Author name(s)

Book title

– Monograph –

October 25, 2010

Springer

Contents

Part I Part Title

Part I
Part Title

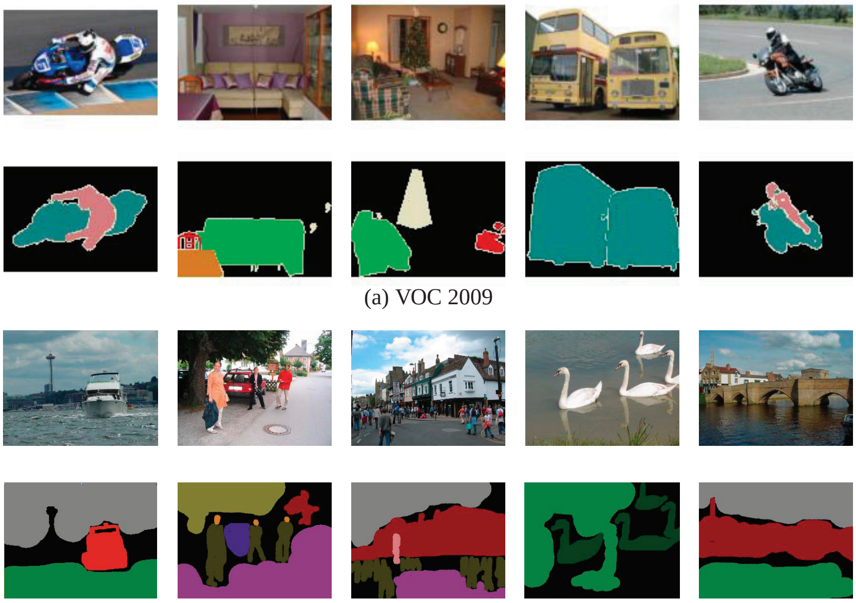
Use the template *part.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your part title page and, if desired, a short introductory text (maximum one page) on its verso page in the Springer layout.

Chapter 1
Semantic Object Segmentation

Abstract Semantic object segmentation is to label each pixel in an image or a video sequence to one of the object classes with semantic meanings. It has drawn a lot of research interest because of its wide applications to image and video search, editing and compression. It is a very challenging problem because a large number of object classes need to be distinguished and there is a large visual variability within each object class. In order to successfully segment objects, local appearance of objects, local consistency between labels of neighboring pixels, and long-range contextual information in an image need to integrated under a unified framework. Such integration can be achieved using conditional random fields. Conditional random fields are discriminative models. Alhtough they can learn the models of object classes more accurately and efficiently, they require training examples labeled at pixel-level and the labeling cost is expensive. The models of object classes can be learned with different levels of supervision. In some applications, such as web-based image and video search, a large number of object classes need to be modeled and therefore unsupervised learning or semi-supervised learning is preferred. Therefore some generative models, such as topic models, are used in object segmentation because of their capability to learn the object classes without supervision or with weak supervision of less labeling work. We will overview different technologies used in each step of the semantic object segmentation pipeline and discuss major challenges for each step. We will focus on conditional random fields and topic models, which are two types of frameworks widely used in semantic object segmentation. In video segmentation, we summarize and compare the frameworks of Markov random fields and conditional random fields, which are the representative models of the generative and discriminative approaches respectively.

1.1 Introduction

The task of semantic object segmentation is to label each pixel in an image or a video sequence to one of the object classes with semantic meanings (see exam-



(a) VOC 2009

(b) MSRC 21

Fig. 1.1 Examples of images (first row) and manually segmented objects (second row) from PASCAL VOC 2009 [1] (a) and MSRC 21 [2] (b). Different colors represent object categories.

ples in Figure 1.1). The object classes can be pre-defined or unsupervised learned from a collection of images or videos. It is different than unsupervised image and video segmentation, which is to group pixels into regions with homogeneous color or texture but without semantic meanings. It has important applications to image and video search, editing and compression. For example, semantic regions with their 2D spatial arrangement sketched by users can be used as query to retrieve image. Segmented objects can be deleted from images or copied between images. Different regions of images can be enhanced in different ways based on their semantic meanings.

Semantic object segmentation is a very challenging problem, because there are a very large number of object classes to be distinguished, some object classes are visually similar, and each object class may have very large visual variability. These object classes can be structured, such as cars and airplanes, or unstructured, such as grass fields and water. Due to variations of viewpoints, poses, illuminations and occlusions, objects of the same class have different appearance across images. In order to develop a successful semantic object segmentation algorithm, there are three important factors to be considered: local appearance, label consistency between neighboring pixels, and long-range contextual information [3]. In order to model the local appearance at a pixel, filter-banks and visual descriptors are applied to the neighborhood around the pixel and their responses are used as the input of a classifier to predict the object label. The filter-banks, visual descriptors and classifiers have to be carefully designed in order to achieve a good balance between high discriminative power and invariance to noise, clutters and the changes of viewpoints and illuminations. In order to obtain smooth segmentation results, the label consistency between neighboring pixels needs to be considered. In order for the segmentation to be consistent with the boundaries of objects, the algorithm should encourage two neighboring pixels to have the same object label if there is no strong edge between them. In addition to smoothness, the likelihood of two object classes being neighbors should also be considered for local consistency. For example, it is more likely for a cup to be on the top of desk than on a tree. Only considering the appearance of an image patch leads to ambiguities when deciding its class label. For example, a flat white patch could be from a wall, a car or an airplane. The long-range contextual information of the image may help to solve the ambiguities to some extent. For example, some object classes such as horses and grass are more likely to co-existing in the same images. If it is known that the image is an outdoor scene, it is more likely to observe sky, grass and cars than computers, desks and floors in that image. Local appearance, local consistency and long-range contextual can be incorporated in a Conditional Random Field (CRF) model [4], which has been popularly used in semantic object segmentation.

The approaches of semantic object segmentation can be supervised or unsupervised. The supervision at the training stage can be at three different levels,

- pixel-level: each pixel in an image is manually labeled as one of the object classes;
- mask-level: an object in an image is located by a bounding box and assigned to a object class;
- image-level: annotating object classes existing in an image without locating or segmenting objects.

Most discriminative object segmentation approaches including CRF need pixel-level or mask-level labeling for training. They can learn the models of object classes more accurately and efficiently. However, as the fast increase of images and videos in many applications such as web-based image and video search, there are a increasing number of object classes to be modeled. The workload of pixel-level and mask-level labeling is heavy and impractical for a very large number of object classes. In recent years, some generative models, such as topic models borrowed from language processing, have become popular in semantic object segmetation. They are able to learned the models of object classes from a collection of images and videos without supervision or supervised by data labeled at the image-level, whose labeling cost is much less. It is also possible to CRF and topic models to integrate the strengths of both types of approaches.

A typical pipeline of semantic object segmentation is shown in Figure 1.2. Filter-banks or visual descriptors are first applied to images to capture the local appearance objects. Their responses are typically quantized into textons or visual words according to codebooks learned in a supervised or unsupervised way. The histograms of textons or visual words are used as input to a classifier to predict labels of object

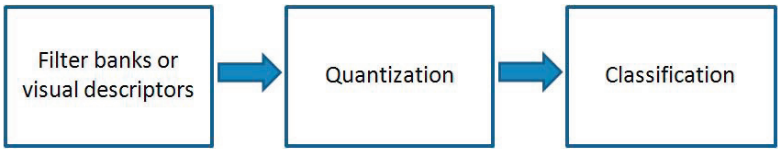
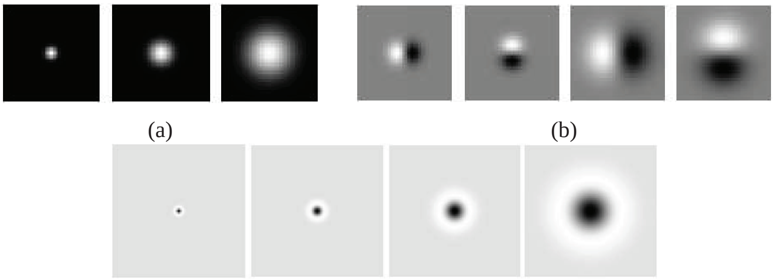


Fig. 1.2 Typical steps of semantic object segmentation. They are done over image pixels, patches or oversegmented superpixels.

classes. In order to well capture local consistency and long-range contextual information, CRF or generative models are used to incorporate with local classifiers. These steps can be on at image pixels, patches or oversegmented superpixels. Many different technologies have been developed to improve each of the three steps. We will review these technologies and discuss the major challenges for these steps. In recent years, some benchmark databases, such as PASCAL VOC 2007 [5], PASCAL VOC 2008 [6], PASCAL VOC 2009 [1], LabelMe [7], LHI [8] and MSRC 21 [2], were published to evaluate the performance of different semantic object segmentation approaches.

In video segmentation, Markov random fields (MRFs) and CRFs are two main frameworks. Statistically, video segmentation formulizes and maximizes a posterior probability of the labels given the observation data. In the case that there is no or only small number of labeled data, some heuristic or prior knowledge based distributions can be selected to describe the observation data. Based on the selected distributions and the prior of labels modeled in a MRF, the MRF approaches formulate the posterior via likelihoods and priors in Baye's rule. On the contrast, CRFs model the posterior directly to improve the predictive performance if there are large quantities of training data. In CRFs, the model of the observation data is obtained by learning from the training data using some classifiers. Compare to MRFs, CRFs relax the assumption of data independence, while large more expensive labeled data is necessary in CRFs.

This chapter is organized as follows. Section 1.2 introduces different types of filter-banks and visual descriptors to capture local appearance, and different techniques to quantize their responses into textons or visual words. Some popular classifiers on local appearance are reviewed in Section 1.3.1. Section 1.3.2 introduces CRF and different approaches of using CRF for semantic object segmentation. Section 1.4 first introduces two classical topic models, Probabilistic Latent Semantic Analysis [9] (pLSA) and Latent Dirichlet Allocation [10] (LDA), which were directly borrowed from language processing and applied to semantic object segmentation. Both pLSA and LDA ignored the spatial distribution of image patches. Spatial Latent Dirichlet Allocation [11], which is an extension of LDA and other topic models incorporating spatial structures of objects are introduced in Section 1.4.2 and Section 1.4.3. The approaches of object segmentations in videos are discussed in Section 1.5. Finally the summary is given in Section 1.6.



(a)

(b)

(c)

Fig. 1.3 A set of filter banks proposed by Winn [2]. (a) Three Gaussian kernels with $\sigma = 1,2,4$. They were applied to each CIE L,a,b channel. (b) Four derivatives of Gaussians divided into the $x-$ and $y-$ aligned sets, each with two different values of $\sigma = 2,4$. They were applied to L channel. (3) Four Laplacian of Gaussians with $\sigma = 1,2,4,8$. They were applied to L channel.

1.2 Local Visual Cues

1.2.1 Filter-banks and visual descriptors

Filter-banks and visual descriptors are used to capture the local appearance of objects. They are calculated from the neighbor of a pixel. On the one hand, they need to be discriminative enough to distinguish a large number of object classes, some of which are visually similar; on the other hand, they need to have invariance to noise, clutters and changes of illuminations and viewpoints. If they are computed at every pixel, computational efficiency is another issue to be considered. In this section we will review some popularly used filter-banks and visual descriptors.

Filter-banks. Filter-banks capture certain frequencies within a neighborhood. Winn et al. [2] proposed a set of filter-banks after testing different combinations of Gaussians, Laplacian of Gaussians (LoG), first and second order derivatives of Gaussians and Gabor kernels on semantic object segmentation. The proposed set of filter-banks included three Gaussians, four LoGs and four first order derivatives of Gaussians. The three Gaussian kernels with different standard deviation parameters $\sigma = 1,2,4$ were applied to each CIE L,a,b channel. The four LoGs(with $\sigma = 1,2,4,8$) and the four first order derivatives of Gaussians (with $\sigma = 1,2,4,8$) were applied to L channel only. The first order derivatives of Gaussians were in x and y directions. See the kernels of the proposed filter-banks in Figure 1.3. Some other filter-banks, such as rotation-invariant filters and maximum-response filters, were also proposed [12, 13, 14]. A comparison study can be found in [15].

SIFT. SIFT (Scale-Invariant Feature Transform) (see Figure 1.4) proposed by Lowe [16] is the most widely used local visual descriptors. It has reasonable invariance to changes in illumination, rotation, scaling and small changes in viewpoints. SIFT keypoints were detected by finding local extrema of Difference-of-Gaussian (DoG) filters at different scales. For each keypoint, its orientation and scale were se-

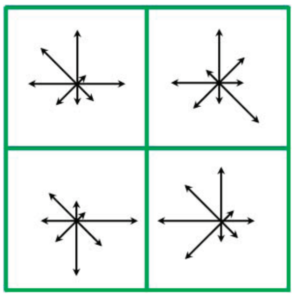


Fig. 1.4 SIFT descriptor [16] is computed by combining the normalized orientation histograms of gradients within subregions of the keypoint into a feature vector.

lected. A SIFT descriptor of a keypoint was obtained by first computing the gradient magnitudes and orientations of pixels in the neighborhood region of the keypoint, using the scale of the keypoint to select a proper Gaussian kernel to blur the image. It order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations were rotated relative to the keypoint orientation. The orientation histograms within the subregions around the keypoint were computed and combined into the SIFT feature vector. This vector was normalized to improve the invariance to changes of illumination. Gradient Location and Orientation Histogram (GLOH) [17] extended SIFT by allowing SIFT descriptor to be computed on a log-polar location grid.

HOG. Histogram of Oriented Gradients proposed by Dalal and Triggs [18] was similar to SIFT. It computed the histograms of gradient orientations in different subregions. Different from SIFT which was computed on detected sparse keypoints, HOG was sampled from a dense and uniform grid and was improved by local contrast normalization in overlapping spatial blocks. Integral Histogram of Oriented Gradients (IHO) [19] is an approxmiation of HOG and can be efficiently computed using integral images.

MSER. Instead of detecting keypoints, Maximally Stable Extremal Regions (MSER) proposed by [20] et al. detected regions which were darker or brighter than surroundings. It was affinely-invariant and robust to changes of illuminations. It was extended to colour in [21].

SURF. Bay et al. [22] proposed the SURF (Speeded Up Robust Features) descriptor, which could be efficiently computed using integral images. The neighborhood of a pixel was uniformly into $P \times Q$ spatial bins. The SURF descriptor was calculated by accumulating the sum of Haar wavelet responses at different spatial bins. Let d_x and d_y be the Haar wavelet responses in the horizontal and vertical directions. The descriptor has a four-dimensional vector $\left(\sum d_x, \sum \|d_x\|, \sum d_y, \sum \left\|d_y\right\|\right)$ for each spatial bin. The resulting $4 \times P \times Q$ dimensions SURF descriptor was L1-normalized.

Spin image and RIFT. Lazebnik et al. [23] proposed two rotation-invariant descriptors, spin image and RIFT (Rotation-Invariant Feature Transform). The spin image was a two-dimensional histogram of image intensities and their distance to the keypoint. To construct the RIFT descriptor, the circular normalized patch around the keypoint was divided into concentric rings of equal width and a gradient orientation histogram was computed within each ring.

Most descriptors described above were applied to intensity images. To increase illumination invariance and discriminative power, color descriptors were proposed. An evaluation of different color descriptors can be found in [24]. It was shown that the combination of different filter-banks of visual descriptors could improve the performance [25].

These filter bank responses and invariant descriptors can be computed at image patches densely sampled or at sparse interest points. Results in [26] showed that densely sampling improved the performance because it captured the most information, but its computation was expensive.

1.2.2 Textons and visual words

In semantic object segmentation, the responses of filter-banks or visual descriptors are usually further quantized to textons or visual words [1] according to a learned codebook. Since the histograms of textons or visual words will be used as the input of classifiers at later stages, the design of codebooks should consider both distinctiveness and repeatability. This means that it should try to assign image patches of different object classes to different codewords and to assign image patches of the same object class to the same codeword. The codebook should be compact in order to avoid overfitting of the classifiers at later stages. Because there are a huge number of image patches in data collections, memory and computation efficiency is another issue to be considered when learning the codebooks.

K-means is the most commonly used clustering methods to generate the codebooks. Some examples of visual words obtained by k-means are shown in Figure 1.5. Since the distribution of image patches in the filter-bank space or in the descriptor space is far from uniform, one of the disadvantages of k-means is that it clusters centres almost exclusively around the densest few regions in descriptor space and cannot over other informative regions well. Based on this consideration, Jurie et al. [27] proposed a new approach building codebooks using mean shift. Some patches in the dense regions were removed and the learned codebooks were more informative.

K-means has high computational cost. It also has the difficulty of balancing the distinctiveness and repeatability by choosing different sizes of codebooks. If the size of the codebook is too small, image patches of different object classes will fall into the same bin. At the other extreme, image patches around the same keypoint observed in different images will fall into different bins. To overcome these difficulties, Nister et al. [28] proposed the vocabulary tree constructed by hierarchical k-means. It allowed a larger and more discriminatory codebook to be used efficiently. Moosmann et al. [29] proposed Extremely Randomized Clustering Forests, which were

[1] Textons are quantized responses of filter-banks and visual words are quantized visual descriptors.

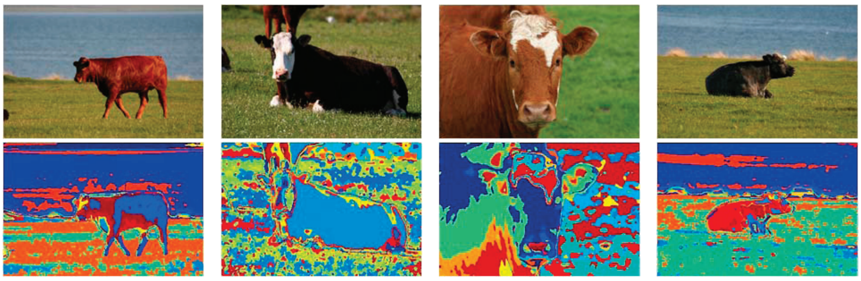


Fig. 1.5 Examples of visual words obtained by the filter-banks proposed in [2] and k-means. The first row are images and the second row are visual words. Colors represent different visual words.

ensembles of randomly created clustering trees, to learn the codebook. It provided more accurate results and was faster than k-means. Elkan [30] used the triangle inequality to dramatically accelerate k-means, while guaranteed always computing exactly the same result as the standard k-means.

K-means assumed hard assignment, i.e. exactly assigning a single visual word to one image feature. If an image feature is relevant to multiple textons or visual words, only the best is selected. If none of the codewords in the codebook well represent the image feature, the best one is still assigned to the image feature. These may cause problems during object segmentation. van Gemert et al. [31] created codebooks using kernel density estimation. It modeled the uncertainty between visual words and image features.

The above approaches are unsupervised. Some supervised approaches learned codebooks incorporating semantic information. These codebooks were more compact and discriminative. Winn et al. [2] learned an optimally compact visual codebook by pair-wise merging of visaul words given segmented images for training. Shotton et al. [32] proposed semantic texton forests, which were randomized decision forests [33] and were learned from image pixels. Perronnin et al. [34] learned different codebooks for different object classes by adapting a universal codebook, which described the content of all the classes of images, using class-specific data. Both the universal codebook and adapted class-codebooks were used for classification.

1.3 Object Segmentation Using Discriminative Approaches

1.3.1 Classifiers on local appearance

The obtained histograms of textons or visual words within local regions capture the features of local appearance and are usually used as the input of classifiers to predict object labels. Support Vector Machines (SVM) and Boosting are widely used to model the appearance of object classes. Marsalek and Schmid [35] estimated the shape mask of an object and its object category using non-linear SVM and with χ^2 distance. The appearance of the object within the shape mask was represented by a histogram of visual words. Shotton et al. [32] used the texton histograms and region priors, which were calculated from their proposed semantic texton forests, of image regions as input of a one-vs-others SVM classifier to assign image regions into different object classes. Gould et al. [36] used the boosting classifier to predict the label of each pixel. Tahir et al. [25] used Spectral Regression Kernel Discriminant Analysis(SRKDA) [37] and achieved better results than SVM on PASCAL VOC 2008 [6]. It was also much more efficient than Kernel Discriminant Analysis(KDA). Aldavert et al. [38] proposed an integral linear classifier, which used integral images to efficiently calculate the outputs of linear classifiers based on histograms of visual words at the pixel level.

1.3.2 Conditional Random Fields

Although classifiers such as SVM and Boosting can predict the object label of a pixel based on the appearance within its neighborhood, they cannot capture local consistency other contextual features, such as "sky" appears above buildings but not the other way around. Local appearance, local consistency and contextual features can be well incorporated under a Conditional Random Fields (CRF) framework.

1.3.2.1 Multiscale conditional random fields

He et al. [39] were the first to use CRF for semantic object segmentation. Their proposed CRF framework is described as following. Suppose $\mathbf{X} = \{x_i\}$ are image patches and $\mathbf{Z} = \{z_i\}$ are their object class labels. In [39], the conditional distribution over $\mathbf{Z}$ given input $\mathbf{X}$ was defined by multiplicatively combining component conditional distributions.

$$P(\mathbf{Z}|\mathbf{X}) \propto P_C(\mathbf{Z}|\mathbf{X})P_R(\mathbf{Z}|\mathbf{X})P_G(\mathbf{Z}|\mathbf{X}). \tag{1.1}$$

P_C, P_R and P_G capture statistical structures at three different spatial scales: local classifier, regional features and global features (see Figure 1.6).

The local classifier P_C produces a distribution over the label z_i given its image patch x_i as input,

$$P_C(\mathbf{Z}|\mathbf{X},\lambda) = \prod_i P_C(z_i|x_i,\lambda), \tag{1.2}$$

where λ is the parameter of the local classifier. A 3-layer multilayer perceptron(MLP) was used in [39].

The regional features P_R represent local geometric relationships between objects. They avoid impossible combinations of neighboring objects such as "ground is above sky" and also encourage the segmentation results to be spatially smooth. A

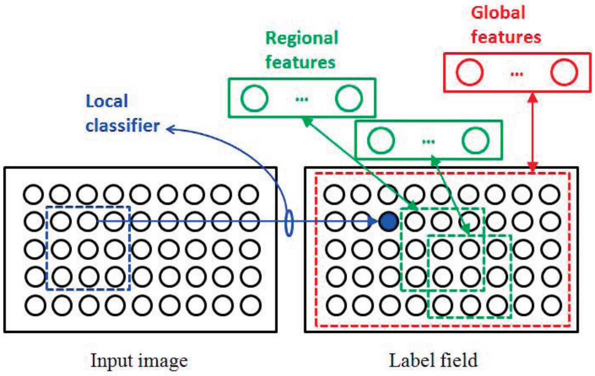


Fig. 1.6 A graphical representation of CRF. Reproduced from [39].

collection of regional features are learned from the training data. Let r be the index of regions and a be the index of the different regional features within each region, and j be the index of image patches within in region r. P_R is defined as

$$P_R(\mathbf{Z},\mathbf{f}) \propto exp\{\sum_{r,a} f_{r,a}\mathbf{w}_a^T\mathbf{z}_r\}. \tag{1.3}$$

$\mathbf{f} = \{f_{r,a}\}$ are binary hidden regional variables. $f_{r,a} = 0,1$ indicating the feature a in region r exists or not. $\mathbf{w}_a = [w_{a,1},\ldots,w_{a,J},\alpha_a]$ are parameters and α_a is a bias term. $w_{a,j}$ connects $f_{r,a}$ with $z_{r,j}$ and specifies preferences for the possible label value of $z_{r,j}$. $\mathbf{z}_r = [z_{r,1},\ldots,z_{r,J},1]$. P_R is high of $\mathbf{z}_r$ matches $\mathbf{w}_a$ and $f_{r,a} = 1$ or $\mathbf{z}_r$ does not match $\mathbf{w}_a$ and $f_{r,a} = 0$.

The global feature P_G is defined over the whole image,

$$P_G(\mathbf{Z},\mathbf{g}) \propto exp\{\sum_b g_b\mathbf{u}_b^T\mathbf{Z}\}. \tag{1.4}$$

b is the index of the global label patterns, which are encoded in the parameters $\{\mathbf{u}_b\}$. $\mathbf{g} = \{g_b\}$ are the binary hidden global variables.

Both hidden variables **f** and **g** can be marginalized, leading to

$$P_R(\mathbf{Z}) \propto_{r,a} \left[1+exp(\mathbf{w}_a^T\mathbf{z}_r)\right], \tag{1.5}$$

$$P_G(\mathbf{Z}) \propto_b \left[1+exp(\mathbf{u}_b^T\mathbf{Z})\right]. \tag{1.6}$$

Thus Eq(1.2) has a closed form,

$$P(\mathbf{Z}|\mathbf{X};\theta) \propto \prod_i P_C(z_i|x_i,\lambda) \times \prod_{r,a}\left[1+exp(\mathbf{w}_a^T\mathbf{z}_r)\right] \times \prod_b\left[1+exp(\mathbf{u}_b^T\mathbf{Z})\right]. \tag{1.7}$$

$\theta = \{\lambda, \{\mathbf{w}_a\}, \{\mathbf{u}_b\}\}$ are parameters. They are learned from a training by maximizing the conditional likelihood in [39]. Once the parameters are learned, the object class labels are inferred by maximizing posterior marginals.

1.3.2.2 TextonBoost

Under the CRF framework, Shotton et al. [40] proposed TextonBoost to learn a discriminative model of object classes incorporating texture, layout and context information. Their CRF includes four types of potentials: texture-layout, color, location and edge.

$$\log P(Z|X,\theta) = \sum_i \overbrace{\psi_i(z_i, \mathbf{X}; \theta_\psi)}^{texture-layout} + \overbrace{\pi(c_i, x_i; \theta_\pi)}^{color} + \overbrace{\ell(z_i, i; \theta_\ell)}^{location} + \sum_{(i,j)\in\varepsilon} \overbrace{\xi(z_i, z_j, g_{ij}(\mathbf{X}); \theta_\xi)}^{edge} - \log C(\theta, \mathbf{X}) \quad (1.8)$$

where i and j are indices of pixels, $(i, j) \in \varepsilon$ are two neighboring pixels, $\theta = \{\theta_\psi, \theta_\pi, \theta_\ell, \theta_\xi\}$ are parameters, and $C(\theta, \mathbf{X})$ is a normalization term.

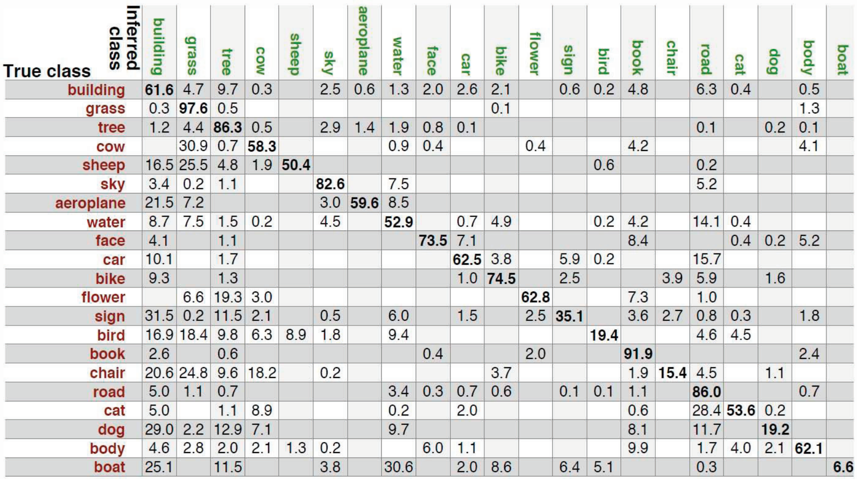
The texture-layout potentials are provided by a boosting classifier combining a set of discriminative features called texture-layout filters. The neighborhood of pixel i is partitioned into regions by a predefined spatial kernel. Each texture-layout $v_{[r,t]}(i)$ is the number of pixels with texton t in region r. Therefore, texture-layout filters are histograms of textons over defined spatial kernels. They capture texture, spatial layout and textural context. Discriminative texture-layout filters are selected as weak classifiers and combined into a powerful classifier by Joint Boost [41]. Joint Boost allows to share weak classifiers among different object classes and the learn classifier has better generalization.

The color potentials model the color distribution of each object class using Gaussian mixture models in CIELab color space.

The location potentials model the dependence between the locations of pixels and object classes. For example, trees and sky tend to appear in the top regions of images while roads tend to appear in the bottom regions of images.

In the edge potentials, g_{ij} measures the edge features between neighbor pixels. A penalty is added if two neighboring pixels have different object class labels unless there is a strong edge between them.

TextonBoost was evaluated on 21 object classes from the MSRC database and achieved 72.2% overall accuracy [40]. The confusion matrix is shown in Figure 1.7. The experimental evaluation showed that although the texture-layout potentials had the most significant contribution to semantic object segmentation, CRF significantly improved the accuracy of results.



Inferred class True class	building	grass	tree	cow	sheep	sky	aeroplane	water	face	car	bike	flower	sign	bird	book	chair	road	cat	dog	body	boat
building	**61.6**	4.7	9.7	0.3		2.5	0.6	1.3	2.0	2.6	2.1		0.6	0.2	4.8		6.3	0.4		0.5	
grass	0.3	**97.6**	0.5								0.1									1.3	
tree	1.2	4.4	**86.3**	0.5		2.9	1.4	1.9	0.8	0.1							0.1		0.2	0.1	
cow		30.9	0.7	**58.3**				0.9	0.4			0.4			4.2					4.1	
sheep	16.5	25.5	4.8	1.9	**50.4**									0.6			0.2				
sky	3.4	0.2	1.1			**82.6**		7.5									5.2				
aeroplane	21.5	7.2				3.0	**59.6**	8.5													
water	8.7	7.5	1.5	0.2		4.5		**52.9**		0.7	4.9			0.2	4.2		14.1	0.4			
face	4.1		1.1						**73.5**	7.1					8.4			0.4	0.2	5.2	
car	10.1		1.7							**62.5**	3.8		5.9	0.2			15.7				
bike	9.3		1.3							1.0	**74.5**		2.5			3.9	5.9		1.6		
flower		6.6	19.3	3.0								**62.8**			7.3		1.0				
sign	31.5	0.2	11.5	2.1		0.5		6.0		1.5		2.5	**35.1**		3.6	2.7	0.8	0.3		1.8	
bird	16.9	18.4	9.8	6.3	8.9	1.8		9.4						**19.4**			4.6	4.5			
book	2.6		0.6						0.4			2.0			**91.9**					2.4	
chair	20.6	24.8	9.6	18.2		0.2					3.7				1.9	**15.4**	4.5		1.1		
road	5.0	1.1	0.7					3.4	0.3	0.7	0.6		0.1	0.1	1.1		**86.0**			0.7	
cat	5.0		1.1	8.9				0.2		2.0					0.6		28.4	**53.6**	0.2		
dog	29.0	2.2	12.9	7.1				9.7							8.1		11.7		**19.2**		
body	4.6	2.8	2.0	2.1	1.3	0.2			6.0	1.1					9.9		1.7	4.0	2.1	**62.1**	
boat	25.1		11.5			3.8		30.6		2.0	8.6		6.4	5.1			0.3				**6.6**

Fig. 1.7 Confusion matrix of object segmentation by TextonBoost [40] on the MSRC 21 database. The figure is reproduced from [40].

1.3.2.3 Other approaches based on conditional random fields

Other semantic object segmentation approaches based CRF were proposed. Fulkerson et al. [42] treated superpixels [43], which were small regions obtained from a conservative oversegmentation, as basic units of segmentation. They assumed that superpixels allowed to measure histograms of visual words on a natural adaptive domain rather than on a fixed patch window. Moreover, superpixels tend to preserve boundaries and created more accurate segmentation. A one-vs-others SVM classifier with a RBF-χ^2 kernel was constructed on the histograms of visual words found in each superpixel. This local classifier was used in a CRF operating on the superpixel graph. CRF was used to add spatial regularization by requiring that if two neighboring superpixels share a long boundary and were similar in appearance, they tended to have the same class label. It discouraged small isolated regions and reduced misclassifications that occurred near the edges of objects. He et al. [44] also first oversegmented images into superpixels. Superpixels were labeled under a mixture of CRF. Images in a database were grouped into several contexts and each context was modeled by a separate CRF.

Torralba et al. [45] proposed Boosted Random Fields for object detection and segmentation. Boosting was used to learn the graph structure and local evidence of a conditional random field. The graph structure of CRF was learned using boosting to select from a dictionary connectivity templates which were derived from labeled segmentations. It exploited the contextual correlations between object classes. Rabinovich et al. [46] explicitly defined the interactions between object classes as semantic context and incorporated it into CRF. The semantic context was modeled as the co-occurrence of object labels and was learned both from the training data and Google Sets [2].

Quattoni et al. [47] used CRF for part-based object recognition and detection. CRF was used to model the spatial arranges of object parts. Ma and Grimson [48] proposed a coupled CRF to decompose the images into contour and texture and to model their interaction. The decomposed low-level cues were adaptively combined for object recognition and different discriminative cues for different object classes were fully leveraged. Reynolds and Murphy [49] proposed a tree-structured CRF for object segmentation.

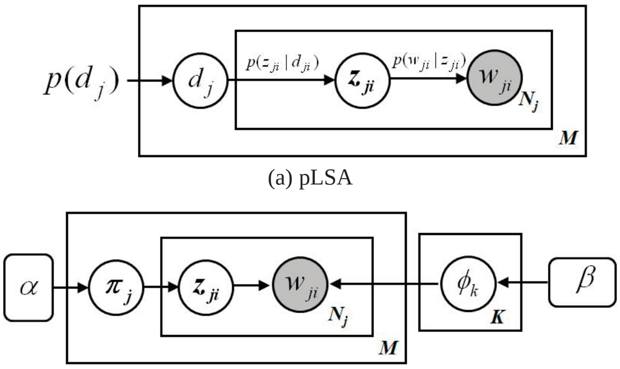
1.4 Object Segmentation Using Topic Models

The discriminative approaches described above required training data to be labeled at pixel-level. If there are a large number of object classes to be modeled, the labeling work is very expensive. Some researchers started to explore approaches of learning the models of object classes from a collection of images or videos without supervision or with weak supervision (such as using training data labeled at image-level). Inspired by the success of topic models, such as Probabilistic Latent Semantic Analysis (pLSA) [9] and Latent Dirichlet Allocation (LDA) [10], in the applications of language processing, they have been aslo applied to semantic object segmentation in recent years. Under pLSA or LDA, words, such as “professor” and “university”, often co-existing in the same documents are clustered into the same topic, such as “education”. The models of topics are automatically without supervision. The word-document analysis has been applied to object segmentation through mapping the concepts of “words” and “documents” to the image and video domains. For example, if images are treated as documents and visual words (or textons) are treated as words, with the assumption that visaul words of the same object classes often co-exist in the same images, the models of object classes can be learned as the models of topics. Object classes are treated as topics. Since an image may include objects of several classes, it is model as a mixture of topics. An advantage of such an approach is that manually segmenting objects at the pixel level is not required for training. Some proposed approaches [50, 51, 11] were totally unsupervised. Some required labeling at the image level [52, 53]. Some semantic object segmentation approaches based on topics models will be reviewed in this section.

1.4.1 pLSA and LDA

Sivic et al. [50] discovered the object classes from a set of unlabeled images and segmented images into different object classes using pLSA and LDA. They modeled

[2] http://labs.google.com/sets



(a) pLSA

(b) LDA

Fig. 1.8 Graphical models of pLSA and LDA.

an image as a bag of visual words and ignored any spatial relationships among visual words. Suppose there are M images in the data set. Each image j has N_j visual words. Each visual word w_{ji} is assigned one of the K object classes according to its label z_{ji}. Under pLSA the joint probability $P(\{w_{ji}\},\{d_j\},\{z_{ji}\})$ has the form of the graphical model shown in Figure 1.8(a). The conditional probability $P(w_{ji}|d_j)$ marginalizing over topics z_{ji} is given by

$$P(w_{ji}|d_j) = \sum_{k=1}^{K} P(z_{ji} = k|d_j)P(w_{ji}|z_{ji} = k). \tag{1.9}$$

$P(z_{ji} = k|d_j)$ is the probability of object class k occurring in image d_j. $P(w_{ji}|z_{ji} = k)$ is the probability of visual word w_{ji} occurring in object class k and is the model of object class k. Fitting the pLSA model involves determining $P(w_{ji}|z_{ji})$ and $P(z_{ji} = k|d_j)$ by maximizing the following objective function using the Expectation Maximization (EM) algorithm:

$$L = \prod_{j=1}^{M}\prod_{i=1}^{N_j} P(w_{ji}|d_j) \tag{1.10}$$

Images are segmented into objects with semantic meanings based on the labels z_{ji} of visual words.

pLSA is a generative model only for training images but not for new images. This shortcoming has been addressed by LDA, whose graphical model is shown in Figure 1.8(b). Under LDA, $\{\phi_k\}$ are models of object classes and are discrete distributions over the codebook of visual words. They are generated from a Dirichlet prior $Dir(\phi_k;\beta)$ given by β. Each image j has a multinomial distribution π_j over K object classes and it is generated from a Dirichlet prior $Dir(\pi_j;\alpha)$. Each patch i on image j is assigned to one of the K object classes and its label z_{ji} is sampled from a discrete distribution $Discrete(z_{ji};\pi_j)$ given by π_j. The observed visual word

w_{ji} is sampled from the model of its object class: $Discrete(w_{ji}|\phi_{z_{ji}})$. α and β are hyperparameters. ϕ_k, π_j and z_{ji} are hidden variables to be inferred. The inference can by implemented by variational methods [10] or collapsed Gibbs sampling [54]. Under LDA, if two visual words often co-occur in the same images, one of the object class models will have large distributions on both of them. pLSA and LDA perform similarly on image classification and object segmentation and their results were promising especially when each image only contained one object. As reported by [50], on a data set consisting of $4,090$ images of five categories from the Caltech 101 database [55], the image classification accuracy achieved by pLSA was 92.5% (see Table 1.1) and its object segmentation accuracy was 49%. Both pLSA and LDA requires the number of object classes to be known in advance. As an extension, Hierarchical Dirichlet Process (HPD) proposed by Teh et al. [54] could automatically learn the number of object classes from data using Dirichlet Processes [56] as priors.

True class →	Faces	Motorbikes	Airplanes	Cars	Background
Class 1 - Faces	94.02	0.00	0.38	0.00	1.00
Class 2 - Motorbikes	0.00	83.62	0.12	0.00	1.25
Class 3 - Airplanes	0.00	0.50	95.25	0.52	0.50
Class 4 - Cars	0.46	0.88	0.38	98.1	3.75
Class 5 - Background I	1.84	0.38	0.88	0.26	41.75
Class 6 - Background II	3.68	12.88	0.88	0.00	23.00
Class 7 - Background III	0.00	1.75	2.12	1.13	28.75

Table 1.1 Confusion table of using pLSA for image classification on a data set of five object categories from the Caltech 101 database [55]. Class number is equal to 7 in pLSA. Three classes correspond to the background. The result was reported in [50].

1.4.2 SLDA

A shortcoming of using pLSA and LDA to segment objects is to treat an image as a document of visual words ignoring the spatial structure among visual words. The assumption that if two types of patches are from the same object class, they often appear in the same images is not strong enough. As an example shown in Figure 1.9, although the sky is far from the vehicles, if they often exist in the same images in the data set, they would be clustered into the same topic (object class) by pLSA or LDA. Since most parts of this image are sky and building, an image patch on a vehicle is likely to be labeled as building or sky as well. Such problems can be solved if the document of an image patch, such as the yellow patch in Figure 1.9, only includes patches falling within its neighborhood, marked by the red dashed window in Figure 1.9 instead of the whole image.

With the assumption that if two types of image patches are from the same object class, they are not only often in the same images but also close in space, a Spatial Latent Dirichlet Allocation (SLDA) was proposed in [11]. Under SLDA, the

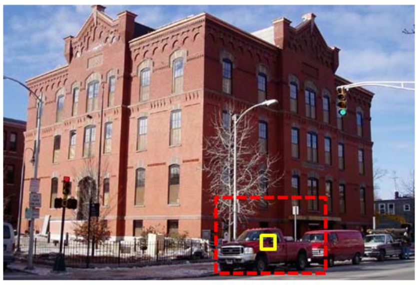


Fig. 1.9 There will be some problems (see text) if the whole image is treated as one document when using LDA to discover classes of objects.

word-document assignment becomes a hidden random variable. There is a generative procedure to assign words to documents. When visual words are close in space or time, they have a high probability to be grouped into the same document. The graphical model SLDA is shown in Figure 1.10. The N visual words in an image set are assigned to M documents. d_j is a hidden variable indicating the document assignment of visual word i. Each document j is associated with a hyperparameter $c_j^d = (g_j^d, x_j^d, y_j^d)$, where $(g_j^d$ is the index of the image where the document is placed and (x_j^d, y_j^d) is the location of the document. Besides the word value w_{ji}, the location (x_i, y_i) and image index g_i of a word i are observed and stored in variable $c_i = (g_i, x_i, y_i)$. The generative procedure is as following.

1. For a topic k, a multinomial parameter ϕ_k is sampled from Dirichlet prior $\phi_k \sim Dir(\beta)$.
2. For a document j, a multinomial parameter π_j over the K topics is sampled from Dirichlet prior $\pi_j \sim Dir(\alpha)$.
3. For a word (image patch) i, a random variable d_i is sampled from prior $p(d_i|\eta)$ indicating to which document word i is assigned. We choose $p(d_i|\eta)$ as a uniform prior.
4. The image index and location of word i is sampled from distribution $p(c_i|c_{d_i}^d, \sigma)$. We may choose this as a Gaussian kernel.

$$p((g_i, x_i, y_i) \,|\, \left(g_{d_i}^d, x_{d_i}^d, y_{d_i}^d\right), \sigma) \propto \delta_{g_{d_i}^d}(g_i)\, exp\left\{ -\frac{\left(x_{d_i}^d - x_i\right)^2 + \left(y_{d_i}^d - y_i\right)^2}{\sigma^2} \right\}$$

   $p(c_i|c_{d_i}^d, \sigma) = 0$ if the word and the document are not in the same image.
5. The topic label z_i of word i is sampled from the discrete distribution of document d_i, $z_i \sim Discrete(\pi_{d_i})$.
6. The value w_i of word i is sampled from the discrete distribution of topic z_i, $w_i \sim Discrete(\phi_{z_i})$.

In [11] both LDA and SLDA were evaluated on the MSRC data set [2] with 240 images for object segmentation. The detection rate and false alarm rate of four

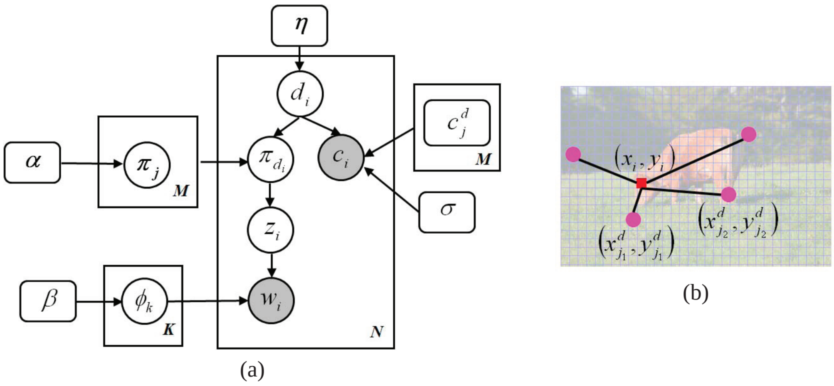


Fig. 1.10 (a) Graphical model of SLDA. (b) Add spatial information when designing documents. Each document is associated with a point (marked in magenta color). These points are densely placed over the image. If an image patch is close to a document, it has a high probability to be assigned to that document.

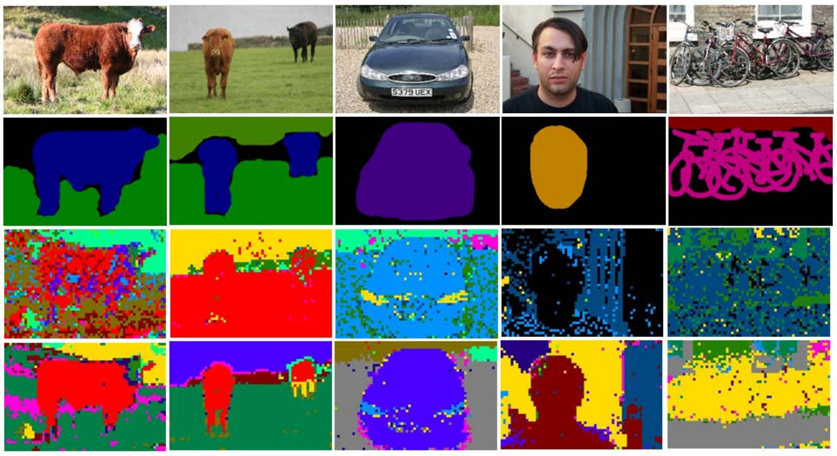


Fig. 1.11 Examples of object segmentation results by LDA and SLDA. The images are from the MSRC data set [2]. The first row shows example images. The second row uses manual segmentation and labeling as ground truth. The third row is the LDA result and the fourth row is the SLDA result. Under the same labeling approach, image patches marked in the same color are in one object cluster, but the meaning of colors changes across different labeling methods. The results are from [11].

classes (cows, cars, faces and bicycles) are shown in Table 1.2. Some examples are shown in Figure 1.11. The segmentation results of LDA were noisy since spatial information was not considered. The patches in the same image were likely to have the same labels. SLDA achieved better results.

In [11] SLDA was also used to segment objects from a video sequence. All the frames were treated as a collection of images and their temporal order was ignored.

	cows		cars		faces		bicycles	
	Det rate	FA rate	Det rate	FA rate	Det rate	FA rate	Det rate	FA rate
LDA	0.3755	0.5576	0.5552	0.3963	0.7172	0.5862	0.5563	0.5285
SLDA	0.5662	0.0334	0.6838	0.2437	0.6973	0.3714	0.5661	0.4217

Table 1.2 Detection(Det) rate and False Alarm (FA) rate of LDA and SLDA on MSRC [2]. The results are from [11].

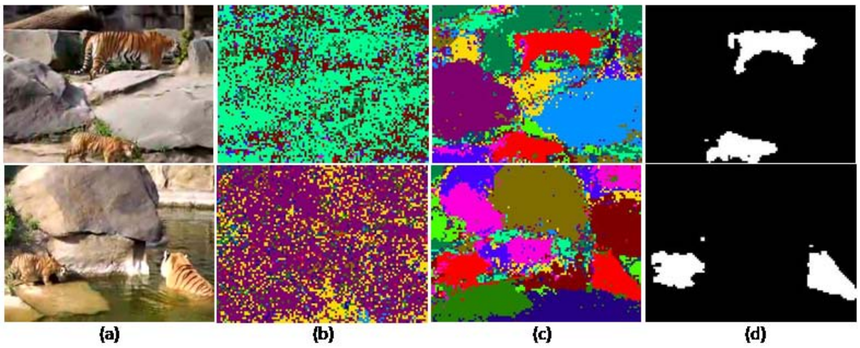


Fig. 1.12 Object segmentation from a video sequence. The first column shows two fames in the video sequence. In the second column, the patches in the two frames as are labeled as different object classes using LDA. The third column plots the object class labels using SLDA. The red color indicates the class of tigers. In the fourth column, tigers are segmented out by choosing all the patches of the class marked by red color. The results are from [11].

Figure 1.12 shows results on two sampled frames. LDA could not segment out any objects. SLDA clustered image patches into tigers, rock, water, and grass.

1.4.3 Other topic models of including spatial information

Some other topic models were also proposed to include spatial information. Russell et al. [51] first obtained multiple segmentations of each image at different scales using normalized cut [57] and then treated each segment instead of an image as a document. These segments captured the spatial relationships among visual words. Some good segments are sifted from bad ones for each discovered object class.

Verbeek et al. [52] proposed two aspect-based spatial field models by combining pLSA/LDA with Markov Random Fields (MRF). One is based on averaging over forests of minimal spanning trees linking neighboring image regions. A tree-structure prior is imposed to the object class labels $Z_j = \{z_{ji}\}$ of image patches in image j,

$$P(Z_j) \propto exp(\sum_i \psi(z_{ji}, z_{j\chi(i)}) + log\theta_j) \qquad (1.11)$$

where $\chi(i)$ is the unique parent of patch i in the tree, and $\psi(z_{ji}, z_{j\chi(i)})$ is a pair-wise potential,

$$\psi(z_{ji}, z_{j\chi(i)}) = \rho[z_{ji} = z_{j\chi(i)}]. \tag{1.12}$$

The other model applies an efficient chain-based Expectation Propagation method for regular 8-neighbor Markov Random Fields. The prior over Z_j is given by

$$P(Z_j) \propto exp(\sum_{i \sim i'} \psi(z_{ji}, z_{ji'}) + log\theta_j), \tag{1.13}$$

where $i \sim i'$ enumerates spatial neighbor patches i, i' in image j. MRF captures the local spatial dependence of image patches. These two models were trained using either patch-level labels or image-level labels. Tested on 240 images of nine object categories from the MSRC data set, when trained using patch-level labels, they achieved object segmentation accuracy of 80.2% and when trained using image-level labels, the accuracy of 78.1% was achieved. The accuracies of pLSA were 78.5% and 74.0% respectively under these two settings. The similar idea was also explored in [58] and a Dirichlet process mixture was introduced to automatically learn the number of object classes from data. In this framework was extended to Conditional Random Field (CRF) [4] to integrated both local and global features in the images [53, 59].

Sudderth et al. [60] proposed a Transformed Dirichlet Process (TDP) model to jointly solve the problem of scene classification and object segmentation. This approach coupled topic models with spatial transformations and consistently accounted for geometric constraints. The spatial relationships of different parts of objects were explicitly modeled under a hierarchical Bayesian model. Cao et al. [61] proposed a Spatially Coherent Latent Topic Model (Spatial-LTM) to simultaneously classifying scene categories and segmenting objects. It over segmented images into regions of coherent latent topic model and coherent latent topic model were considered as visual words. It enforced the spatial coherency of the model by requiring that only one single latent-topic was assigned to the image patches within each region.

1.5 Object Segmentation in Videos

A video is composed of a sequence of images. Different from still image segmentation, video segmentation should take account the temporal information. Many statistical models have been proposed for video segmentation, either generative or discriminative. In the discriminative model, a large number of expensive labeled data is required to train an excellent classifier. On the contrary, the generative model can handle the incomplete data problem and address the large number of unlabeled data via small number of expensive labeled data. Therefore, the generative model is popular for video segmentation. On the other hand, the discriminative model relax the conditional independence assumption and has better predictive performance than the generative model. This attract many attentions to the discriminative model in video segmentation. MRFs [62], [63] and CRFs [64], [65], [66], [67] are representative generative and discriminative models in video segmentation, respectively.

Let $\mathbf{X} = \{x_i\}_{i \in S}$ and $\mathbf{Z} = \{z_i\}_{i \in S}$ be the observation and labels of a video, where $S = \{s_i\}$ is the set of units (they can be pixels, patches, or semantic regions) in the video. Then video segmentation is to maximize the posterior $p(\mathbf{Z}|\mathbf{X})$.

1.5.1 MRF Model

In the MRF model, the posterior is expressed proportioned to the joint probability using the Baye's rule as:

$$p(\mathbf{Z}|\mathbf{X}) \propto p(\mathbf{Z}|\mathbf{X}) = p(\mathbf{X}|\mathbf{Z})p(\mathbf{Z}), \tag{1.14}$$

where the prior $p(\mathbf{Z})$ is modeled as a MRF.

In the MRF model, the strong assumption of conditional independency of the observed data is enforced. Therefore, the likelihood $p(\mathbf{X}|\mathbf{Z})$ is assumed to have a factorized form, i.e.,

$$p(\mathbf{X}|\mathbf{Z}) = \prod_{s_i \in S} p(x_i|z_i). \tag{1.15}$$

Here $p(x_i|z_i)$ indicates the probability that the unit s_i has the label z_i based on the observation x_i at s_i. Here x_i can be features incorporating the color, texture, and motion information. To adapt to changes of environment, some features robust to illumination changes are utilized, like gradient direction, shadow models, and color co-occurrence.

To model the distribution of $p(x_i|z_i)$, several ways have been proposed. The most traditional approach is model the distribution in terms of the Gaussian Mixture Models (GMMs) and the Expectation Maximization (EM) algorithm is used to estimate the model parameters. The GMM model has several shortcomings: it is sensitive to the initialization, the EM algorithm takes long time to converge, and a suitable number of Gaussian components has to be set. To address these problem, a non-parametric way, smoothed histograms in the YUV color space [64], has been proposed. It learns the histograms from some labeled region and stored in 3D look-up tables with smoothing. Then the value of $p(x_i|z_i)$ is searched from the histogram tables.

In the MRF model, $p(\mathbf{Z})$ is used to enforce the Markov properties of the labels. In the Bayesian view, the prior $p(\mathbf{Z})$ does not depend on the observed data $\mathbf{X}$. It is assumed to be an Potts model, i.e.,

$$p(\mathbf{Z}) = \exp\left(\sum_{s_i \in S} \sum_{s_j \in \mathcal{N}_i} \lambda T(z_i \neq z_j)\right), \tag{1.16}$$

where $\mathcal{N}_i$ is the neighborhood system of s_i, λ is a negative constant, and $T(\cdot) = 1$ if its argument is true and $T(\cdot) = 0$ if false. In video segmentation, the neighborhood system includes two parts, the spatial and temporal neighborhoods. The prior in the spatial neighborhood system incorporates the spatial smoothness constraint, which can reduce the effect of noise. The prior in the temporal neighborhood system is used to incorporate the inter-frame information. In the case of binary class problem (e.g., in foreground/background segmentation, $z_i \in \{1, -1\}$), the prior $p(\mathbf{Z})$ can be transformed as an isotropic Ising model, i.e.,

$$p(\mathbf{Z}) = \exp\left(\sum_{s_i \in S} \sum_{s_j \in \mathscr{N}_i} \lambda z_i z_j\right). \tag{1.17}$$

As noted above, the prior $p(\mathbf{X})$ does not depend on the observed data. But in the applications of video segmentation, observed data-dependent prior is necessary. In the part of spatial neighborhood system, the contrast information is incorporated by modulating the prior according to the intensity gradients. In the temporal part, the intensity difference is used to control the probability of s_i and s_j having the same label. Therefore, in video segmentation, the prior is expressed as

$$p(\mathbf{Z}) = \exp\left(\sum_{s_i \in S} \sum_{s_j \in \mathscr{N}_i} \lambda T(z_i \neq z_j) \cdot \exp\left(-\Delta_{i,j}^2/\sigma\right)\right), \tag{1.18}$$

where $\Delta_{i,j}$ is the intensity difference between s_i and s_j and σ is a positive constant. From the equation we can see that: if s_i and s_j have larger intensity difference, then they have a higher probability of being different labels.

Combing (1.14), (1.15), and (1.18), the posterior in MRF model is expressed as

$$p(\mathbf{X}|\mathbf{Z}) = \frac{1}{C}\exp\left(\sum_{s_i \in S} \log\left(p(x_i|z_i)\right) + \sum_{s_i \in S} \sum_{s_j \in \mathscr{N}_i} \lambda_m T(z_i \neq z_j)\right), \tag{1.19}$$

where C is the partition function and $\lambda_m = \lambda \exp(-\Delta_{i,j}^2/\sigma)$.

1.5.2 CRF Model

Compared with MRFs, the CRF model formulates the posterior $p(\mathbf{Z}|\mathbf{X})$ directly instead of formulating the joint probability $p(\mathbf{Z},\mathbf{X})$ via the likelihood $p(\mathbf{X}|\mathbf{Z})$ and $p(\mathbf{Z})$ by the Baye's rule. Generally, the posterior in MRFs is written as,

$$p(\mathbf{Z}|\mathbf{X}) = \frac{1}{C}\exp\left(\sum_{s_i \in S} u_i(z_i, \mathbf{X}) + \sum_{s_i \in S} \sum_{s_j \in \mathscr{N}_i} v_{ij}(z_i, z_j, \mathbf{X})\right), \tag{1.20}$$

where C is the partition function, $-u_i$ and $-v_{ij}$ are the unary and pairwise potential, respectively.

Comparing (1.20) with (1.19), the definitions of unary potential and the pairwise potential are different between MRFs and CRFs. In CRFs, the unary potential is a function in the term of the whole observed data **X**, while in MRFs the unary potential for s_i is a function in term of observed data at s_i due to the conditionally independent assumption. Theoretically, in MRFs the pairwise potential is a function of only labels (actually a function of labels and the intensity difference in the applications of video segmentation) while it is a function of labels and the whole observed data **X** in CRFs.

Since the potentials are in term of the whole observed data in CRFs, they are designed by using some arbitrary local discriminative classifiers. In discriminative classifiers, it is important to select a good feature space. Compared with MRFs, the CRF model selects more discriminative features besides colors, constant, and other features used in MRFs. For example in [65], texture, location, and histogram of oriented gradient (HOG) features are used for scene labeling. In [66], motion-shape cues are used for bilayer video segmentation. The features of "motons" (related to textons) are used for modeling the motion information in videos. A shape-filter modeling long-range correlations is selected to describe the shape features. Acturally, any fusion of discriminative features used in images can be selected in video segmentation. The difference between the video and image applications is good discriminative features describing the motion information may be used to improve the video segmentation results.

The second important thing in the discriminative model is classifier selection. In common, the classification algorithms build strong classifiers from a combination of weak classifiers. The difference between these algorithms is the way that the weak classifiers combine. In [66], the authors construct a tree cube taxonomy for helping to select classification algorithms. Fig. 1.13 is the tree cube taxonomy of classifiers. The origin is the weak learner and the axes H, A, and B are three basic ways of combining weak learners: hierarchically (H), by averaging (A), and via boosting (B). Different strong classifiers, i.e., different combinations of weak classifiers, correspond to different paths along the edges of the cube in Fig. 1.13.

1.5.3 MRFs Vs. CRFs

This subsection summarizes some main differences between MRFs and CRFs.

Formulation: In MRFs, the posterior is proportioned to the joint probability using the Baye's rule, and the joint probability is modeled by defining the likelihood and prior. While CRFs model the posterior directly. In MRFs, the unary and pairwise potentials are functions of observed data at individual site and only the labels, respectively. While in CRFs, the unary and pairwise potentials are functions of the whole observed data and labels.

Feature Space: In MRFs, since the distributions of the observed data should be modeled, low-dimension features, like color and motion, are used in common.

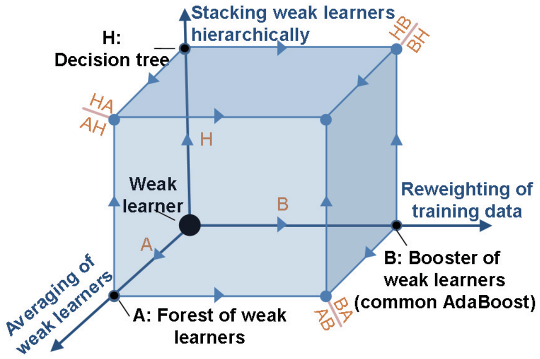

Stacking weak learners hierarchically
H: Decision tree
HB
BH
HA
AH
H
Weak learner
B
Reweighting of training data
A
B: Booster of weak learners (common AdaBoost)
Averaging of weak learners
A: Forest of weak learners
BA
AB

Fig. 1.13 The tree cube taxonomy of classifiers. The figure is taken from [66].

While in CRFs, more complex discriminative features would be selected to improve the predictive performance.

Performance: Compared with CRFs, MRFs can handle data missing problem and new class adding problem. While CRFs have better predictive performance since CRFs model the posterior directly. On the other hand, since CRFs relax the assumption of conditional independence of the observed data, they can incorporate global information in the model.

Training Data: MRFs can augment small number of expensive labeled data with large number of unlabeled data. While CRFs need much labeled data for training.

Data Modeling: In MRFs, appropriate distributions need to be selected to model the observed data. In CRFs, good classifier algorithms should be design for learning from labeled data.

Model Selection: At last, our question is which model should be selected in applications. For the tasks of segmentation for video without prior knowledge, like object cutout in video editing [63] and foreground segmentation in surveillance [62], since there is no labeled data or a few interactively labeled data, the MRF model would be selected in common. For the tasks of class labeling problem with large quantities of labeled data, like scene detection in dynamic image sequences [65], the CRF model is used commonly. Actually, the MRF and CRF formulations used in the applications of video segmentation do not strictly comply with the definition of MRFs and CRFs. For instant, the pairwise potential in MRFs is the function of not only the labels but also the intensity difference. In CRFs, the color features are often incorporated in the model by adding the same likelihood term as in MRFs (for example in [66]). This enforces the data independence assumption in CRFs.

1.6 Summary

In summary, this chapter overviews different technologies developed for each step of the pipeline for semantic object segmentation and discusses major challenges at different steps. In order to achieve good performance on semantic object segmentation, local appearance, local consistency and long-range contextual information need to be considered together. To capture local appearance, filter-banks, visual descriptors and their quantization schemes need to be well designed. They need to have both high discriminative power and good invariance to noise, clutters, and changes of illuminations and viewpoints. Because of the large number of image patches to be processed during object segmentation, computational efficiency is also an important issue to be considered. Conditional random fields provide a powerful framework to integrate local appearance, local consistency and long-range contextual information. However, it requires training data to be labeled at the pixel-level, which is expensive for a large number of object classes. Topic models can learn the models of object classes without supervision or with weak supervision. By including spatial structures, topic models are able to capture long-range contextual information as well as local consistency. However, its capability of modeling local appearance is relatively weak compared with discriminative approaches which use strong classifiers such as SVM and Boosting to model local appearance. It is expected to achieve better performance if the strengths of both generative models and discriminative models can be well combined. For video segmentation, we compare two main statistical frameworks, Markov random fields (MRFs) and conditional random fields (CRFs). The generative approach, MRFs, models the observation data by selecting some conditionally independent distributions. CRFs have better predictive performance since in CRFs the assumption of conditional independency for the observation data is relaxed. But to achieve good enough results, a large number of labeled data should be provided in CRFs. Actually in many real applications, the MRF and CRF model is combined to obtain better results.

References

1. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.
2. J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.
3. G. Csurka and F. Perronnin. An efficient approach to semantic segmentation. *International Journal of Computer Vision*, 2010.
4. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, 2001.
5. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

6. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.
7. B. C. Russell and A. Torralba. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008.
8. Z. Y. Yao, X. Yang, and S. C. Zhu. Introduction to a large scale general purpose groundtruth dataset: Methodology, annotation tool, and benchmarks. In *Proc. Int'l Conf. on EMMCVPR*, 2007.
9. T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence*, 1999.
10. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
11. X. Wang and E. Grimson. Spatial latent dirichlet allocation. In *Proc. Neural Information Processing Systems Conf.*, 2007.
12. T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43:29–44, 2001.
13. C. Schmid. Constructing models for content-based image retrieval. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2001.
14. J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43:7–27, 2001.
15. M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62:61–81, 2005.
16. D. Lowe. Distinctive image features from scale-invariant key points. *International Journal of Computer Vision*, 60:91–110, 2004.
17. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27:1615–1630, 2005.
18. N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.
19. Q. Zhu, M. Yeh, K. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.
20. J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. British Machine Vision Conference*, 2002.
21. P. Forssen. Maximally stable colour regions for recognition and matching. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007.
22. H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110:346–359, 2008.
23. S. Lazebnik, S. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27:1265–1278, 2005.
24. K. E. A. Sande, T. Gevers, and G.M. Snoek. Evaluation of color descriptors for object and scene recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.
25. M.A. Tahir, K. Sande, J. Uijlings, F. Yan, X. Li, K. Mikolajczyk, J. Kittler, T. Gevers, and A. Smeulders. Surreyuva-srkda method. In *Pascal VOC 2008 Workshop, Marseille, France*, 2008.
26. E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *Proc. European Conf. Computer Vision*, 2006.
27. F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proc. Int'l Conf. Computer Vision*, 2005.
28. D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.
29. F. Moosmann, B. Tigggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *Proc. Neural Information Processing Systems Conf.*, 2006.
30. C. Elkan. Using the triangle inequality to accelerate k-means. In *International Conference on Machine Learning*, 2003.

31. J.C. Van Gemert, J. Geusebroek, C.J. Veenman, and A.W.M. Smeulders. Kernel codebooks for scene categorization. In *Proc. European Conf. Computer Vision*, 2008.
32. J. Shotton, M. Johnson, and Cipolla. Semantic texton forests for image categorization and segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007.
33. P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 36:3–42, 2006.
34. F. Perronnin, C. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *Proc. European Conf. Computer Vision*, 2006.
35. M. Marszalek and C. Schmid. Accurate object localization with shape masks. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007.
36. S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80:300–316, 2008.
37. D. Cai, X. He, and J. Han. Efficient kernel discriminant analysis via spectral regression. In *Proc. IEEE Int'l Conf. Data Mining*, 2007.
38. D. Aldavert, A. Ramisa, R.L. Mantaras, and R. Toledo. Fast and robust object segmentation with the integral linear classifier. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2010.
39. X. He, R. S. Zemel, and M. A. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.
40. J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multiclass object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81:2–23, 2009.
41. A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19:854–869, 2007.
42. B.A. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *Proc. Int'l Conf. Computer Vision*, 2009.
43. X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. Int'l Conf. Computer Vision*, 2003.
44. X. He, R. S. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *Proc. European Conf. Computer Vision*, 2006.
45. A. Torralba, K.P. Murphy, and W. Freeman. Contextual models for object detection using boosted random fields. In *Proc. Neural Information Processing Systems Conf.*, 2004.
46. A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Proc. Int'l Conf. Computer Vision*, 2007.
47. A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In *Proc. Neural Information Processing Systems Conf.*, 2004.
48. X. Ma and W.E.L. Grimson. Learning coupled conditional random field for image decomposition with application on object categorization. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.
49. J. Reynolds and K. Murphy. Figure-ground segmentation using a hierarchical conditional random field. In *Proc. of Canadian Conference on Computer and Robot Vision*, 2007.
50. J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *Proc. Int'l Conf. Computer Vision*, 2005.
51. B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.
52. J. Verbeek and B. Triggs. Region classification with markov field aspect models. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007.
53. J. Verbeek and B. Triggs. Scene segmentation with conditional random fields learned from partially labeled images. In *Proc. Neural Information Processing Systems Conf.*, 2007.
54. T. L. Griffiths and M. Steyvers. Finding scientific topics. In *Proc. of the National Academy of Sciences of the United States of America*, 2004.

55. L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach teseted on 101 object categories. In *in Proc. IEEE CVPR Worshop of Generative Model Based Vision*, 2004.
56. T. S. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1:209–230m, 1973.
57. J Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.
58. D. Larlus, J. Verbeek, and F. Jurie. Category level object segmentation by combining bag-of-words models with dirichlet processes and random fields. *International Journal of Computer Vision*, 88:238–253, 2010.
59. G. Passino, I. Patras, and E. Izquierdo. Latent semantics local distribution for crf-based image semantic segmentation. In *Proc. British Machine Vision Conference*, 2009.
60. E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, 77:291–330, 2007.
61. L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. In *Proc. Int'l Conf. Computer Vision*, 2007.
62. J. Sun, W. Zhang, X. Tang, and H. Shum. Background cut. In *Proc. European Conf. Computer Vision*, 2006.
63. Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in nd images. In *Proc. Int'l Conf. Computer Vision*, 2002.
64. A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.
65. C. Wojek and B. Schiele. A dynamic conditional random field model for joint labeling of object and scene classes. In *Proc. European Conf. Computer Vision*, 2008.
66. P. Yin, A. Criminisi, J. Winn, and M. Essa. Tree-based classifiers for bilayer video segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007.
67. Y. Wang and Q. Ji. A dynamic conditional random field model for object segmentation in image sequences. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.