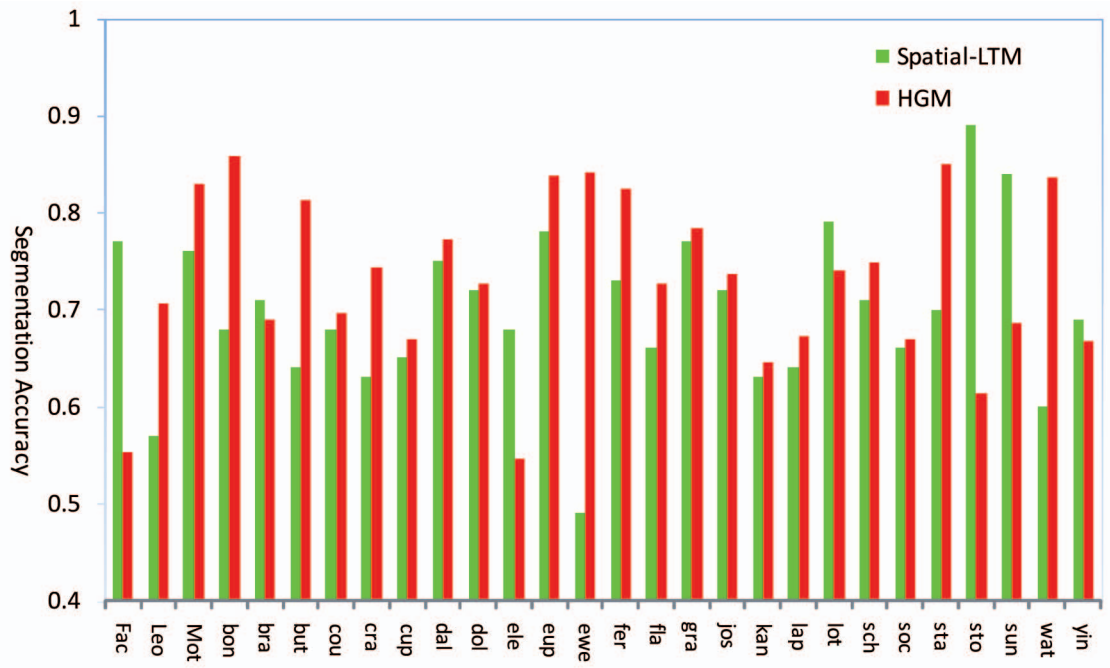


Fig. 9. Comparison of HGM and Spartial-LTM [25] on the 28 classes from the Caltech101 data set. The horizontal axis shows the abbreviated names of the classes and the vertical axis represents the average segmentation accuracy. Note here that we test on all the 6,200 color images, whereas Cao and Fei-Fei [25] only select 30 images for each class.

subset corresponds to a cluster. We may use an extra adjacency matrix $C = [c_{ij}]_{n\times n}$ to record the information of samples being in the same cluster:

$$c_{ij} = \begin{cases} 1, & \text{if } v_i \text{ and } v_j \text{ belong to the same cluster,} \\ 0, & \text{otherwise.} \end{cases}$$

This amounts to adding extra undirected edges to graph G (or adjusting the weights if the undirected edges already exist). Denote this new graph by $G_C = (V; P, A + C)$, which has the same conditional dependence matrix P and a new homogeneous association matrix $A + C$. We have to determine an optimal C such that the objective function $\mathcal{E}(G_C, \overrightarrow{\pi})$ defined in (5) is minimized over all possible C and $\overrightarrow{\pi}$.

Note that, with fixed C, the minimal value of $\mathcal{E}(G_C, \overrightarrow{\pi})$ is exactly the minimal eigenvalue $\lambda_0(G_C)$ of $M(G_C)$, where $M(\cdot)$ is defined in (6). So, we have to find C that minimizes $\lambda_0(G_C)$. Unfortunately, computing $\lambda_0(G_C)$ for all possible clusterings is prohibitive. To make the optimization tractable, we adopt a greedy grouping algorithm (Algorithm 1). At the beginning, each sample v_i is a cluster. And, in each iteration, a pair of clusters S_i and S_j are merged if they are adjacent and the increase in $\lambda_0(G_C)$ due to merging S_i and S_j is minimal. The iteration ends when there are only K groups left.

TABLE 2
Comparison with Borenstein et al. [31] (Supervised) and LOCUS [19] (Unsupervised) on the Two Image Sets They Used

object class	*# of images*	"accuracy" (defined in Equation (9))		
		Borenstein et al.	LOCUS	HGM
Car	50	-	0.914	0.958
Horse	200	0.936	0.931	0.961

Algorithm 1. Greedy Grouping

1: **Inputs:** the data $V = \{v_1, \ldots, v_n\}$, a matrix P, a matrix A and a positive integer K.
2: initialize $\mathcal{C} := \{\{v\} \mid v \in V\}, G := (V; P, A)$ and $\overrightarrow{\pi} := \arg\min_{\overrightarrow{\pi}}(G, \overrightarrow{\pi})$.
3: **while** $|\mathcal{C}| > K$ **do**
4: choose different and adjacent groups S_1 and S_2, such that $\lambda_0(G_{\mathcal{C}})$ is minimal after merging S_1 and S_2.
5: $\mathcal{C} := (\mathcal{C} \setminus \{S_1, S_2\}) \cup \{S_1 \cup S_2\}$
$G := (V; P, A + C)$ and $\overrightarrow{\pi} := \arg\min_{\overrightarrow{\pi}}(G, \overrightarrow{\pi})$
6: **end while**
7: **return** $\mathcal{C}$ and $\overrightarrow{\pi}$

5.2 Efficient Computation by Approximation

The inverse power method [41] is very efficient in computing the minimal eigenvalue and the corresponding eigenvector of a symmetric matrix. However, during clustering $\lambda_0(G_C)$ should be computed for every pair of clusters at each iteration, resulting in unaffordable computation load, especially at the beginning of the grouping when the number of clusters is large. Fortunately, at the early stage of clustering, C only changes slightly; so, $\lambda_0(G_C)$ also changes slightly as it is a continuous function of C. Therefore, we may approximate $\lambda_0(G_C)$ by

$$\lambda_0(G_C) \approx \tilde{\lambda}_0(G_C) = \lambda_0(G) + \delta(C), \tag{11}$$

where

$$\delta(C) = \sum_{i,j} c_{ij}(\pi_0(G)_i - \pi_0(G)_j)^2$$

is the cut of the undirected subgraph when the weight is C and the vertex values are $\overrightarrow{\pi}_0(G) = (\pi_0(G)_1, \ldots, \pi_0(G)_n)^T$.

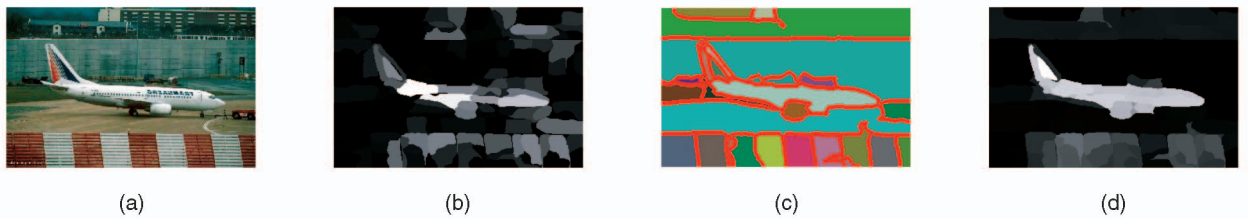


Fig. 10. The evolutions of the score vector (mask map) during a segmentation process. (a) An image. (b) The mask map estimated by using P only. (c) Results of the oversegmentation operator. (d) The mask map estimated by using both P and A.

So, we just need to check $\delta(C)$ for the optimal C, which is computationally efficient. When $\delta(C)$ gets large, the approximation is no longer reliable. So, the approximation stops when $\delta(C) > t_\delta$, where t_δ is a threshold. At this moment, usually there are only a small number of clusters left. So, the inverse power method is affordable. In our experiments, we have found that the approximation (11) is reliable when choosing $t_\delta = 1.0$.

5.3 Experimental Results

In this section, we present the experimental results of using the extended HGM for unsupervised multiclass object segmentation. Given a set of images containing objects of several classes, we follow the same procedure in Section 3 to construct the HGM. In this case, the visual words come from all the object classes and even the backgrounds, just as the single-class case and the spatial relation table is built alike. To segment a new image that may contain object(s) of several classes, we use the clustering algorithm described above to cluster the superpixels into K regions, $K-1$ of which each containing instance(s) of an object class (the remaining one region may be another object class or the background that is unstructured across all the images). We assume that K is known a priori for each test image because it has been widely recognized that it is difficult for a clustering algorithm to reliably estimate the number K of clusters.

One should be reminded that, in the case of multiclass segmentation, HGM cannot assign class labels to each cluster because the visual words are not classified among the object classes. While, in single-class segmentation, since the spatial relation table actually records the shape configuration of *one* object class and some noise due to the background, HGM can recognize the region(s) of the class.

We apply our approach to MSRC image data set [42] which contains 23 object classes and has 591 images in total. We simply feed all images into our system, rather than manually sorting the images into different sets according to their objects and then applying our system one by one. And we have to tell the computer how many regions each image should be broken into (to make this automated, the computer simply reads the number of regions from the ground truth mask). The examples in Fig. 13 demonstrate that our approach achieves more meaningful segmentation results than N-Cut does, which just utilizes the low-level information within each image. Please observe that, for different images, different objects are segmented (but no object labels are assigned).

To quantitatively evaluate the segmentation results, we define an error measure as follows:

$$error = \frac{1}{log_2 K} \sum_{i=1}^{K} w_i \cdot entropy(C_i),$$

where C_i is the ith cluster of the segmentation result and $entropy(C_i)$ is the inconsistency between the segmentation and the ground truth mask, defined as

$$entropy(C_i) = -\sum_{j=1}^{K} q_{ij} \log_2 q_{ij},$$

in which $q_{ij} = |C_i \cap T_j|/|C_i|$ and T_j is the jth cluster in the ground truth mask. $entropy(C_i)$ actually measures how scattered the cluster C_i distributes across the ground truth clusters. And w_i is the weight of C_i; it is computed by

$$w_i = \frac{|C_i|^{-1}}{\sum_{j=1}^{K} |C_j|^{-1}},$$

which is inversely proportional to the area of the cluster C_i. We give larger weights to smaller clusters because this can better measure the accuracy of separating small objects from a large background. Otherwise, the segmentation accuracy of large objects or a large background will dominate the error; one can easily see that $0 \leq error \leq 1$. Using our error measure, we can compute the segmentation error for every image. Then, the average error of our approach on MSRC data set is 0.2992, while that of N-Cut is 0.5001.

6 Conclusion

In this paper, we propose the HGM for performing class-specific object segmentation without annotated training images. The core is a general learning algorithm based on the hybrid graph topology. Object segmentation is achieved by combining top-down and bottom-up information naturally: We first obtain local shape priors of object class(es) (top-down) and color/texture priors of each image (bottom-up), then use an HGM to integrate them into a unified framework. We demonstrate our algorithm on two tasks: unsupervised single-class and multiclass object segmentation. They are both based on minimizing an energy function that combines the random walk on the directed subgraph and the Min-Cut on the undirected subgraphs.

It is worth noting that HGM is a general framework. It can be applied to various problems as long as the meanings of the graph vertices, the relationship represented by the directed/undirected edges and the two matrices P and A can be interpreted appropriately (see an example in the Appendix, which can be found on the Computer Society

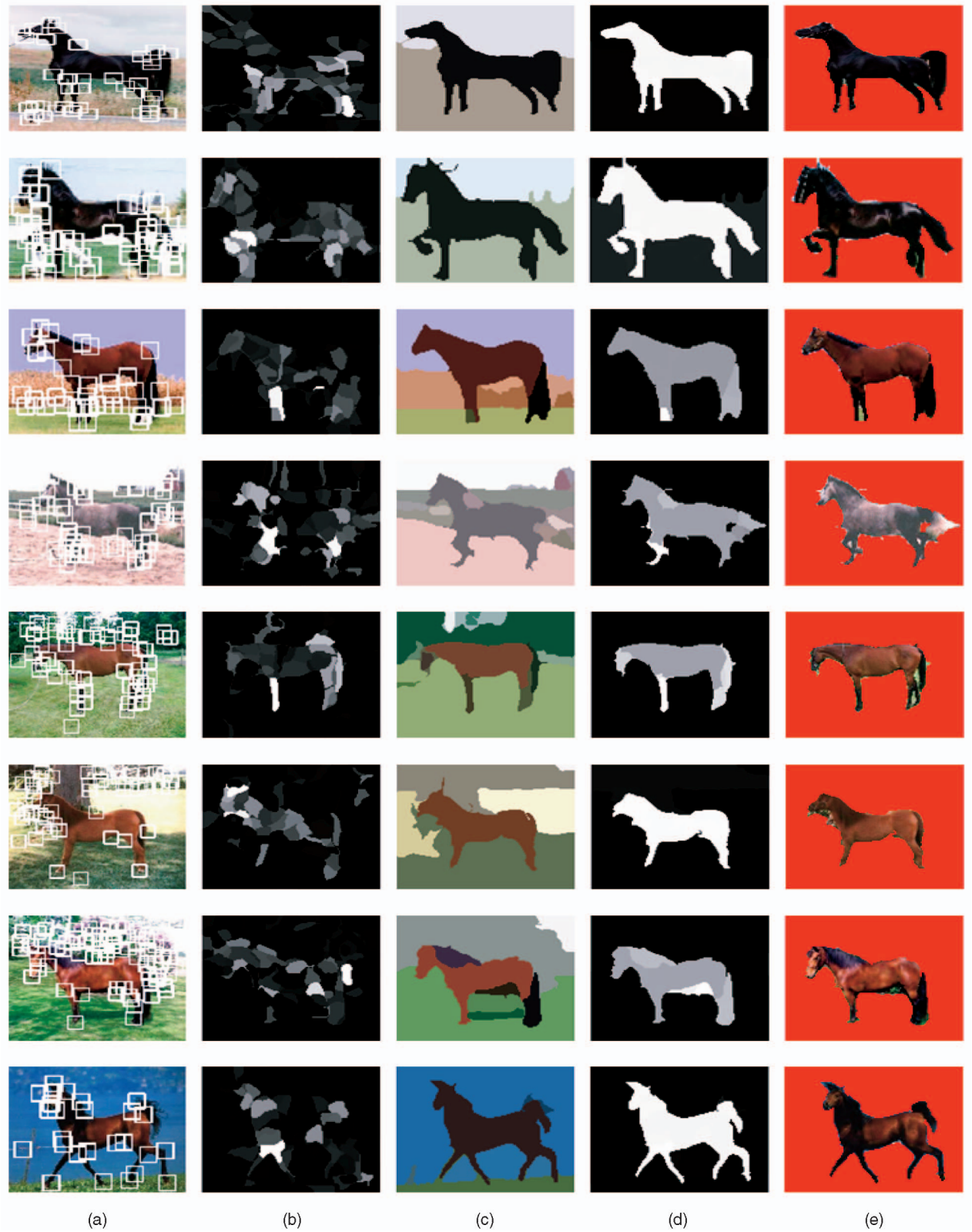


Fig. 11. Some segmentation results, including the intermediate results, of the Horse data set. (a) The image and its detected object parts; (b) the mask map estimated by using P only; (c) result of the oversegmentation operator; (d) the mask map estimated by using both P and A; (e) the final segmentation result.

Digital Library at http://doi.ieeecomputersociety.org/101109/TPAMI.2009.40). As a clustering algorithm, HGM is able to make effective use of two kinds of prior information (asymmetric and symmetric relationship) simultaneously. This characteristic makes HGM attractive in semantic data clustering.

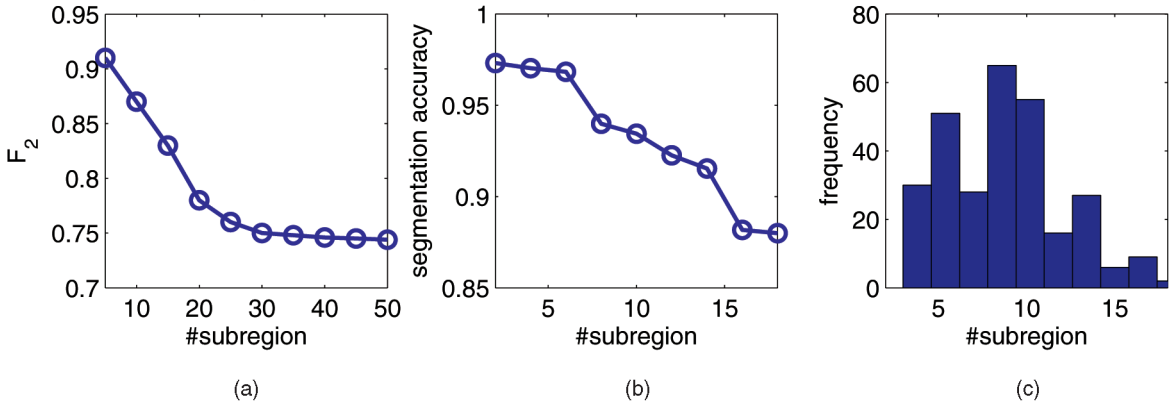


Fig. 12. The impact of oversegmentation. (a) The curve of F_2 as a function of "#subregion." These statistics are collected from 3,200 images (see the beginning of Section 4.2). (b) The curve of "segmentation accuracy" as a function of "#subregion." These statistics are collected from 328 Horse images. (c) The histogram of "#subregion" of the Horse data set. We ignore the subregions whose areas are less than 1 percent of an image.

REFERENCES

[1] G. Liu, Z. Lin, X. Tang, and Y. Yu, "A Hybrid Graph Model for Unsupervised Object Segmentation," *Proc. 11th IEEE Int'l Conf. Computer Vision,* pp. 1-8, 2007.

[2] E. Sharon, A. Brandt, and R. Basri, "Segmentation and Boundary Detection Using Multiscale Intensity Measurements," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 469-476, 2001.

[3] M. Galun, E. Sharon, R. Basri, and A. Brandt, "Texture Segmentation by Multiscale Aggregation of Filter Responses and Shape Elements," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 716-723, 2003.

[4] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, "Hierarchy and Adaptivity in Segmenting Visual Scenes," *Nature,* vol. 442, no. 7104, pp. 810-813, June 2006.

[5] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.

[6] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of Multivariate Mixed Data via Lossy Data Coding and Compression," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 9, pp. 1546-1562, Sept. 2007.

[7] B. Leibe, A. Leonardis, and B. Schiele, "Combined Object Categorization and Segmentation with an Implicit Shape Model," *Proc. Workshop Sixth European Conf. Computer Vision,* pp. 17-32, 2004.

[8] S. Agarwal and D. Roth, "Learning a Sparse Representation for Object Detection," *Proc. Seventh European Conf. Computer Vision,* pp. 113-130, 2002.

[9] Z. Tu, X. Chen, A.L. Yuille, and S.C. Zhu, "Image Parsing: Unifying Segmentation, Detection, and Recognition," *Proc. Toward Category-Level Object Recognition,* pp. 545-576, 2006.

[10] S.X. Yu and J. Shi, "Object-Specific Figure-Ground Segregation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 39-45, 2003.

[11] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active Shape Models, Their Training and Application," *Computer Vision and Image Understanding,* vol. 61, no. 1, pp. 38-59, 1995.

[12] H.E.A.E. Munim and A.A. Farag, "A Shape-Based Segmentation Approach: An Improved Technique Using Level Sets," *Proc. 10th IEEE Int'l Conf. Computer Vision,* pp. 930-935, 2005.

[13] A.L. Yuille, P.W. Hallinan, and D.S. Cohen, "Feature Extraction from Faces Using Deformable Templates," *Int'l J. Computer Vision,* vol. 8, no. 2, pp. 99-111, 1992.

[14] L. Zhao and L.S. Davis, "Closely Coupled Object Detection and Segmentation," *Proc. 10th IEEE Int'l Conf. Computer Vision,* pp. 454-461, 2005.

[15] S.X. Yu, R. Gross, and J. Shi, "Concurrent Object Recognition and Segmentation by Graph Partitioning," *Proc. Neural Information Processing Systems,* pp. 1383-1390, 2002.

[16] E. Borenstein and S. Ullman, "Class-Specific, Top-Down Segmentation," *Proc. Seventh European Conf. Computer Vision,* pp. 109-124, 2002.

[17] T. Cour and J. Shi, "Recognizing Objects by Piecing Together the Segmentation Puzzle," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2007.

[18] E. Borenstein and S. Ullman, "Learning to Segment," *Proc. Eighth European Conf. Computer Vision,* pp. 315-328, 2004.

[19] J.M. Winn and N. Jojic, "Locus: Learning Object Classes with Unsupervised Segmentation," *Proc. 10th IEEE Int'l Conf. Computer Vision,* pp. 756-763, 2005.

[20] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of Image Pairs by Histogram Matching—Incorporating a Global Constraint into MRFs," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 993-1000, 2006.

[21] B.C. Russell, W.T. Freeman, A.A. Efros, J. Sivic, and A. Zisserman, "Using Multiple Segmentations to Discover Objects and Their Extent in Image Collections," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1605-1614, 2006.

[22] A. Levin and Y. Weiss, "Learning to Combine Bottom-Up and Top-Down Segmentation," *Proc. Ninth European Conf. Computer Vision,* pp. 581-594, 2006.

[23] C. Harris and M. Stephens, "A Combined Corner and Edge Detection," *Proc. Fourth Alvey Vision Conf.,* pp. 147-151, 1988.

[24] G. Mori, "Guiding Model Search Using Segmentation," *Proc. 10th IEEE Int'l Conf. Computer Vision,* pp. 1417-1423, 2005.

[25] L. Cao and L. Fei-Fei, "Spatially Coherent Latent Topic Model for Concurrent Segmentation and Classification of Objects and Scenes," *Proc. 11th IEEE Int'l Conf. Computer Vision,* pp. 1-8, 2007.

[26] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples," *J. Machine Learning Research,* vol. 7, pp. 2399-2434, 2006.

[27] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems,* pp. 107-117, 1998.

[28] D. Zhou, J. Huang, and B. Schölkopf, "Learning from Labeled and Unlabeled Data on a Directed Graph," *Proc. 22nd Int'l Conf. Machine Learning,* pp. 1036-1043, 2005.

[29] M. Meila and J. Shi, "Learning Segmentation by Random Walks," *Proc. Neural Information Processing Systems,* pp. 873-879, 2000.

[30] T. Cour, N. Gogin, and J. Shi, "Learning Spectral Graph Segmentation," *Proc. 10th Int'l Workshop Artificial Intelligence and Statistics,* 2005.

[31] E. Borenstein, E. Sharon, and S. Ullman, "Combining Top-Down and Bottom-Up Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2004.

[32] J. Puzicha, T. Hofmann, and J.M. Buhmann, "Non-Parametric Similarity Measures for Unsupervised Texture Segmentation and Image Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 267-272, 1997.

[33] R. Fergus, P. Perona, and A. Zisserman, "A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 380-387, 2005.

[34] "Corel Photo Library," Corel Corp., 2006.

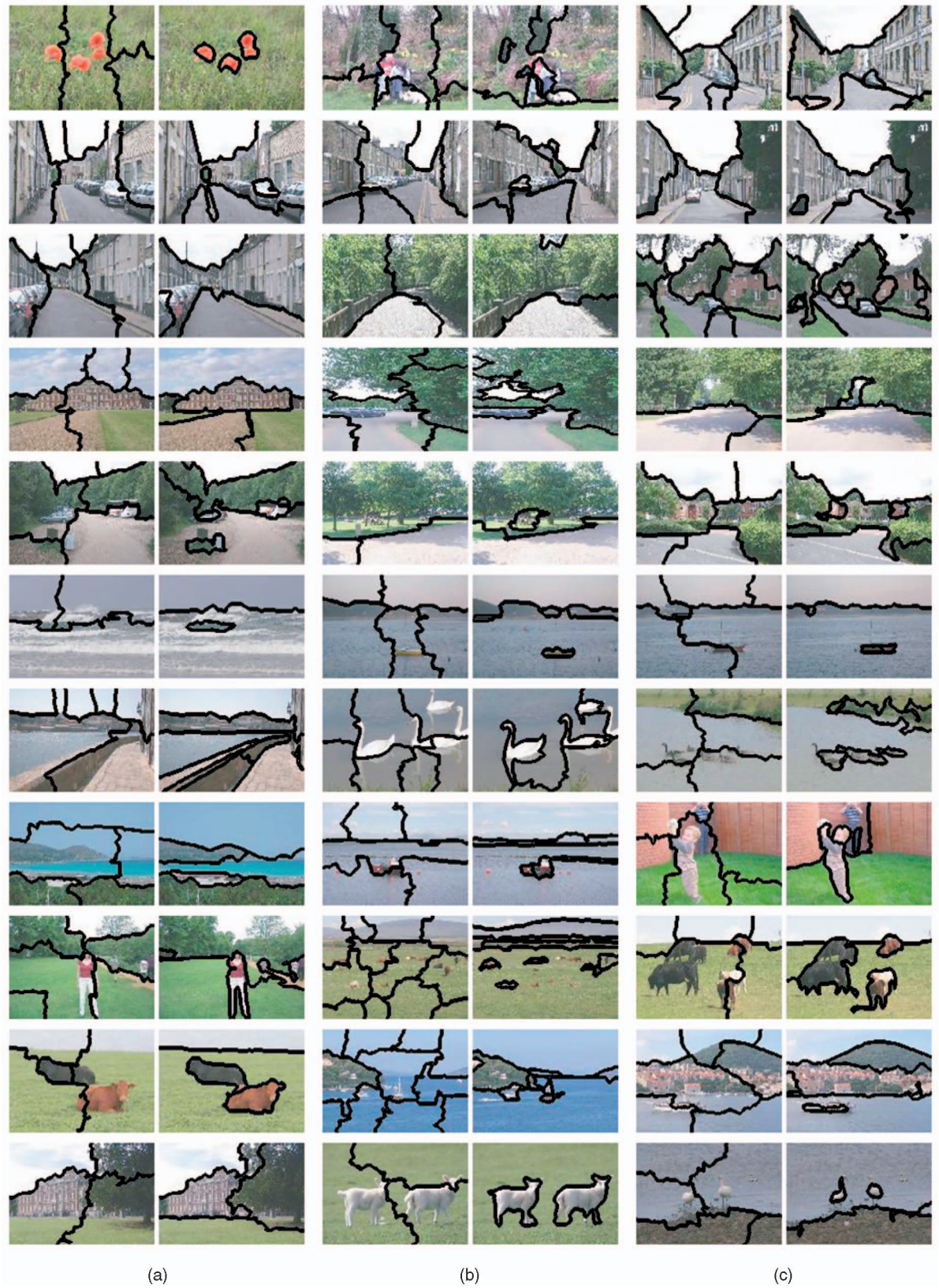
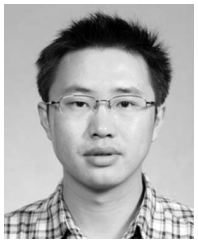


Fig. 13. Some multiclass segmentation results on the MSRC data set. For each group of images: (Left) Result of N-Cut; (Right) Result of HGM. The number K of clusters in each image is obtained by checking the ground truth mask of the image.

[35] L. Fei-Fei, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[36] J.B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. Fifth Berkeley Symp. Math. Statistics and Probability*, pp. 281-297, 1967.

[37] J.C. Dunn, "A Fuzzy Relative of the Isodata Process and Its Use in Detecting Compact Well-Separated Clusters," *J. Cybernetics,* vol. 3, pp. 32-57, 1973.
[38] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm." *Proc. Neural Information Processing Systems,* pp. 849-856, 2001.
[39] M. Szummer and T. Jaakkola, "Partially Labeled Classification with Markov Random Walks," *Proc. Neural Information Processing Systems,* pp. 945-952, 2001.
[40] N. Tishby and N. Slonim, "Data Clustering by Markovian Relaxation and the Information Bottleneck Method," *Proc. Neural Information Processing Systems,* pp. 640-646, 2000.
[41] G.H. Golub and C.F.V. Loan, *Matrix Computations,* second ed. Johns Hopkins Press, 1989.
[42] J. Shotton, J.M. Winn, C. Rother, and A. Criminisi, "TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation," *Proc. Sixth European Conf. Computer Vision,* pp. 1-15, 2006.



Guangcan Liu received the bachelor's degree in applied mathematics from Shanghai Jiao Tong University in 2004. He is currently a PhD student in the Department of Computer Science and Engineering at Shanghai Jiao Tong University and is also a visiting student at Microsoft Research Asia. His research interests include machine learning and computer vision.

Zhouchen Lin received the PhD degree in applied mathematics from Peking University in 2000. He is currently a researcher in the Visual Computing Group at Microsoft Research Asia. His research interests include computer vision, computer graphics, pattern recognition, machine learning, and numerical computation. He is a senior member of the IEEE.

Yong Yu received the master's degree from the Computer Science Department of East China Normal University. He began working at Shanghai Jiao Tong University in 1986. Now he is the vice president of the Department of Computer Science and Engineering, a PhD candidate tutor, and the chairman of the E-Generation Technology Research Center (SJTU-IBM-HKU). He was the teacher of the course "Computer Graphics and Human-Machine Interface" and the course "Next Generation Web Infrastructure." As the head coach of the SJTU ACM-ICPC team, he and his team won the 2002 and 2005 ACM-ICPC Championships. His research interests include semantic Web, Web mining, information retrieval, and computer vision.

Xiaoou Tang received the BS degree from the University of Science and Technology of China, Hefei, in 1990 and the MS degree from the University of Rochester, New York, in 1991. He received the PhD degree from the Massachusetts Institute of Technology, Cambridge, in 1996. He is a professor in the Department of Information Engineering at the Chinese University of Hong Kong. He worked as the group manager of the Visual Computing Group at Microsoft Research Asia from 2005 to 2008. His research interests include computer vision, pattern recognition, and video processing. He received the Best Paper Award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2009. He is a program chair of the IEEE International Conference on Computer Vision (ICCV) 2009 and an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*TPAMI*) and the *International Journal of Computer Vision* (*IJCV*). He is a fellow of the IEEE.