Int J Comput Vis (2009) 82: 185–204
DOI 10.1007/s11263-008-0194-9

Detection and Segmentation of Multiple, Partially Occluded Objects by Grouping, Merging, Assigning Part Detection Responses

Bo Wu · Ram Nevatia

Received: 7 July 2008 / Accepted: 31 October 2008 / Published online: 5 December 2008
© Springer Science+Business Media, LLC 2008

Abstract We propose a method that detects and segments multiple, partially occluded objects in images. A part hierarchy is defined for the object class. Both the segmentation and detection tasks are formulated as binary classification problem. A whole-object segmentor and several part detectors are learned by boosting local shape feature based weak classifiers. Given a new image, the part detectors are applied to obtain a number of part responses. All the edge pixels in the image that positively contribute to the part responses are extracted. A joint likelihood of multiple objects is defined based on the part detection responses and the object edges. Computation of the joint likelihood includes an inter-object occlusion reasoning that is based on the object silhouettes extracted with the whole-object segmentor. By maximizing the joint likelihood, part detection responses are grouped, merged, and assigned to multiple object hypotheses. The proposed approach is demonstrated with the class of pedestrians. The experimental results show that our method outperforms the previous ones.

Keywords Object detection · Object segmentation

1 Introduction

Detection and segmentation (*i.e.* accurate delineation) of objects of one or more given classes is of fundamental interest in computer vision. Traditionally, one would segment an image into regions, and then try to classify these regions as belonging to one of the desired classes. This approach works well when objects of interest have relatively homogeneous properties in some image attributes, such as intensity, color, or texture. For example, many early methods attempt to detect faces by segmenting skin color regions. However, for many common objects of interest, *e.g.* humans, the surfaces are not uniform and the texture can be arbitrarily complex due to clothing. In such cases, effective algorithms for bottom-up segmentation are difficult to devise; existing methods tend to over or under segment an image.

When multiple objects are present in the image and overlap one another, the problem becomes even more difficult, because the image appearance of multiple inter-occluded objects is not independent. If objects of interest are moving, motion-based segmentation can be more reliable, but even here, merging of motion blobs with adjacent objects and with shadows and reflections can be problematic.

In recent years, methods for direct detection of objects have become popular, and promising results have been achieved for several object classes, including faces (Viola and Jones 2001; Huang et al. 2007), pedestrians (Wu and Nevatia 2005, 2007c; Sabzmeydani and Mori 2007; Tuzel et al. 2007; Gavrila 2007; Viola et al. 2003; Zhu et al. 2006; Mikolajczyk et al. 2004), and cars (Wu and Nevatia 2007a; Schneiderman and Kanade 2000). In these methods, no prior segmentation is applied; rather, the image is scanned by windows of various size and a determination as to the presence of the desired object is made in this window. While such methods show good performance at the detection level, object segmentation is not very precise; typically a bounding box which contains the object as well as some of the background is detected. A more accurate delineation process may then be applied inside the bounding box, as in Leibe et al. (2005).

B. Wu (✉) · R. Nevatia
Institute for Robotics and Intelligent Systems,
University of Southern California, Los Angeles, CA 90089-0273,
USA
e-mail: bowu@usc.edu

R. Nevatia
e-mail: nevatia@usc.edu

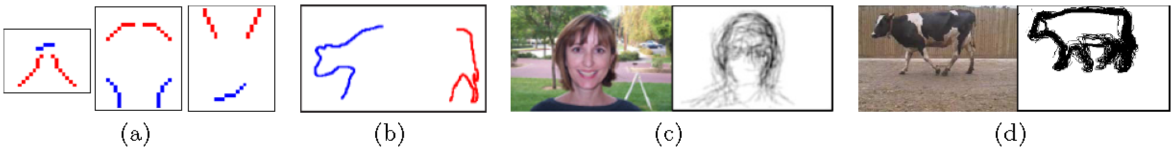


Fig. 1 Local shape features in Opelt et al. (2006), Shotton et al. (2005), Wu and Nevatia (2005): (**a**) Edgelets selected for people (Wu and Nevatia 2005); (**b**) Boundary fragments selected for cows (Opelt et al. 2006); (**c**) Feature responses of a face (Shotton et al. 2005); (**d**) Feature responses of a cow (Opelt et al. 2006)

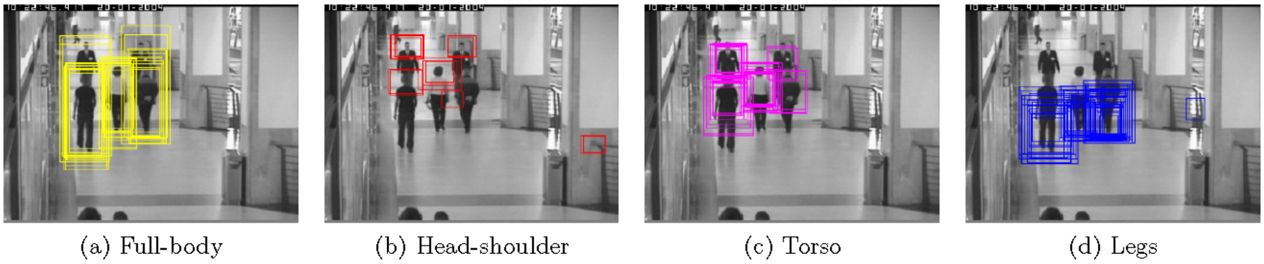

(a) Full-body (b) Head-shoulder (c) Torso (d) Legs

Fig. 2 Examples of part detection responses for pedestrians

Many existing methods select informative shape oriented features to model the appearance of the objects, such as the contour fragment features in Opelt et al. (2006), Shotton et al. (2005), and our *edgelet* features in Wu and Nevatia (2005). The selected features lie on the object boundary; their responses on the query image help to delineate the object (see Fig. 1). Based on this observation, we formulate the segmentation task as a binary classification problem, and design classifiers for both detection and segmentation based on local shape features.

For the cases with partial, inter-object occlusions, part based representations can be used. For each part, a detector is learned and the part detection responses are combined to form object hypotheses. The part detectors are typically applied to overlapping windows and the windows are classified independently. Consequently, one local feature may contribute to multiple overlapped responses for one object (see Fig. 2). Some false detections may also occur, as local features may not be discriminative enough. Because of poor image cues or partial occlusions, some object parts may not be detected. To get a one-to-one mapping from part detection responses to object hypotheses, we need to group the responses and explain inconsistency between the observation and the hypotheses.

When objects are close to one another, both the one-object-multiple-response problem and the part-object assignment problem require joint consideration of multiple objects, instead of treating them independently. One important component of the joint consideration is the analysis of the occlusion relation between multiple objects in the 2-D image space. To obtain an accurate occlusion model, pixel level segmentation of objects is necessary. We propose a unified framework using segmentation based joint analysis of multiple objects to group, merge and assign part responses. We demonstrate this approach with the object class of pedestrians. The experimental results show that our methods outperforms the previous ones.

Our previous work published in Wu and Nevatia (2005) and Wu and Nevatia (2007c) also addresses the problem of pedestrian detection. This paper describers several improvements and enhancements to the major components, including the part detectors, and joint analysis of multiple objects. Experimental results show that the new method significantly outperforms the old one. Additionally, the output of the earlier method consists only of bounding boxes of the objects, while the new method outputs pixel level segmentation.

The rest of this paper is organized as follows: Section 2 introduces the related previous works; Section 3 gives an outline of our approach; Section 4 describes our part detection module; Section 5 presents our segmentation method for individual separate objects; Section 6 presents our part based joint detection and segmentation method for multiple, inter-occluded objects; Section 7 shows the experimental results; some conclusions and discussions are given in the last section.

2 Related Work

The literature on object detection and segmentation in images is large and a complete survey would be difficult. As we take pedestrians as the class of interest for demonstration, we focus the introduction of related work mainly on the pedestrian related methods.

2.1 Detection of Individual, Separated Objects

Many previous efforts on object detection and segmentation focus on individual separate objects. These methods assume appearance independence between multiple objects, therefore they attempt to classify and estimate the status of each object independently.

For detection, many recent methods (*e.g.* Viola and Jones 2001; Huang et al. 2007; Sabzmeydani and Mori 2007; Tuzel et al. 2007; Dalal and Triggs 2005; Viola et al. 2003; Zhu et al. 2006; Schneiderman and Kanade 2000) learn object classifiers, whose input is a rectangular image sub-window and whose output is a prediction of the presence/absence of an object in this window. To capture the salient image characteristics of the objects class, a large variety of image features has been developed.

Some of these features are spatially global, *e.g.* the edge template in Gavrila and Philomin (1999), Gavrila (2000, 2007). However, most recent methods use local features, because the local features are less sensitive to occlusions and other types of partially missing observations. Some examples are the wavelet descriptors in Schneiderman and Kanade (2000), the Haar like features in Viola and Jones (2001), the sparse rectangle features in Huang et al. (2006, 2007), the SIFT like orientation features in Mikolajczyk et al. (2004), the Histogram of Oriented Gradients (HOG) descriptors in Dalal and Triggs (2005), the code-book of local appearance in Leibe et al. (2004, 2005), the boundary fragments in Opelt et al. (2006), the biologically-motivated sparse, localized features in Mutch and Lowe (2006), the shapelet features in Sabzmeydani and Mori (2007), the covariance descriptors in Tuzel et al. (2007), the motion enhanced Haar features in Viola et al. (2003), the Internal Motion Histograms (IMH) in Dalal et al. (2006), and the edgelet features used in our previous work (Wu and Nevatia 2005, 2007c). The above features are mostly shape oriented, because shape is the most consistent and salient image cue for many object classes.

To reduce the inner-class variation, the size and position of the object w.r.t. the rectangular sub-window are usually normalized. Therefore, to detect multiple objects, the object classifier needs to be applied to sub-windows of different sizes and positions. As the number of sub-windows to process is usually large, a cascade classifier structure proposed by Viola and Jones (2001) is widely adopted to improve the computational efficiency. One cascade classifier consists of a series of classifiers, called *layers*. If and only if one sub-window is classified as object by all layers, it is accepted as object; if and only if one sub-window is classified as non-object by any layer, it is rejected as non-object. Most of the non-object sub-windows are rejected by the first few layers. The layer classifiers can be learned by different learning methods, such as an SVM in Dalal and Triggs (2005), Dalal et al. (2006) and AdaBoost in Viola and Jones (2001), Viola et al. (2003), Huang et al. (2007), Zhu et al. (2006), Tuzel et al. (2007), Sabzmeydani and Mori (2007).

However, the sub-window classification results are not the final outputs of a detection system. One main post-process is to "merge" the positive responses having large overlap and expect that each of the resulting clusters corresponds to one object, *e.g.* the aggregate clustering algorithm used in Rowley et al. (1998) and the adaptive bandwidth mean-shift used in Tuzel et al. (2007). Usually, thresholding on overlap is used to determine if two responses are from the same object. Setting this threshold can be tricky when objects are close to one another.

Some other methods attempt to generate human hypotheses by directly grouping image features, *e.g.* the recent work by Sharma and Davis (2007). This method classifies and groups the edges in a sub-window of interest to form object hypothesis by an MRF-based approach. The output is a prediction of the presence of an object and its contour. However, the method has been applied to isolated human objects only. It is unclear as to how the method may be extended to the case of multiple, inter-occluded objects. An MRF based model of image edges for multiple objects would be very complicated, and searching for the best solution in a high dimensional joint space could be difficult.

2.2 Detection of Multiple, Occluded Objects

When there are multiple objects partially inter-occluding one another in the scene, the independent appearance assumption is not valid any more. Although the local feature based methods can work with partial occlusions to some extent, when the occlusion is significant, the whole-object classifier tends to miss the object, and the response clustering and object segmentation tend to merge close-by objects into one. (For static images, we do not expect to detect fully occluded objects, as no observation is available.)

Recently, part based representations and joint analysis of multiple objects have been adopted to solve partially occluded cases (*e.g.* Wu and Nevatia 2005, 2007c; Shet et al. 2007; Lin et al. 2007, 2004). In these methods, objects are represented as an assembly of several parts. Mohan et al. (2001) divide human body into four parts: head-shoulder, legs, left arm, and right arm. Shashua et al. (2004) divide human body into nine overlapping sub-regions. Some of the sub-regions correspond to natural human body parts, such as head, torso, arms, and legs; some do not. Mikolajczyk et al. (2004) divide human body into seven parts, face/head for frontal view, face/head for profile view, head-shoulder for frontal and rear view, head-shoulder for profile view, and legs. Lin et al. (2007) divide human body into four parts: head, torso, legs, and feet. Shet et al. (2007) divide human body into three parts: head, torso, and legs, besides a full-body model. Most of the existing methods do the partition based on natural human body structure, mainly because the natural body parts have relatively consistent appearance and are well defined.

For object parts, detectors are learned by supervised learning. One way to build a set of part detectors is to train them independently, like in Wu and Nevatia (2005, 2007c), Shet et al. (2007). However, this increases the time complexity of training linearly w.r.t. the number of parts. Another way is to build one part detector as a true subset of the whole-object detector. For example, in Lin et al. (2004) each sub-region detector use a subset of features of the whole-region detector and only the decision thresholds are different. The main limitation of this method is that a subset of features of the whole-object model may not be sufficient to construct a good part model.

For detection, the part detectors are applied to the input image and the detection responses are merged with some clustering method, as in the case of single object detector. After obtaining the part detection responses, some early methods (*e.g.* Mohan et al. 2001; Shashua et al. 2004; Mikolajczyk et al. 2004) do the part combination independently for each human. The combination is usually based on majority voting (*e.g.* checking if more than half of the parts are detected) or weighted sum (thresholding the weighted sum of the part detection confidences, where the weights are determined by the performance of the part detectors) of the part detection results. This type of methods does not consider the occlusion relation of multiple humans.

Recently, several methods are developed (Lin et al. 2007; Shet et al. 2007) to do joint part combination of multiple humans to detect humans with inter-occlusions. In these method, a joint image likelihood of multiple objects is computed by awarding successful part detection and penalizing missed detection of visible parts and false alarms. Different hypotheses configurations are tested, and the one with the highest likelihood is kept as the final interpretation of the image. The input of the part combination stage are the bounding boxes of parts. These are relatively coarse representations from which we can not get an accurate occlusion model. In addition, the errors from the overlap thresholding at the response merging stage are difficult to correct at the part combination stage. Different from the part combination methods, Leibe et al. (2005) propose an Implicit Shape Model (ISM) based approach to detect multiple humans. Joint analysis is performed to cover occluded objects.

Compared to the whole-object detector, the part based detection system has better performance on partially occluded cases. However, for very crowded situations, the performance of the current detection and segmentation methods is far from perfect. It is difficult to segment multiple humans in a large, dense crowd, because for each individual human, the visible part is usually very small. Some recent methods (*e.g.* Kong et al. 2006; Chan et al. 2008) attack this problem by reducing the objective from localization to counting, *i.e.* given an input image, estimate the number of present objects without explicitly telling their locations and sizes. These methods consider the crowd as a textured region, and estimate the number of objects by a regression method. One major assumption is that the crowd is homogeneous, *i.e.* the crowd only contains the objects of the interested class. This assumption limits the application of such methods.

2.3 Segmentation of Objects

The output of the object detection methods is a set of bounding boxes of the objects, which can be seen as a rough segmentation. However, an accurate pixel-level figure-ground segmentation is necessary for a number of high level tasks. For example, for tracking, we need to build appearance models for individual objects, which requires the knowledge of which regions in the image belong to the objects of interest. The main difference between the general image segmentation methods (*e.g.* Tu et al. 2001), and the segmentation of objects of a known class is the use of the prior knowledge, *i.e.* an object model, of the concerned class. In addition to guiding segmentation, the object models can also function as discriminative models for recognition and detection (*e.g.* Winn and Shotton 2006; Todorovic and Ahuja 2006; Opelt et al. 2006; Shotton et al. 2006; Kapoor and Winn 2006; Winn and Jojic 2005; Leibe et al. 2004; Pawan Kumar et al. 2005) or generative models for pose estimation (*e.g.* Bray et al. 2006).

Similar to the problem of object detection, many recent object segmentation methods build the object models based on some global or local image features instead of pixel intensity. Unlike the dominating popularity of shape oriented features in object detection, color, *e.g.* the mixtures of Gaussian color model in Shotton et al. (2005, 2006) and the kernel density estimation of color distribution in Zhao and Davis (2005), and texture, *e.g.* the texton in Shotton et al. (2006), are two other commonly used cues for segmentation.

When global features are used, the object models are sometimes equal to the features, *e.g.* the edge template models in Gavrila and Philomin (1999), Zhao and Davis (2005). When local features are used, we need some mechanism to organize the features to form the object models. Many existing segmentation methods use random field approaches, *e.g.* the Layout Consistent Conditional Random Field in Winn and Shotton (2006), the Located Hidden Random Field in Kapoor and Winn (2006), the texton based CRF in Shotton et al. (2006), the pose-specific MRF in Bray et al. (2006), the Pictorial Structure enhanced MRF in Pawan Kumar et al. (2005). The inference of the CRF models usually requires loopy belief propagation or sequential tree-reweighted message passing; while graph cut is a widely used solution for inference in the MRF models. These techniques are computationally expensive.

Although the random field based methods are usually for multiple classes of objects, they do not segment multiple objects of the same class. For example, the method in Shotton et al. (2006) gives a single label to several cows in an image. Segmenting individual objects is obviously necessary for many tasks. The segmentation accuracy of the method in Shotton et al. (2006) is also rather low for specific categories; for example, the accuracy on humans is only about 62.1%.

Some other methods use constellation type models to organize the local features, *e.g.* the Boundary-Fragment-Model in Opelt et al. (2006), and the Implicit Shape Model in Leibe et al. (2004, 2005). These models are star-shaped, which can be inferred efficiently by a Hough Transformation much more efficiently. However, both the random field methods and the constellation methods usually assume a fixed object size so that the solution space is greatly restricted. Some of these methods (*e.g.* Todorovic and Ahuja 2006; Winn and Shotton 2006; Opelt et al. 2006; Kapoor and Winn 2006; Shotton et al. 2006; Winn and Jojic 2005; Zhao and Davis 2005; Pawan Kumar et al. 2005) result in simultaneous detection and segmentation.

Unlike the random field based approaches and the constellation model based approaches, the boosting methods originally proposed for detection encode the shape of the objects by including a number of local features within the sample window. The relative positions of these local features model the global shape implicitly. Although some existing methods (*e.g.* Opelt et al. 2006; Shotton et al. 2006) use boosting as feature selector for segmentation, none of them directly learn the ensemble classifier as a segmentor.

Segmentation is usually performed after object detection, because a known location and size facilitate the segmentation operation. However, some existing methods utilize segmentation to improve detection in a feedback loop. For example, Zhao and Davis (2005) utilize color based segmentation to reduce the effect of background clutters on edge template matching; Leibe et al. (2004, 2005) utilize the top-down segmentation to refine and verify object hypotheses with a Minimal Description Length (MDL) based approach.

3 Outline of Our Approach

Figure 3 shows an overview diagram of our approach. We define a part hierarchy for an object class, in which each part is a sub-region of its parent, except for a whole-object node. Because building part detectors independently is time consuming and building them as sub-set of the whole-object detector may not achieve a desirable accuracy, we choose a tradeoff between these two approaches. For each part, a detector is learned by a *Cluster Boosted Tree* (CBT) method (Wu and Nevatia 2007a). The image features used are the edgelet features (Wu and Nevatia 2005). A child node in the hierarchy inherits image features from its parent node and if a target performance can not be achieved only from the inherited features, more features are selected and added to the child node.

For the whole-object node, in addition to the detector, a pixel-level segmentor is learned. We formulate segmentation as a binary classification problem and train the segmentor by a supervised learning algorithm. In the training procedure, for each feature in a large feature pool, a pair of weak classifiers for detection and segmentation is built. A boosting algorithm is adopted to select informative features from the pool. The input of the segmentor is an image sample and a pixel location within the sample window; the output

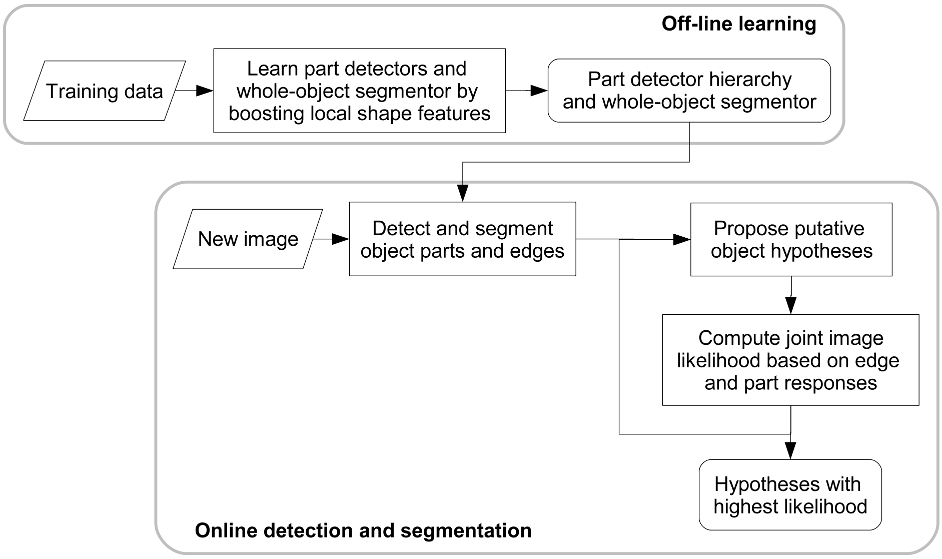

Off-line learning
Training data
Learn part detectors and whole-object segmentor by boosting local shape features
Part detector hierarchy and whole-object segmentor
New image
Detect and segment object parts and edges
Propose putative object hypotheses
Compute joint image likelihood based on edge and part responses
Hypotheses with highest likelihood
Online detection and segmentation

Fig. 3 Overview diagram of our approach

is the figure-ground prediction. We define an effectiveness measure of an edgelet feature for its local neighborhood. A figure-ground distribution weighted by the effectiveness measure is learned, based on which a local weak segmentor is determined. The final boosted ensemble classifier with a cascade decision strategy works as a detector as well as a segmentor.

Given a new image, all the part detectors are applied. The image edge pixels that positively contribute to the detection responses are extracted. The part responses and the object edges form an informative intermediate representation of the original image. We do not divide the tasks of merging responses and part combination into two separate stages; instead, we attempt to solve them under the same framework. From the part detection responses, multiple object hypotheses are proposed. For each hypothesis, a pixel-level segmentation is obtained by applying the whole-object segmentor, and the silhouette is extracted. We perform occlusion reasoning for multiple objects, and compute a 1-D silhouette based visibility score, instead of the region based 2-D visibility score in the previous methods (Wu and Nevatia 2005; Shet et al. 2007; Lin et al. 2007). We define a joint image likelihood of multiple objects, which gives rewards for successful detection of visible parts, and penalties for missed detections and false alarms. The likelihood also includes a matching score between the visible silhouettes and the object edges. Our joint analysis method enforces the *exclusiveness* of low level features, *i.e.* one image feature can contribute to at most one hypothesis.

Our approach is a unified MAP framework that solves part merging, grouping, and assigning together. The main contributions of this method include:

1. a part hierarchy design that enables efficient learning of part detectors by feature sharing,
2. an individual object segmentation method based on boosted classifiers,
3. an accurate occlusion reasoning approach based on object silhouettes, and
4. a joint image likelihood based on both the detection responses and the object edges, which are assigned to object hypotheses exclusively.

We demonstrate our approach on the class of pedestrians. Every module in our approach contributes to the robustness of the whole system. Though the situations to the advantage of any single module may not occur frequently, together they result in a statistically significant improvement compared to the previous methods.

Parts of the boosting based individual object segmentation method have been published in Wu and Nevatia (2007b); parts of the part based joint analysis method for multiple, partially inter-occluded objects have been published in Wu et al. (2008). This paper provides a unified and detailed presentation, and additional experimental results.

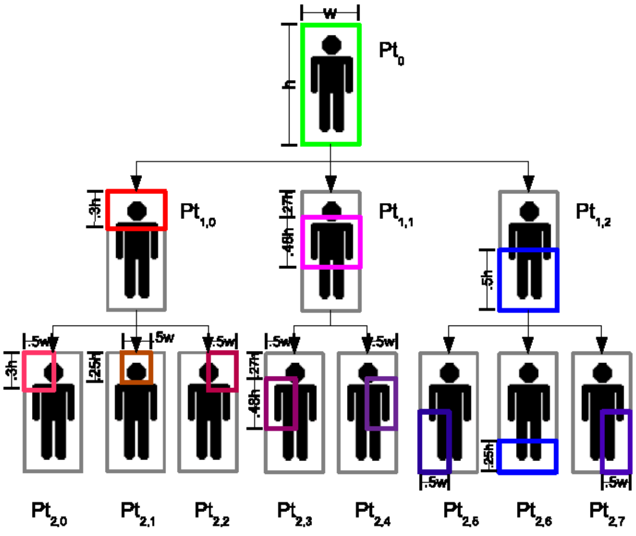


Fig. 4 Hierarchy of human body parts. (Pt_0 is full-body; $Pt_{1,0}$ head-shoulder; $Pt_{1,1}$ torso; $Pt_{1,2}$ legs; $Pt_{2,0}$ left shoulder; $Pt_{2,1}$ head; $Pt_{2,2}$ right shoulder; $Pt_{2,3}$ left arm; $Pt_{2,4}$ right arm; $Pt_{2,5}$ left leg; $Pt_{2,6}$ feet; $Pt_{2,7}$ right leg. The *left* and *right* sides here are w.r.t. the 2-D image space)

4 Hierarchy of Body Part Detectors

We use the class of pedestrians to illustrate and validate our approach. We define a part hierarchy for human body, which consists of three levels including a full-body node and 11 body part nodes. See Fig. 4.

4.1 Learning Part Detectors

For each node, a detector is learned. Because we define the part hierarchy such that the region of one child node is a subregion of its parent node, feature sharing between the parent and child nodes is possible. For each part node, a boosting algorithm is applied to select informative local shape features and construct a classifier as detector.

Following our previous work (Wu and Nevatia 2005, 2007c), the image features used are *edgelets*. Based on one edgelet, a weak classifier for detection task is defined. The weak detection classifier is a function from the image space $\mathcal{X}$ to a real valued object/non-object classification confidence space. The definition of the weak detection classifiers is the same as that in Wu and Nevatia (2007a). Given a labeled sample set $S = \{(\mathbf{x}_i, y_i)\}$, where $\mathbf{x} \in \mathcal{X}$ is the image patch, and $y_i = \pm 1$ is the class label of $\mathbf{x}$, the weak detection classifier $h^{(d)}$ is defined as a piecewise function:

$$\begin{aligned} &\text{if} \quad f(\mathbf{x}) \in \left[\frac{j-1}{n_d}, \frac{j}{n_d}\right), \\ &h^{(d)}(\mathbf{x}) = \frac{1}{2} \ln\left(\frac{W_+^j + \epsilon}{W_-^j + \epsilon}\right), \quad j = 1, \ldots, n_d \end{aligned} \tag{1}$$

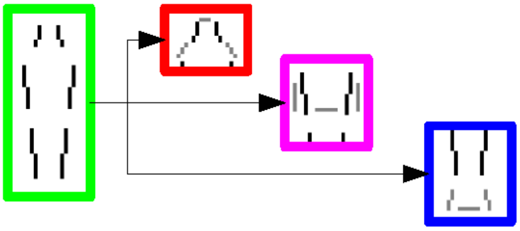


Fig. 5 Illustration of feature sharing in part detector hierarchy. (The *black* points are the inherited features, and the *gray* are the newly selected features)

where n_d is the number of sub-ranges of the feature value for detection (in our experiment $n_d = 32$), ϵ is a smoothing factor (Schapire and Singer 1999), and $W_\pm$ is the probability distribution of the feature value for positive/negative samples, implemented as a histogram:

$$W_\pm^j = P\left(f(\mathbf{x}) \in \left[\frac{j-1}{n_d}, \frac{j}{n_d}\right), \ y = \pm 1\right), \quad j = 1, \ldots, n_d \tag{2}$$

Before the regular boosting procedure, the detector of one node, except for the whole-object node, inherits from its parent node all the edgelet features that have a minimum number of points (4 points in our experiments) falling in its sub-region. For each inherited edgelet, the points that are out of the part's sub-region are removed. With the inherited features fixed, the classification functions of the inherited features and the cascade decision strategy are re-trained by a boosting algorithm. The detector usually can not achieve a desirable target accuracy from only the inherited features. The regular boosting algorithm is then applied to add more features to the classifier. Figure 5 gives an illustration of the feature sharing.

In practice, the learning algorithm used is the CBT method (Wu and Nevatia 2007a). This method builds tree structured object classifier without a predefined sub-categorization, such as viewpoint categories. (The detector learning algorithm is not the focus of this paper. More details of the CBT method can be found in Wu and Nevatia 2007a.) More details of the experimental setting are given later in Sect. 7.1.

The difference between this part detection system and that in our previous method (Wu and Nevatia 2005, 2007c) is mainly twofold. First, in our previous method, only the first two levels of the hierarchy (full-body, head-shoulder, torso, and legs) are used; here, a finer partition enables the algorithm to work with greater partial occlusions. Second, in our previous method, the *Vector Boosting Tree* (VBT) method (Huang et al. 2005) is used to learn view-based part classifiers, and there is no feature sharing between parts; here, we use the CBT learning method, which has better performance than the view-based VBT method for pedestrian detection (Wu and Nevatia 2007a), and we share features between different parts to improve the computational efficiency.

4.2 Detecting Body Parts and Object Edges

Given a new image, the part detectors are applied. In addition to part responses, we extract image edges that correspond to objects. For each edgelet feature f in the classifier, we call it a *positive feature* if it has higher average matching score on the positive (object) class than on the negative (non-object) class, *i.e.*

$$E\{f(\mathbf{x}) | \mathbf{x} \in \mathcal{X}_+\} > E\{f(\mathbf{x}) | \mathbf{x} \in \mathcal{X}_-\} \tag{3}$$

where $\mathcal{X}_\pm$ is positive/negative sample space. The average matching scores are evaluated during the off-line learning stage. For one sub-window that is classified as object, the positive features in the sub-window are ranked according to their matching scores. The positive features with the top 5% scores are retained.

Because one detector usually contains about one thousand positive features, a large number of edgelets are retained for one image. Some of these edgelets correspond to the same edge pixels. We apply a clustering algorithm to prune redundant edgelets. An edgelet consists of a chain of 2-D points. Denote the positions of the points in an edgelet E by $\{\mathbf{u}_i\}_{i=1}^k$, where k is the length of the edgelet. Given two edgelets E_1 and E_2 with the same length, we define an affinity between them by

$$A(E_1, E_2) \triangleq \frac{1}{k} \sum_{i=1}^{k} \langle \mathbf{u}_{1,i} - \bar{\mathbf{u}}_1, \mathbf{u}_{2,i} - \bar{\mathbf{u}}_2 \rangle \cdot e^{-\frac{1}{2}\|\bar{\mathbf{u}}_1 - \bar{\mathbf{u}}_2\|^2} \tag{4}$$

where $\bar{\mathbf{u}}$ is the average position of $\{\mathbf{u}_i\}$. If the two features have different numbers of points, k_1 and k_2, they are first aligned by their center points, and the longer feature is then truncated to the length of the shorter one by removing points from the two ends. The affinity defined by (4) multiplied by a factor of $\frac{\min\{k_1,k_2\}}{\max\{k_1,k_2\}}$ is taken as the affinity for these edgelets.

The clustering algorithm used is an iterative algorithm. We find the edgelet with the highest feature response, and then remove all edgelets with an affinity larger than a given threshold (0.01 in our experiments) to it. This procedure is repeated until all object edgelets are examined. The remaining edgelets are the observations that support the putative object hypotheses. See Fig. 6 for an example. Compared to the general edge based image segmentation methods, where all edges are retained, our edge extraction method attempts to remove edges from background clutters and focuses on object shapes. These object edges, together with the bounding boxes, are the input for the joint analysis of multiple objects.

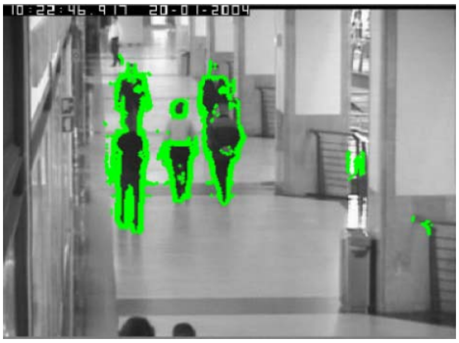


Fig. 6 Extracted object edgelet pixels

5 Segmentation of Individual Object

For the whole-object node, we learn a pixel-level figure-ground segmentor in addition to the detector. Note that we do not learn segmentors for the other body parts. Because the whole-object segmentor is based on local features, even when the object is partially occluded, the whole-object segmentor can still segment the visible part well based on the visible features. For the occluded parts, it tends to output a prior shape that is learned from the training data.

5.1 Design of the Weak Segmentator

Similar to the learning of part detectors, we build the whole-object segmentor by boosting simple feature based weak classifiers. For one edgelet feature f, in addition to the weak classifier for detection $h^{(d)}$ described in Sect. 4.1, we build a weak classifier for segmentation $h^{(s)}$, *i.e.* a pair of classifiers sharing the same feature.

Note that there are several feature sharing concepts in our system. In the CBT method for detection (Wu and Nevatia 2007a), feature sharing is between the weak classifiers for different sub-categories of the object class; in the learning of part detector hierarchy, feature sharing is between different overlapping parts; here, feature sharing is between the weak detectors and the weak segmentors.

The weak segmentation classifier is a function from the space $\mathcal{X} \times \mathcal{U}$ to a real valued figure-ground classification confidence space, where $\mathcal{U}$ is the 2-D image coordinate space, *i.e.* $\mathcal{U} = \mathbb{Z}^+ \times \mathbb{Z}^+$, where $\mathbb{Z}^+$ is the set of all non-negative integers. Intuitively, a local feature only contributes to the shape around its neighborhood. It is inefficient to estimate the status of the feet from an edgelet falling on the head-top. Based on this observation, we define an *effective field* of the edgelet based on a saliency decay function. This design is motivated by the tensor voting method for shape grouping (Medioni et al. 2000). See Fig. 7(a) for an illustration. O is a point on an edgelet feature, whose normal $\mathbf{n}$ and tangent $\mathbf{v}$ are known, P is a neighbor of O, and $\overset{\frown}{OP}$ is the arc of the *osculating circle* at O that goes through P. The effect of O on P is defined by

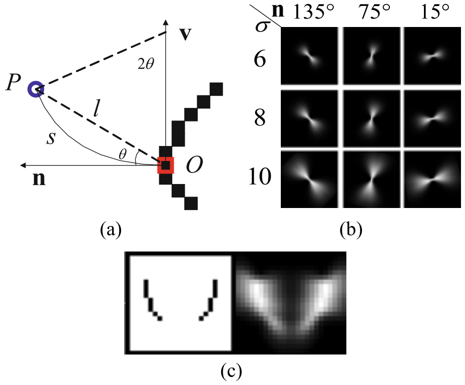


Fig. 7 Effective field: (**a**) definition of effectiveness; (**b**) effective field bases of individual edge points; (**c**) effective field of an edgelet feature

$$DF(s, \kappa, \sigma) = \exp\left(-\frac{s^2 + c\kappa^2}{\sigma^2}\right) \tag{5}$$

where l is the Euclidean distance between O and P, θ is the angle between $\mathbf{n}$ and $\overrightarrow{OP}$, $s = \frac{l\theta}{2\sin\theta}$ is the length of the arc $\overset{\frown}{OP}$, $\kappa = \frac{2\sin\theta}{l}$ is the curvature, c is a constant that controls the decay with high curvature, and σ is the scale of analysis, which determines the size of the effective field. Note that σ is the only free parameter. In practice, σ is quantized to five values, 2, 4, 6, 8, and 10, according to the size of our training samples, and the normal orientation of the edgelet point is quantized to six bins, $[\frac{\pi}{6}(i-1), \frac{\pi}{6}i)$, $i = 1, \dots, 6$. There are thus overall 30 bases of the effective field (see Fig. 7(b)). For a k point edgelet, denoted by $\mathbf{F}_i$ the effective field of the i-th point, the effective field of the whole feature is then defined by

$$\mathbf{F}(\mathbf{u}) = \max\{\mathbf{F}_1(\mathbf{u}), \dots, \mathbf{F}_k(\mathbf{u})\}, \quad \mathbf{u} \in \mathcal{U} \tag{6}$$

Figure 7(c) shows the effective field of an edgelet feature.

The definition of the weak segmentation classifiers is similar to that of detection. For the positive training samples, their segmentation ground-truth are given as binary masks. Let $S_+ = \{(\mathbf{x}_i, y_i = 1, \mathbf{m}_i)\}$ be the positive sample set, where $\mathbf{m}$ is the segmentation mask that has the same dimension as $\mathbf{x}$. $\mathbf{m}(\mathbf{u}) = +1$ means that the pixel $\mathbf{u}$ belongs to figure; $\mathbf{m}(\mathbf{u}) = -1$ means that the pixel $\mathbf{u}$ belongs to background. Assume that the effective field $\mathbf{F}$ of a feature f has been determined (how to optimize the shape of the effective field is described later in Sect. 5.2.2). Similar to the weak detection classifier $h^{(d)}$, the weak segmentation classifier $h^{(s)}$

is also defined as a piecewise function:

$$\begin{aligned}&\text{if}\quad f(\mathbf{x}) \in \left[\frac{j-1}{n_s}, \frac{j}{n_s}\right),\\ &\quad h^{(s)}(\mathbf{x};\mathbf{u}) = \frac{1}{2}\ln\left(\frac{\mathbf{W}_+^j(\mathbf{u})+\epsilon}{\mathbf{W}_-^j(\mathbf{u})+\epsilon}\right), \quad j=1,\dots,n_s\end{aligned} \tag{7}$$

where n_s is the number of sub-ranges of the feature value for segmentation (in our experiments $n_s = 16$), and $\mathbf{W}_\pm(\mathbf{u})$ is the feature value histogram of figure/ground pixels weighted by the effective field:

$$\begin{aligned}&\mathbf{W}_\pm^j(\mathbf{u}) = \mathbf{F}(\mathbf{u}) \cdot P\left(f(\mathbf{x}) \in \left[\frac{j-1}{n_s}, \frac{j}{n_s}\right),\ \mathbf{m}(\mathbf{u}) = \pm 1\right),\\ &\quad j = 1,\dots,n_s\end{aligned} \tag{8}$$

In practice, both $h^{(d)}$ and $h^{(s)}$ are implemented as look-up-table (LUT), and nd and n_s are the number of bins in the tables. The difference is that each bin of $h^{(d)}$ is a real valued scalar, while each bin of $h^{(s)}$ is a real valued matrix.

5.2 Boosting Ensemble Classifier for Segmentation and Detection

Let $\mathcal{H}$ be the weak classifier pool that consists of the weak classifier pairs built from all possible edgelets. Each element in the pool is a pair of weak detection and segmentation classifiers, *i.e.* $(h^{(d)}, h^{(s)})$. We use a variation of boosting algorithm to learn an ensemble classifier from $\mathcal{H}$ as strong detector and segmentor.

5.2.1 Sample Weight Evolution

One important feature of boosting algorithms is the sample weight evolution. For traditional detection problems, each sample is assigned a real valued weight $D^{(d)}$ representing its importance or difficulty. During the boosting procedure, the weights of the misclassified samples are increased while those of the correctly classified samples are decreased, so that more and more attention is paid to the difficult part of the sample space. For segmentation, not only do the difficulties of different samples vary, but the difficulties of different positions of the same sample also vary. Intuitively, for the less articulated parts of the human body, *e.g.* torso, segmentation is relatively easy, not much more than a constant mask; for the highly articulated part, *e.g.* legs, more features need to be evaluated before making the final decision. Therefore, for segmentation, we assign a weight field $\mathbf{D}^{(s)}$ to each positive sample.

$\mathbf{D}^{(s)}(\mathbf{u})$ represents the importance of the pixel at position $\mathbf{u}$. During the boosting procedure, the weight fields for segmentation are evolved in the same way as the weights for detection. Let the pair of weak detector and segmentor selected at the t-th boosting round be $(h_t^{(d)}, h_t^{(s)})$, and the sample weights for detection and segmentation be $D_t^{(d)}$ and $\mathbf{D}_t^{(s)}$ respectively. For all samples, the sample weights for detection of the $t+1$-th round are calculated by

$$D_{t+1}^{(d)}(\mathbf{x}) = D_t^{(d)}(\mathbf{x})\exp\left[-y h_t^{(d)}(\mathbf{x})\right], \quad \forall \mathbf{x} \in S \tag{9}$$

For all positive samples, the sample weights for segmentation of the $t+1$-th round are calculated by

$$\begin{aligned}&\mathbf{D}_{t+1}^{(s)}(\mathbf{x};\mathbf{u}) = \mathbf{D}_t^{(s)}(\mathbf{x};\mathbf{u})\exp\left[-\mathbf{m}(\mathbf{u}) h_t^{(s)}(\mathbf{x};\mathbf{u})\right],\\ &\quad \forall \mathbf{u} \in \mathcal{U}\end{aligned} \tag{10}$$

5.2.2 Optimization of Weak Classifier

At each boosting round, the best weak classifier pair is selected from $\mathcal{H}$, where two components need to be optimized: the edgelet feature and the effective field. The edgelet features are enumerated in the feature pool. An effective field is defined by the shape of its edgelet and the parameter σ. As we allow different σ's for different points in one edgelet, for a k point edgelet there are 5^k possible field shapes. When the sample size is 24 × 58 pixels, there are overall 857,604 possible edgelets. It would be very time consuming to perform brute force search in the Cartesian space. Instead, we separate the optimization into two steps: first search for the best edgelet with a default σ value, and then search for the best σ value.

With a fixed σ, the best edgelet is selected according to the following criterion:

$$\begin{aligned}\left(h_t^{(d)}, h_t^{(s)}\right) = \mathop{\arg\min}_{(h_t^{(d)}, h_t^{(s)}) \in \mathcal{H}} &\left\{\lambda 2 \sum_j \sqrt{W_+^j W_-^j}\right.\\ &\left. + (1-\lambda)\frac{1}{\nu}\sum_j \sum_{\mathbf{u}\in\mathcal{U}} \sqrt{\mathbf{W}_+^j(\mathbf{u})\mathbf{W}_-^j(\mathbf{u})}\right\}\end{aligned} \tag{11}$$

where $\nu = \sqrt{\sum_j \sum_{\mathbf{u}} \mathbf{W}_+^j(\mathbf{u}) \sum_j \sum_{\mathbf{u}} \mathbf{W}_-^j(\mathbf{u})}$ is a normalizing factor. This criterion encodes the discriminative power of the feature for both detection and segmentation. The coefficient λ represents the relative importance of the two tasks. In our experiments, $\lambda = 0.7$.

The value of σ is optimized in a greedy way. At one time, the σ of one edgelet point is optimized, while the others remain fixed. Figure 8 shows the first several selected features and their learned segmentors. It can be seen that they are evenly distributed and correspond to natural body parts. More experimental results of individual object segmentation are given later in Sect. 7.2.

Figure 9 shows the full algorithm of simultaneous learning of detector and the segmentor. The output of this algorithm is an ensemble classifier with a cascade decision strategy for detection. As segmentation is a balanced classification problem, we take the default threshold to be zero, *i.e.*

the pixel $\mathbf{u}$ of $\mathbf{x}$ is classified as figure, if and only if

$$H_T^{(s)}(\mathbf{x};\mathbf{u}) = \sum_{t=0}^{T} h_t^{(s)}(\mathbf{x};\mathbf{u}) > 0 \qquad (12)$$

The segmentation result by the boosted classifier is refined by applying twice *erosion* and twice *dilation* operations to remove some noises and fill some small holes. For simplicity of presentation, we use the cascade classifier structure to demonstrate the algorithm. However, in practice, this approach is integrated with the CBT learning method (Wu and Nevatia 2007a), where each branch of a tree classifier is a cascade. In addition to the weak classifiers selected

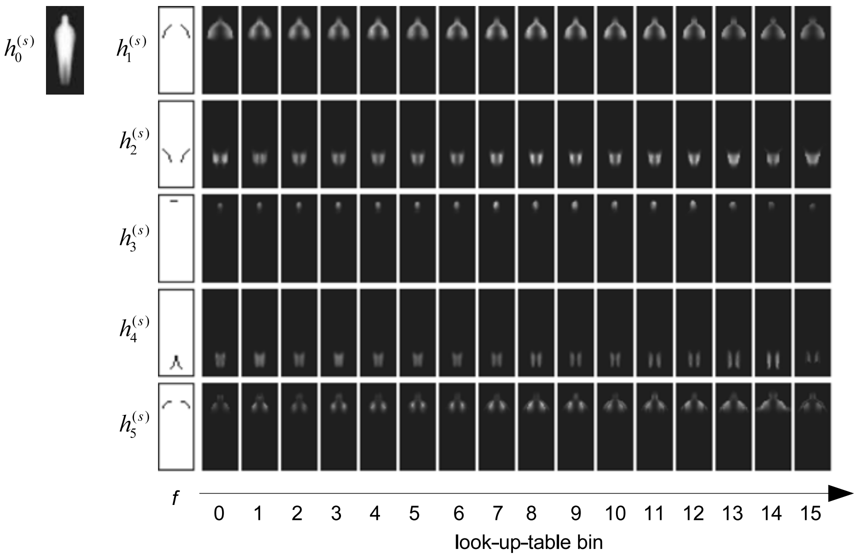


Fig. 8 The first five features selected and their segmentors learned for pedestrians. (The 0-th segmentor is the prior distribution. Each edgelet based weak segmentor is implemented as a histogram. Each bin of the histogram is a real-valued matrix defined by (7) with the same dimension of the training samples. In our experiments, a segmentor histogram has 16 bins. In this figure, we visualize the matrices of the histogram bins by normalizing them to [0, 255] gray scales. *White* is for larger value and *black* for lower value)

```
• Given the initial sample set $S = S_+ \cup S_-$, where $S_+ = \{(\mathbf{x}_i, +1, \mathbf{m}_i)\}$ and $S_- = \{(\mathbf{x}_i, -1)\}$, and a negative images set;
• Set the algorithm parameters: the maximum weak classifier number $T$, the positive passing rates $\{P_t\}_{t=1}^{T}$, the target false alarm rate $F$, the relative importance of detection to segmentation $\lambda$, and the threshold for bootstrapping $\theta_B$;
• Initialize the sample detection weights $D_0^{(d)}(\mathbf{x}) = \frac{1}{\|S\|}$ for all samples, the sample segmentation weight fields $\mathbf{D}_0^{(s)}(\mathbf{x}) = \frac{1}{\|S_+\|\|\mathbf{x}\|}$ for all positive samples, the current false alarm rate $F_0 = 1$, and $t = 0$;
• Construct the weak classifier pool, $\mathcal{H}$, from the edgelet features;
• While $t < T$ and $F_t < F$ do

    1. Search for the best edgelet
        (a) For each pair $(h^{(d)}, h^{(s)})$ in $\mathcal{H}$, generate the effective field for segmentation with a default value of $\sigma$ (=4), calculate $h^{(d)}$ and $h^{(s)}$ by (1) and (7) respectively. $W_\pm$ and $\mathbf{W}_\pm$ are calculated under the weight distribution $D_t^{(d)}$ and $\mathbf{D}_t^{(s)}$ respectively;
        (b) Select the best weak classifier pair from the classifier pool $\mathcal{H}$ according to (11);
    2. Search for the best shape of the effective field
        (a) For each point of the edgelet, set $\sigma = 2, 4, 6, 8, 10$, find the best value according to (11).
        (b) With the new effective field, recompute the classification function of $h_t^{(s)}$ by (7).
    3. Update sample weights by (9) and (10), and normalize $D_{t+1}^{(d)}$ and $\mathbf{D}_{t+1}^{(s)}$ to p.d.f.
    4. Select the threshold $b_t$ for the partial sum $H_t^{(d)}$, so that a portion of $P_t$ positive samples are accepted; and reject as many negative samples as possible;
    5. Remove the rejected samples from $S$. If the remaining negative samples are less than $\theta_B$ percent of the original, refill $S_-$ by bootstrapping on the negative image set;

• Output $\{(h_t^{(d)}, h_t^{(s)}), b_t\}$ as the cascade classifier for detection and segmentation.
```

Fig. 9 Algorithm of simultaneously learning detector and segmentor. In our experiments, $T = 1{,}000$, $F = 10^{-6}$, $\lambda = 0.7$, and $\theta_B = 75\%$. The setting of $\{P_t\}$ is similar to the original cascade's layer acceptance rates. The cascade is divided into 20 segments, the lengthes of which grow gradually. The weak classifiers at the end of the segments have positive passing rate of 99.8%, and the other weak classifiers have passing rate of 100.0%

by the boosting algorithm, we use the prior figure-ground distribution as the first weak segmentation classifier $h_0^{(s)}$ (see Fig. 8):

$$\forall \mathbf{x}, \quad h_0^{(s)}(\mathbf{x};\mathbf{u}) = \frac{1}{2}\ln\left(\frac{W_+(\mathbf{u})+\epsilon}{W_-(\mathbf{u})+\epsilon}\right) \tag{13}$$

where $W_\pm(\mathbf{u})$ is the prior probabilities of figure/ground labels of the pixel at the position $\mathbf{u}$:

$$W_\pm(\mathbf{u}) = P(\mathbf{m}(\mathbf{u}) = \pm 1) \tag{14}$$

6 Joint Analysis for Multiple Objects

For multiple overlapping objects, joint consideration is necessary. Similar to the previous methods (Wu and Nevatia 2005; Shet et al. 2007; Lin et al. 2007), our joint analysis algorithm takes the detection results as input, and searches for the multiple object configuration with the best image likelihood. The difference is that we enforce feature exclusiveness among multiple hypotheses, compute a 1-D silhouette based *visibility score* for occlusion reasoning, and add the object edge information into the likelihood definition. Figure 10 lists the main steps of the joint analysis algorithm.

1. Propose initial object hypotheses sorted such that the y-coordinates of their feet are in descending order.
2. Segment object hypotheses and extract their silhouettes.
3. Examine the hypotheses one by one, from front to back. For each hypothesis H, compare two multi-object configurations: with and without H
   (a) For the two configurations, compute the joint occlusion map for silhouettes of multiple objects;
   (b) Match the detection responses and object edgelets with visible silhouettes;
   (c) Compute the image likelihood with H, $P_w(H)$, and the likelihood without H, $P_{w/o}(H)$;
   (d) If $P_w(H) > P_{w/o}(H)$, accept the hypothesis; otherwise reject it.
4. Output all remaining hypotheses.

Fig. 10 Searching for the best multiple object configuration

6.1 Proposing Object Hypotheses

Initially, object hypotheses are proposed from the detection responses of a subset of parts. For pedestrians, we use full-body, head-shoulder, left/right shoulder, and head to propose the initial hypotheses. During detection, only the part detectors used for hypothesis proposals are applied to the whole image; the others are applied to the local neighborhood around the initial hypotheses. The hypotheses with a large overlap ratio, defined as the area of their intersection over the area of their union, are merged. Different from the traditional merging step (Rowley et al. 1998), we use a high overlap threshold (0.7 in our experiments) to obtain a set of "under-merged" responses, in which one object may have multiple hypotheses but hypotheses of different objects are unlikely to be merged. Although this under-merging reduces the search space, it tends to retain the responses of close objects separate for further joint analysis. We sort the object hypotheses by their vertical coordinates such that their y-coordinates are in a descending order. See Fig. 11(a) for an example.

6.2 Joint Occlusion Map of Silhouettes

For each hypothesis, segmentation is computed by applying the boosted whole-object segmentor and its silhouette is extracted. Same as in Wu and Nevatia (2005), Shet et al. (2007), Lin et al. (2007), we assume that objects are on a ground plane and that the camera looks down towards the plane, so that the relative depths of the objects can be inferred from their image coordinates. We render the segmentation masks of the ordered hypotheses by a z-buffer-like

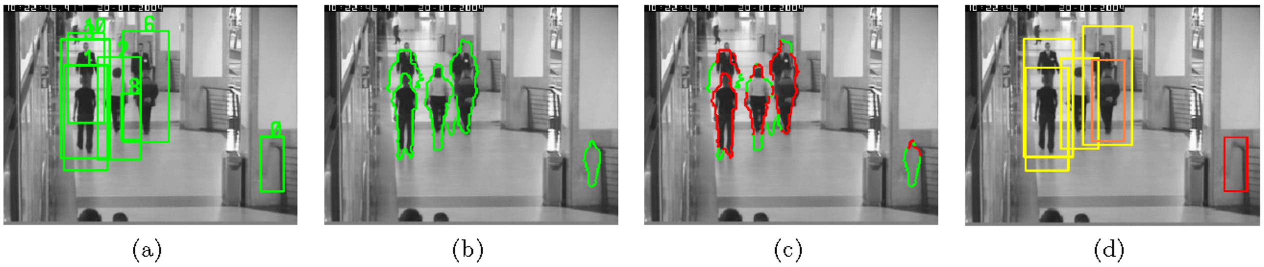
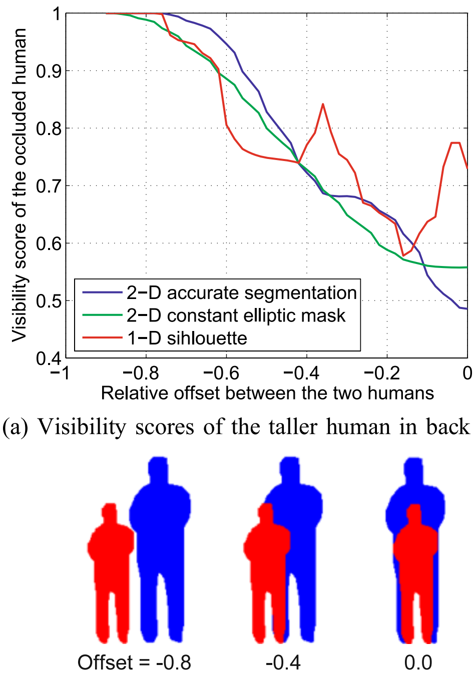


Fig. 11 (Color online) Computing joint image likelihood for multiple objects. (**a**) The examined multiple object configuration. (**b**) The visible silhouettes obtained by occlusion reasoning. (**c**) The parts of the silhouettes that have matched edgelets (*red* points). (**d**) Result of matching full-body detection responses in Fig. 2(**a**) with the current hypotheses (*yellow*: matched responses; *orange*: response not matched with any hypothesis; *red*: hypothesis without matched response)

method, and remove the invisible parts of the silhouettes that are out of image frame or occluded by other objects (see Fig. 11(b)).

For each part of an object hypothesis, a visibility score is defined as the ratio between the length of the visible silhouette and the length of the whole silhouette. In the previous methods, Wu and Nevatia (2005) uses a constant elliptic mask to approximate the pedestrian shape, and define a 2-D region based visibility score by the ratio between the area of the visible regions to the area of the whole region; Lin et al. (2007) uses an edge template model to obtain a more accurate segmentation, from which a 2-D region based visibility score is computed. Compared to these 2-D region based visibility scores, the 1-D silhouette based visibility score is more accurate and meaningful for the shape based detectors. Figure 12 shows a comparison of different visibility scores on a toy example. Two humans are present in one image. A shorter one is in front; a taller one is in back. The size of the shorter one is 75% of the taller one. The y-coordinate of the shorter human's feet is one pixel larger than that of the taller human; the difference between their horizontal positions is measured by the pixel difference between their x-coordinates normalized by the pixel width of the taller human (the x-axis of Fig. 12(a)). It can be seen that while the 2-D region based visibility scores changes monotonically when the location difference decreases, the 1-D silhouette based visibility score fluctuates. However, when the occlusion is significant (see Fig. 12(b) for examples), the silhouette based score is usually larger than the region based scores. This retains the hypothesis of the taller human for further analysis.



(a) Visibility scores of the taller human in back

(b) Occlusion maps of two humans

Fig. 12 Comparison of 2-D region based and 1-D silhouette based visibility scores

6.3 Matching Object Edges with Visible Silhouettes

After obtaining the visible silhouettes, we assign the object edgelets extracted during part detection to the hypotheses by matching them with the visible silhouettes. For each edgelet, we find the closest silhouette to it and align the edgelet with the silhouette. Figure 13 gives the algorithm.

To assign edgelets to silhouettes, we first compute the distance transformation for each visible silhouette, and then compute the Chamfer matching scores between all the edgelets and all the silhouettes through distance transformation. One edgelet is assigned to the silhouette that has the highest matching score with it. If one edgelet has low scores with all the silhouettes, then it is not assigned to any. This procedure is similar to the traditional edge template matching. The difference is that we take silhouette points as edge points and edgelet features as edge template.

To align one edgelet with its corresponding silhouette, we first find the silhouette point **c** closest to the edgelet through distance transformation, and then search a small neighborhood of **c** along the silhouette, ±5 pixels. For each position, we cut a segment from the silhouette with the same length as the edgelet and compute its shape affinity to the edgelet by (4). The position with the highest affinity is taken as the aligned position, and the corresponding segment of the silhouette is marked as "supported" (see Fig. 11(c)). The ratio between the length of the supported segments and the overall length of the silhouette is called the *edge coverage* of the silhouette.

The above algorithm guarantees that one edgelet contributes to at most one hypothesis. If one silhouette can not get enough supporting edgelets (the edge coverage score is smaller than 0.25), the corresponding hypothesis will be removed. This solves the one-object-multiple-hypotheses problem in a natural way and prunes some false alarms. For example, the hypothesis 8 in Fig. 11(a) is removed in this way.

```
1. Compute distance transformation for all silhouettes;
2. For each object edgelet
   (a) Compute Chamfer matching scores to all the silhouettes, and assign the edgelet to the silhouette with the largest score;
   (b) Find the silhouette point c nearest to the edgelet and locally align the edgelet with the silhouette around c;
   (c) Mark the part of the silhouette that is covered by the edgelet as "supported";
```

Fig. 13 Matching and aligning edgelets with silhouettes

6.4 Matching Detection Responses with Visible Parts

For each hypothesis, we remove the body parts whose visibility scores are smaller than a threshold (0.7 in our experiments). The remaining parts are considered observable to the detectors. Matching part detection responses with the visible part hypotheses is a standard assignment problem, which we solve by the Hungarian algorithm (Kuhn 1955). For each response-hypothesis pair we compute their overlap ratio. If a pair's overlap ratio is larger than a threshold (0.5 in our experiments), it is considered a potential match. After matching, we apply under-merging to the remaining part responses to group the false alarms. Then we count the successful detections, false alarms, and missed detections, see Fig. 11(d) for an example.

6.5 Computing Joint Image Likelihood

Denote one visible part of an object hypothesis and one part detection response by $\mathbf{z}$ and $\mathbf{r}$ respectively; denote the set of matched response-hypothesis pairs by SD (successful detection). The sets of false alarms and missed detections are defined by $FA = \{\mathbf{r}|\mathbf{r} \notin SD\}$ and $FN = \{\mathbf{z}|\mathbf{z} \notin SD\}$ (false negative) respectively. Denote the object edgelets from the response $\mathbf{r}$ by $E(\mathbf{r})$. The joint image likelihood of multiple objects is defined by

$$P(O|Z) = \prod_{\{\mathbf{z},\mathbf{r}\}\in SD} P_{SD}(\mathbf{r}, E(\mathbf{r})|\mathbf{z}) \times \prod_{\mathbf{r}\in FA} P_{FA}(\mathbf{r}) \prod_{\mathbf{z}\in FN} P_{FN}(\mathbf{z}) \tag{15}$$

where O packs all observations, and Z for all hypotheses. The first term on the right side of (15) is the reward for successful detections. It is decomposed as

$$P_{SD}(\mathbf{r}, E(\mathbf{r})|\mathbf{z}) = P(\mathbf{r}|E(\mathbf{r}), \mathbf{z})P(E(\mathbf{r})|\mathbf{z}) \tag{16}$$

To model $P(\mathbf{r}|E(\mathbf{r}), \mathbf{z})$, we evaluate the distribution of the part detector's true positive rate under different edge coverage scores of the silhouette. The distribution is represented as a histogram. Spatial error between the response and the hypothesis or poor contract reduces the edge coverage score. Lower edge coverage usually corresponds to lower true positive rate. We assumes that $P(E(\mathbf{r})|\mathbf{z})$ is an uniform distribution, hence it is ignored in practice. The second term of the right side of (15) is the penalty for false alarms. It is computed by one minus the detector's precision. The third is the penalty for missed detection. It is computed by one minus the detection rate. These properties are evaluated for different part detectors independently during the off-line learning stage.

6.6 Searching for the Best Configuration

To search for the best interpretation of the image, we examine the initial object hypotheses one by one, in the descending order of their y-coordinates. See Fig. 14 for an example. If there are several hypotheses for one object, the algorithm will find the one that best aligns with the object edges and part responses. For example, the hypotheses h_1, h_3, h_4, h_5 in Fig. 14 correspond to one human. Our algorithm chooses the best one (h_1) and removes the others. If there are inter-object occlusions, the algorithm will ignore the occluded parts. For example, the legs of hypothesis h_{12} are not detected, but this can be explained by occlusion from h_7. Therefore, h_{12} is retained.

7 Experimental Results

We demonstrate our approach with the class of pedestrians. We first evaluate the performance of our boosted segmentor on un-occluded human samples, and then evaluate our joint analysis algorithm for multiple, partially occluded objects on two public image sets.

7.1 Training Part Detector Hierarchy

To train the part detectors, we collected about 5,000 pedestrian samples covering different viewpoints, and 7,000 background images without humans from the Internet. The full-body samples were resized to 24 × 58 pixels, and aligned according to the head and feet positions. Figure 15 shows some full-body training samples. The sizes of the other body parts were derived based on their definitions in Fig. 4.

For detection, the overall target false alarm rates of the classifiers for the nodes in the first two levels of the part hierarchy were set to 10^{-6}; the overall target false alarm rates for the bottom level nodes were set to 10^{-5}. With only inherited features, a detector can usually achieve a false alarm rate between 10^{-3} to 10^{-4}; by adding additional features, it achieves the target false alarm rate. Although feature sharing cuts training time by about a half, it requires about a week to train all the part detectors.

For the pedestrian class, we manually labeled the segmentation ground-truth for 1,800 samples. (Note that the algorithm in Fig. 9 does not require that all the positive samples have segmentation ground-truth.) We use a polygon to delineate the object, however, the boundary pixels are sometimes ambiguous and can not be classified clearly. Hence we mark a two pixel width do-not-care (DNC) boundary (see Fig. 15). The DNC pixels are ignored in both training and testing. This strategy is similar to that in Shotton et al. (2006). To evaluate segmentation, four fifths of the 1,800 segmented samples are used as training data, the rest one

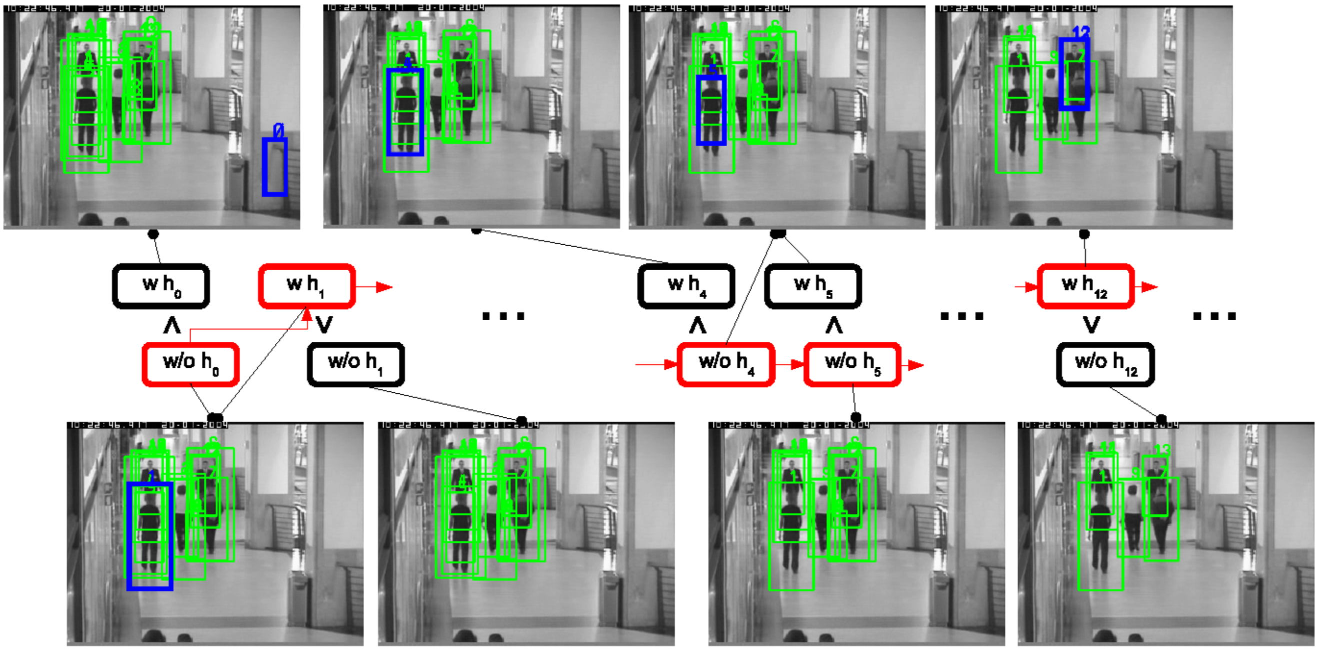


Fig. 14 (Color online) An example of searching for the best multi-object configuration. (The *blue rectangles* indicate the hypotheses being examined. The *red boxes* indicate the states kept after comparing the image likelihoods with/without one hypothesis. When examining a hypothesis, one of the "with" and "without" likelihoods can be inherited from the previous round to reduce computational cost. For example "without h_0" and "with h_1" are the same state, as h_0 is removed)

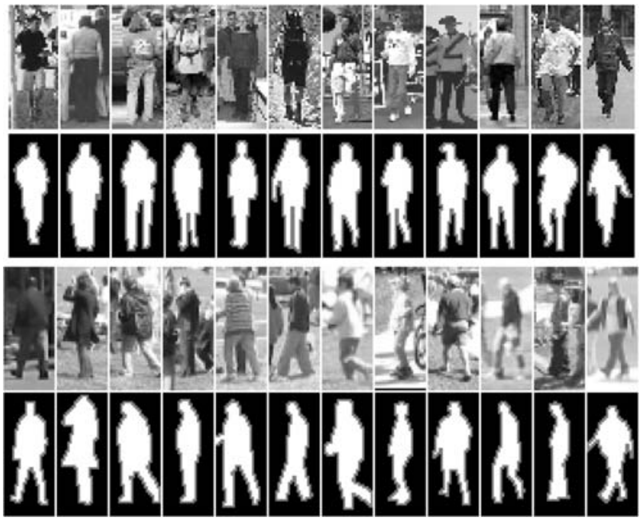


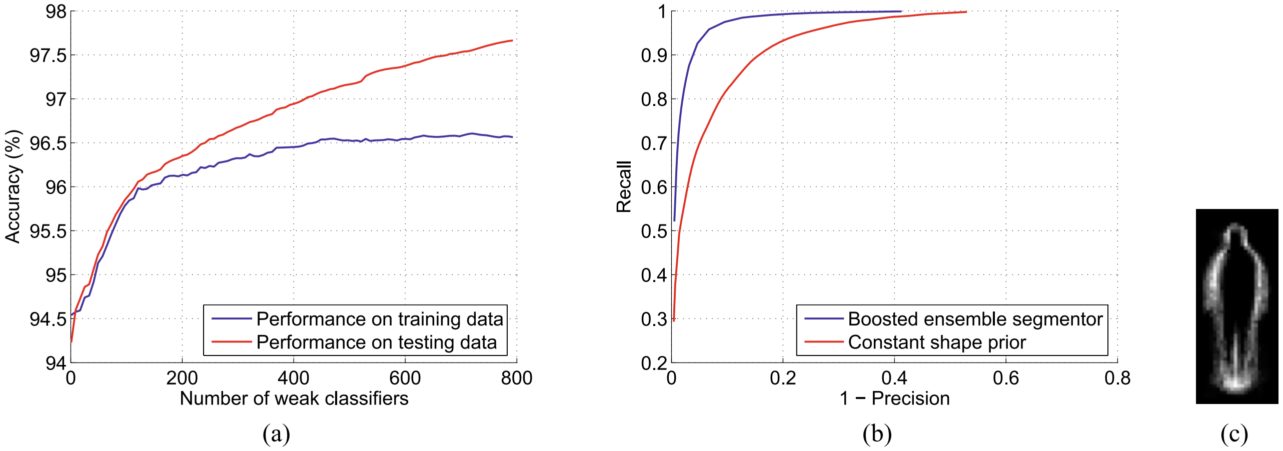
Fig. 15 Examples of pedestrian samples and their segmentation ground-truth. White pixels are for figure, *black* for background, and *gray* for do-not-care

fifth are used as testing data. Our pedestrian data set contains 4,640 training samples, 1,440 of which have segmentation ground-truth, and 360 test samples with segmentation ground-truth. For evaluation on case of multiple, occluded objects, we have additional test data that are fully independent from the training data.

7.2 Evaluation on Separate Objects

For the full-body node, we learn a segmentor in addition to the detector. Overall 800 edgelet features, one for each weak classifier, are selected by the boosting algorithm. We evaluate the segmentation performance with different numbers of weak classifiers. Figure 16(a) shows the segmentation accuracy on both the training data and the testing data. The segmentation accuracy is defined as the ratio of the number of pixels classified correctly to the number of all pixels. It can be seen that after about 500 weak classifiers, though the accuracy on the training data continues to increase, the accuracy on the testing data does not change much. However, no over-fitting is observed.

In addition to accuracy, we evaluate the segmentation *precision* and *recall rate* of the final segmentor. For segmentation, precision is defined as the ratio of the number of true object pixels that are classified as figure to the number of all pixels that are classed as figure; recall rate is defined as the ratio of the number of true object pixels that are classified as figure to the number of all true object pixels. Figure 16(b) shows the precision-recall (PR) curves on the testing data. For a comparison, we evaluate the segmentation performance of the constant shape prior, *i.e.* $h_0^{(s)}$ in Fig. 8. The equal-precision-recall rate of the boosted segmentor is about 94.6% on the testing data; while the equal-precision-recall rate of the shape prior is about 86.8%.



(a) (b) (c)

Fig. 16 Segmentation performance on separate humans: (**a**) segmentation accuracy with different numbers of weak classifiers; (**b**) segmentation precision and recall rate; (**c**) spatial distribution of segmentation errors. (*White* pixels are for higher error rate, and *black* for lower error rate)

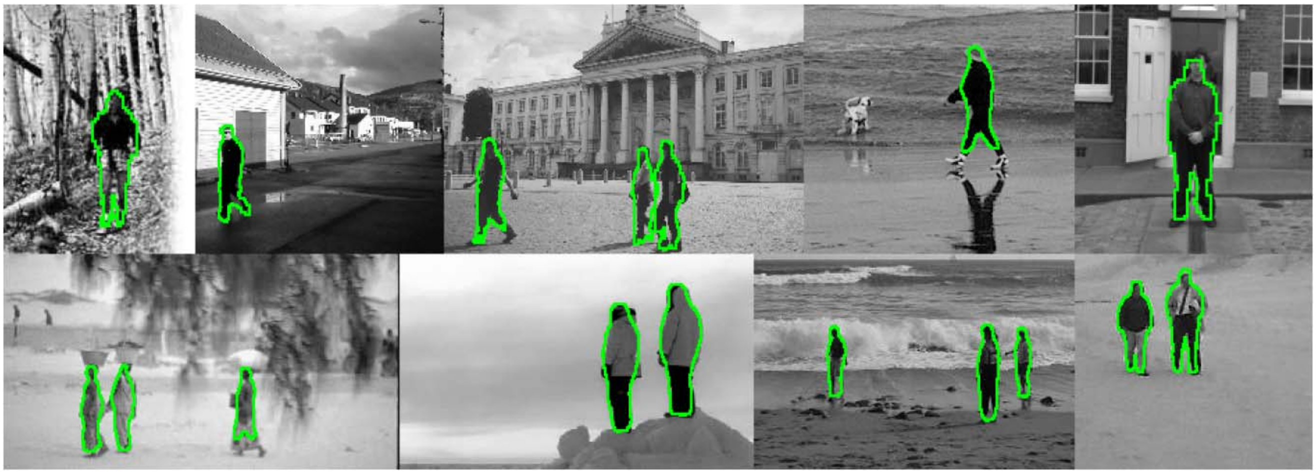


Fig. 17 Example detection and segmentation results on separate humans

Figure 16(c) shows the spatial distribution of the segmentation errors. For one pixel, the segmentation error rate is defined as the probability of its figure/ground label being classified incorrectly. It can be seen that most errors are around the object boundaries and the articulated parts, *e.g.* legs of humans. Figure 17 shows some example results on real images collected from the Internet. Because the humans are well separate in these examples, we only apply the full-body detector. It can be seen that our method can work reasonably well with different viewpoints and articulations.

Table 1 Performance of part detectors on the USC pedestrian set B. (The performance of right shoulder/arm/leg is similar to their left counterparts)

Part	Recall	Precision
Full-body	0.7638	0.9367
Head-shoulder	0.7269	0.9471
Torso	0.7934	0.9110
Legs	0.5720	0.8470
Head	0.6679	0.7702
Left shoulder	0.6863	0.8857
Left arm	0.7860	0.8694
Left leg	0.5240	0.8208
Feet	0.5092	0.7624

7.3 Evaluation on Partially Occluded Objects

We evaluated the whole multiple object detection and segmentation system on the "USC pedestrian set B" (Wu and Nevatia 2005)[1] and the "Zurich Mobile Pedestrian Sequences" (Ess et al. 2007).[2] Unlike the other popular test sets for pedestrian detection, *e.g.* the INRIA set (Dalal and Triggs 2005), the Daimler Chrysler set (Munder and Gavrila 2006), and the MIT set (Papageorgiou et al. 1998) which use normalized, separate human samples, these two sets con-

[1] See http://iris.usc.edu/~bowu/DatasetWebpage/dataset.html.

[2] See http://www.vision.ee.ethz.ch/~aess/iccv2007/.

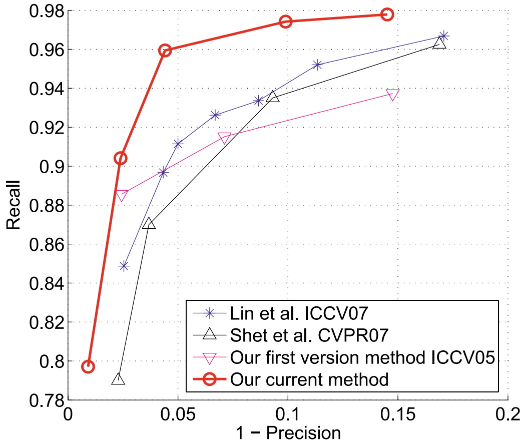


Fig. 18 Evaluation of detection performance on the USC pedestrian test set B

tain images with multiple interacting humans. They are very challenging because of the frequent occlusions. The detectors and segmentor used here are learned from the same training data described in Sect. 7.1, which is totally independent of the test sets.

7.3.1 *Results on the USC Test Set*

The USC pedestrian set B contains 54 images with 271 humans from the CAVIAR corpus.[3] In this set, 75 humans are partially occluded by others, and 18 humans are partially out of the scene. Table 1 lists the performance of our individual part detectors on this set. It can be seen that with occlusions,

[3] See http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

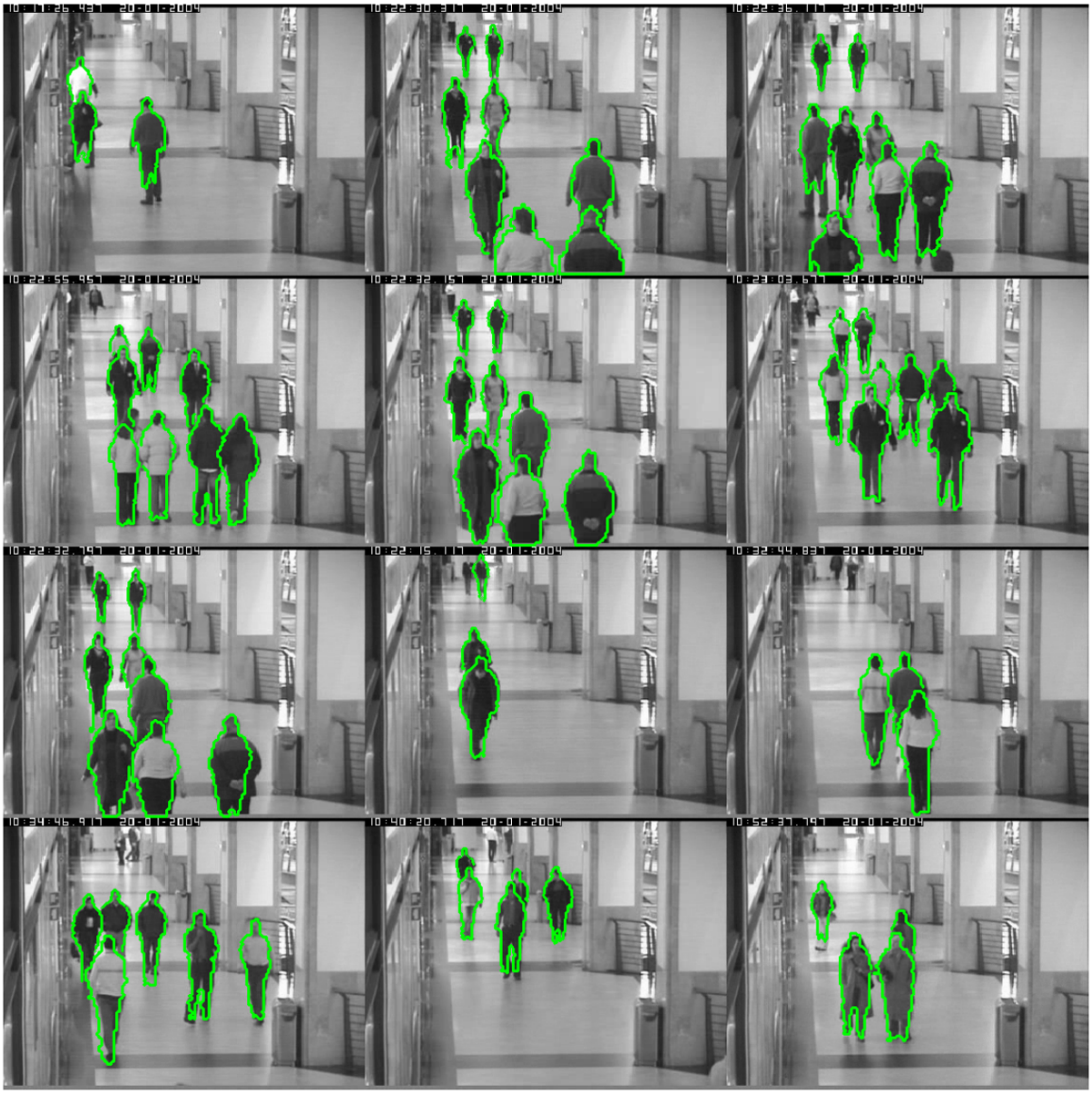


Fig. 19 Example detection and segmentation results on the USC pedestrian set B

the performance of the part detectors drop greatly. The feet and legs detectors have the poorest performance, as the occlusions usually happen to the lower body.

We compare the end-to-end performance of our system with some previous multiple human detection methods, including our previous system (Wu and Nevatia 2005, 2007c). Figure 18 shows the detection precision-recall curves. It can be seen that our method is significantly better than the other state-of-the-art methods, and all the combined detection methods are much better than any individual part detector on occluded examples.

Table 2 shows the detection rates on different degrees of occlusion. It can be seen that the detection rate on partially occluded humans is only slightly lower than the overall detection rate and declines slowly when the degree of occlusion increases.

To evaluate the pixel level segmentation performance, we manually label the segmentation ground-truth for this set. We first compute the segmentation precision and recall rate of every successfully detected human, and then compute the average scores. With a detection rate of 97.8%, the segmentation precision and recall rate are 83.18% and 84.78% respectively; with a detection rate of 96.0%, the segmentation precision and recall rate are 83.25% and 85.61% respectively. Note that we ignore the missed humans and false alarms when computing the scores for segmentation, because otherwise the segmentation scores would be dominated by the detection errors. The segmentation performance on the real images is not as good as that on the normalized samples, because the detection bounding boxes are usually not perfectly aligned with the objects.

In this experiment, we do not use any scene structure or background subtraction to facilitate detection. The test image size is 384 × 288 pixels. We search humans from 24 to 80 pixels wide. We use four threads to run detection of different parts simultaneously. Our experimental machine is a dual-core dual-processor Intel Xeon 3.0 GHz CPU. The average speed on this set is about 3.6 second per image. Figure 19 shows some example results on this set.

Table 2 Detection rates (%) on different degrees of occlusions. The detection rates of Shet et al. (2007), Wu and Nevatia (2005) are obtained with 19 false alarms; the detection rate of this method is obtained with 12 false alarms

Occlusion degree (%)	25 ∼ 50	50 ∼ 75	>75
Human number	34	31	10
Shet et al. (2007)	87	91.4	92.6
Our previous method (Wu and Nevatia 2005)	91.2	90.3	80
This method	94.12	93.55	90

7.3.2 *Results on the Zurich Test Set*

The Zurich set contains three test sequences captured by a stereo pair of cameras mounted on a children's stroller. Same as Ess et al. (2007), we only use the frames from the left camera for testing. The first test sequence contains 999 frames with 5,193 annotated humans; the second contains 450 frames with 2,359 humans; the third contains 354 frames with 1,828 humans. The frame size is 640 × 480 pixels. To compare with the results in Ess et al. (2007), which combines scene analysis with the object detection method in Leibe et al. (2005), we develop a simple method to estimate the ground plane, which is used to facilitate detection. We first use the full-body detector to search for humans from 58 to 483 pixel high. Then from the full-body responses, we do a RANSAC style algorithm to estimate a linear mapping from the 2-D image position to the 2-D human height: $ax + by + c = h$, where x, y are the image position, h is the human height, and a, b, c are the unknowns. With ground plane, the other part detectors only search the valid regions in the position-scale space. This saves some computational cost and reduces the false alarm rate.

Figure 20 shows the detection precision-recall curves of our methods and those in Ess et al. (2007). It can be seen that on all the three sequences our method dominates. Figure 20 also contains a curve of our method with only the full-body detector, which has similar performance compared to the method of Ess et al. (2007). This shows the necessity of part based system for such a complex scene. However, the efforts of this work and that in Ess et al. (2007) focus on different aspects. Ess et al. (2007) attempt to integrate scene struc-

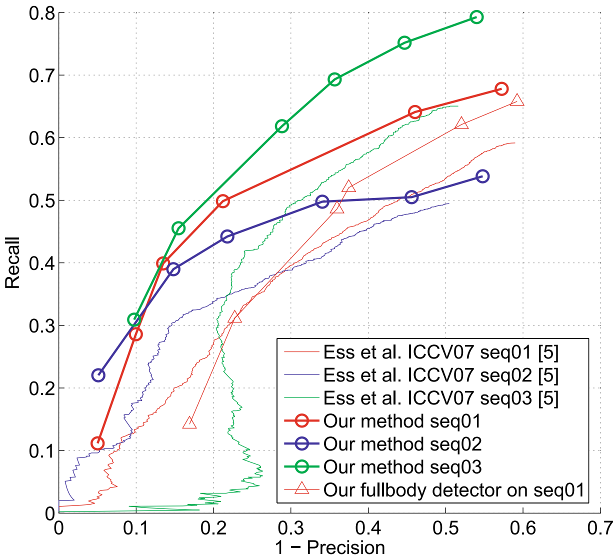


Fig. 20 Detection precision-recall curves on the Zurich mobile pedestrian sequences. (Following Ess et al. 2007's evaluation, only humans higher than 60 pixels are counted. The curves of Ess et al. 2007 are for their full-system, *i.e.* with ground plane and stereo depth)

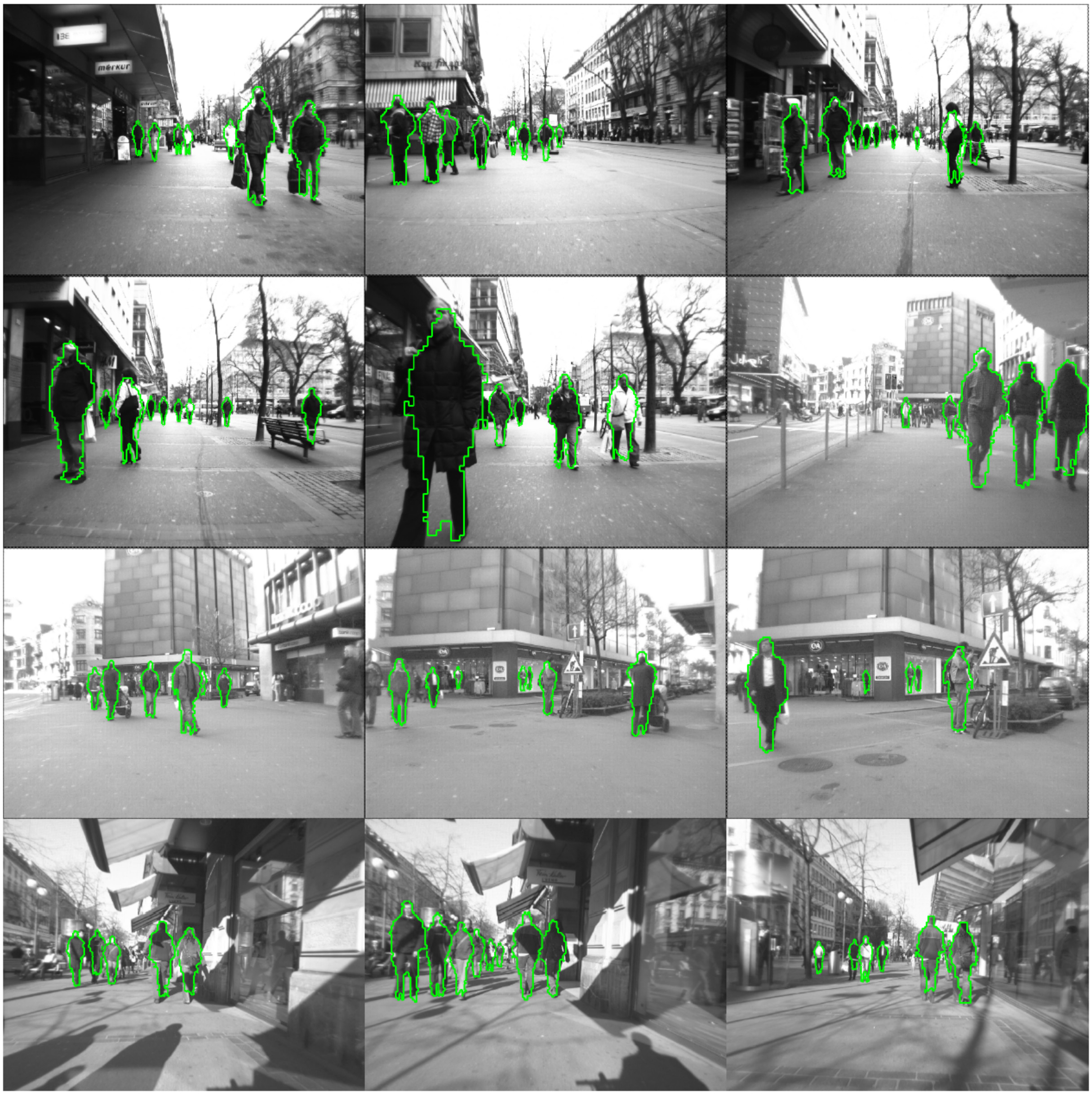


Fig. 21 Example detection and segmentation results on the Zurich mobile pedestrian sequences

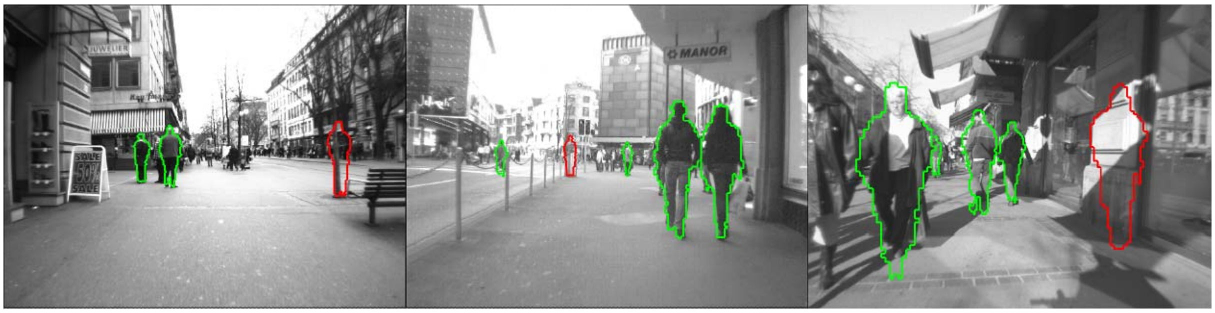


Fig. 22 (Color online)Examples of detection false alarm (*red* contours)

ture analysis and object detection, while our approach attempts to segment multiple, occluded objects jointly. These two complementary methods can be combined for further improvement. The average speed of our system on this set is about 2.5 second per image. Figure 21 shows some example results.

The performance on the USC set is much better than that on the Zurich set. This is mainly because the background of the Zurich set (outdoor) is much more cluttered than that of the USC set (indoor). At a similar detection rate, the false alarm rate is much higher on the Zurich set. Figure 22 shows some examples of false alarms. During testing, the only different parameter for the two sets is the search range of human sizes.

8 Conclusion and Discussion

We described a method to detect and segment multiple, possibly inter-occluded objects. Boosted classifiers are learned for the nodes in a part hierarchy. For the whole-object node, a segmentor is learned by boosting local shape feature based weak classifiers. For multiple object cases, occlusion reasoning is performed based on the object silhouette extracted from segmentation. A joint likelihood of multiple objects is maximized to find the best interpretation of the input image. We demonstrated our approach on the class of pedestrians. The experimental results show that our method outperforms the previous ones.

In this method, shape is used as the image cue for segmentation. In addition to shape, color and texture are also important cues for both segmentation and detection tasks. However, a holistic representation of color or texture model is sometimes uninformative and can not capture the details of the objects. Some part based representation could be helpful.

In our current method, although the detection and segmentation share the same set of image features, there is little interaction between these two modules. Intuitively, segmentation results could be used to verify the detection hypotheses. If the segmentor produces an unusual shape, it may suggest an error of the detector. There are existing methods exploring this direction (*e.g.* Zhao and Davis 2005).

To apply our approach to other object classes, some components need to be modified according to the class of interest. First, the design of the part hierarchy is class dependent. Different object classes require different partitions. Second, the ground plane assumption is valid for some objects in some applications, such as cars and pedestrians in surveillance videos, but not for all objects in all situations. When this assumption is not true, we need to infer the objects' relative depths by other techniques. Third, though the feature exclusiveness idea should be helpful for any feature based detection, it may require different implementations for different features.

References

Bray, M., Kohli, P., & Torr, P. (2006). POSECUT: Simultaneous segmentation and 3D pose estimation of humans using dynamic graph-cuts. In *ECCV*.

Chan, A. B., Liang, Z. S. J., & Vasconcelos, N. (2008). Privacy preserving crowd monitoring: counting people without people models or tracking. In *CVPR*.

Dalal, N., Triggs, B., & Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *ECCV*.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*.

Ess, A., Leibe, B., & Gool, L. V. (2007). Depth and appearance for mobile scene analysis. In *ICCV*.

Gavrila, D. M. (2000). Pedestrian detection from a moving vehicle. In *ECCV*.

Gavrila, D. M. (2007). A Bayesian, exemplar-based approach to hierarchical shape matching. *PAMI*, *29*(8), 1408–1421.

Gavrila, D. M., & Philomin, V. (1999). Real-time object detection for smart vehicles. In *ICCV*.

Huang, C., Ai, H., Li, Y., & Lao, S. (2005). Vector boosting for rotation invariant multi-view face detection. In *ICCV*.

Huang, C., Ai, H., Li, Y., & Lao, S. (2006). Learning sparse features in granular space for multi-view face detection. In *FG*.

Huang, C., Ai, H., Li, Y., & Lao, S. (2007). High performance rotation invariant multi-view face detection. *PAMI*, *29*(4), 671–686.

Kapoor, A., & Winn, J. (2006). Located hidden random fields: learning discriminative parts for object detection. In *ECCV*.

Kong, D., Gray, D., & Tao, H. (2006). A viewpoint invariant approach for crowd counting. In *ICPR*.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*, 83–87.

Leibe, B., Leonardis, A., & Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *Workshop on statistical learning in computer vision, in conjunction with ECCV*.

Leibe, B., Seemann, E., & Schiele, B. (2005). Pedestrian detection in crowded scenes. In *CVPR*.

Lin, Y.-Y., Liu, T.-L., & Fuh, C.-S. (2004). Fast object detection with occlusion. In *ECCV*.

Lin, Z., Davis, L. S., Doermann, D., & DeMenthon, D. (2007). Hierarchical part-template matching for human detection and segmentation. In *ICCV*.

Medioni, G., Lee, M. S., & Tang, C. K. (2000). *A computational framework for segmentation and grouping*. Amsterdam: Elsevier.

Mikolajczyk, C., Schmid, C., & Zisserman, A. (2004). Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*.

Mohan, A., Papageorgiou, C., & Poggio, T. (2001). Example-based object detection in images by components. *IEEE Transactions on PAMI*, *23*(4), 349–361.

Munder, S., & Gavrila, D. M. (2006). An experimental study on pedestrian classification. *IEEE Transactions on PAMI*, *28*(11), 1863–1868.

Mutch, J., & Lowe, D. (2006). Multiclass object recognition with sparse, localized features. In *CVPR*.

Opelt, A., Pinz, A., & Zisserman, A. (2006). A boundary-fragment-model for object detection. In *ECCV*.

Papageorgiou, C., Evgeniou, T., & Poggio, T. (1998). A trainable pedestrian detection system. In *Proceedings of intelligent vehicles*.

Pawan Kumar, M., Torr, P., & Zisserman, A. (2005). OBJ CUT. In *CVPR*.

Rowley, H., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *PAMI*, *20*(1), 23–38.

Sabzmeydani, P., & Mori, G. (2007). Detecting pedestrians by learning shapelet features. In *CVPR*.

Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, *37*, 297–336.

Schneiderman, H., & Kanade, T. (2000). A statistical method for 3D object detection applied to faces and cars. In *CVPR*.

Sharma, V., & Davis, J. W. (2007). Integrating appearance and motion cues for simultaneous detection and segmentation of pedestrians. In *ICCV*.

Shashua, A., Gdalyahu, Y., & Hayun, G. (2004). Pedestrian detection for driving assistance systems: single-frame classification and system level performance. In *IEEE intelligent vehicles symposium*.

Shet, V. D., Neumann, J., Ramesh, V., & Davis, L. S. (2007). Bilattice-based logical reasoning for human detection. In *CVPR*.

Shotton, J., Blake, A., & Cipolla, R. (2005). Contour-based learning for object detection. In *ICCV*.

Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2006). TextonBoost: joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*.

Todorovic, S., & Ahuja, N. (2006). Extracting subimages of an unknown category from a set of images. In *CVPR*.

Tu, Z., Zhu, S.-C., & Shum, H.-Y. (2001). Image segmentation by data driven Markov chain Monte Carlo. In *ICCV*.

Tuzel, O., Porikli, F., & Meer, P. (2007). Human detection via classification on Riemannian manifolds. In *CVPR*.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *CVPR*.

Viola, P., Jones, M., & Snow, D. (2003). Detecting pedestrians using patterns of motion and appearance. In *ICCV*.

Winn, J., & Jojic, N. (2005). LOCUS: Learning object class with unsupervised segmentation. In *ICCV*.

Winn, J., & Shotton, J. (2006). The layout consistent random field for recognition and segmentation partially occluded objects. In *CVPR*.

Wu, B., & Nevatia, R. (2005). Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. In *ICCV*.

Wu, B., & Nevatia, R. (2007a). Cluster boosted tree classifier for multi-view, multi-pose object detection. In *ICCV*.

Wu, B., & Nevatia, R. (2007b). Simultaneous object detection and segmentation by boosting local shape feature based classifier. In *CVPR*.

Wu, B., & Nevatia, R. (2007c). Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, *75*(2), 247–266.

Wu, B., Nevatia, R., & Li, Y. (2008). Segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. In *CVPR*.

Zhao, L., & Davis, L. (2005). Closely coupled object detection and segmentation. In *ICCV*.

Zhu, Q., Avidan, S., Yeh, M.-C., & Cheng, K.-T. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *CVPR*.