# Combined Top-Down/Bottom-Up Segmentation

Eran Borenstein and Shimon Ullman

**Abstract**—We construct an image segmentation scheme that combines top-down (TD) with bottom-up (BU) processing. In the proposed scheme, segmentation and recognition are intertwined rather than proceeding in a serial manner. The TD part applies stored knowledge about object shapes acquired through learning, whereas the BU part creates a hierarchy of segmented regions based on uniformity criteria. Beginning with unsegmented training examples of class and nonclass images, the algorithm constructs a bank of class-specific fragments and determines their figure-ground segmentation. This fragment bank is then used to segment novel images in a TD manner: The stored fragments are first used to recognize images containing class objects and then to create a complete cover that best approximates these objects. The resulting TD segmentation is then integrated with BU multiscale grouping to better delineate the object boundaries. Our experiments, applied to a large set of four classes (horses, pedestrians, cars, and faces), demonstrate segmentation results that surpass those achieved by previous TD or BU schemes. The main novel aspects of this work are the fragment learning phase, which efficiently learns the figure-ground labeling of segmentation fragments, even in training sets with high object and background variability, combining the resulting TD segmentation with BU criteria, and the use of segmentation to improve recognition.

**Index Terms**—Class-specific top-down segmentation, multiscale segmentation, learning to segment, combining top-down and bottom-up segmentation, object cover, fragment-based representation, combined segmentation and recognition.

## 1 Introduction

Understanding a visual scene requires the ability to recognize objects and their location in the image. These two goals—essentially the problems of recognition and segmentation—present considerable computational challenges.

The dominant approach to segmentation has been that of a bottom-up (BU) process, primarily involving the incoming image, without using stored object representations. The image is first segmented into regions that are relatively homogeneous in terms of color, texture, and other image-based criteria, and a recognition process is then used to group regions corresponding to a single, familiar, object. According to this approach, segmentation thus precedes and facilitates recognition.

Another approach to segmentation is that of a top-down (TD), high-level visual process, in which segmentation is primarily guided by stored object representations: The object is first recognized as belonging to a specific class and then segmented from its background using prior knowledge about its possible appearance and shape. In other words, according to this approach, recognition facilitates segmentation.

Recent state-of-the-art BU segmentation algorithms provide impressive results in the sense that they can be applied to any given image to detect image discontinuities that are potentially indicative of object boundaries. Their major difficulties, however, include the splitting of object regions and the merging of object parts with their background. These shortcomings are due to unavoidable ambiguities that cannot be solved without prior knowledge of the object class at hand, since most objects are nonhomogeneous in terms of color, texture, etc. Moreover, object parts do not necessarily contrast with their background, potentially causing the two to be merged.

TD segmentation uses prior knowledge of the object class at hand to resolve these BU ambiguities. However, it also has difficulties due primarily to the large variability of objects within a given class, which limits the ability of stored representations to account for the exact shape of novel images.

In this work, we introduce a segmentation scheme that addresses the above challenges by combining TD and BU processing to draw on their relative merits (Fig. 1). As discussed in Section 2.4, this also appears to be in closer agreement with human psychophysics and physiology. The TD part applies learned "building blocks" representing a class to derive a preliminary segmentation of novel images. This segmentation is then refined using multiscale hierarchical BU processing.

Our basic TD approach was introduced in [1], and later extended to include automatic learning from unsegmented images [2], as well as a preliminary scheme for combining BU processing [3]. The current version formulates the TD, as well as the combination components using a computationally efficient framework. It presents a fragment extraction stage that, unlike previous methods, produces a full cover of the object shape. This improvement is due to a modified mutual information criterion that measures information in terms of pixels, rather than images. This version also refines the automatic figure-ground labeling of the extracted fragments through an iterative procedure relying on TD/BU interactions. Another new aspect is the use of segmentation for improving recognition.

- *E. Borenstein is with the Division of Applied Mathematics, Brown University, 182 George Street, Providence, RI 02912. E-mail: eran_borenstein@brown.edu.*
- *S. Ullman is with the Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100 Israel. E-mail: shimon.ullman@weizmann.ac.il.*

*Manuscript received 16 Mar. 2007; revised 2 Oct. 2007; accepted 7 Nov. 2007; published online 17 Dec. 2007.*
*Recommended for acceptance by E. Saund.*
*For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-03-0169.*
*Digital Object Identifier no. 10.1109/TPAMI.2007.70840*

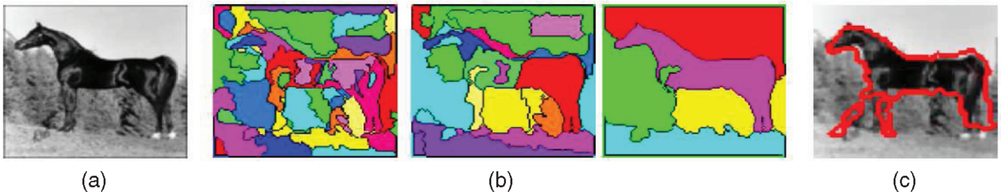


Fig. 1. Relative merits of TD and BU segmentation. (a) Input image. (b) BU (here at three scales) can be applied to any image to detect salient image discontinuities that potentially correspond to object boundaries. However, objects are usually segmented into multiple regions, some of which may merge with the background. (c) TD groups together dissimilar image regions corresponding to a single object and separates similar regions generated by different objects. However, it may not precisely follow image discontinuities.

The class-representation decomposes the global shape into smaller building blocks consisting of image fragments and their figure-ground labeling. As noted, these fragments produce a complete, rather than partial object cover, which enables better validation of the object detection and improved final segmentation. The fragments, as well as their figure-ground organization, are learned automatically. Each object part can be covered by a number of alternative fragments, which can also move with respect to each other, as long as they preserve figure-ground and spatial consistency. These features make the model effective in addressing large shape variability, as well as highly cluttered backgrounds.

In our combined approach, a central feature is the integration of multiscale information. Segments identified at multiple scales are grouped together or broken apart to form salient image regions (as guided by BU criteria) that are also consistent with the TD processing.

This paper is organized as follows: Section 2 describes related work. Section 3 presents the TD approach, which can be divided into two main stages: learning and segmenting. Section 4 combines the relative merits of TD and BU segmentation using an efficient message-passing algorithm. Section 5 describes experiments applied to a large set of complex images; and Section 6 presents the discussion and conclusions.

## 2 Related Work

### 2.1 Bottom-Up Approaches

BU segmentation emphasizes region-based properties such as the homogeneity of color (for example, [4]), intensity, or texture (for example, [5]), the smoothness and continuity of bounding contours (for example, [6] and [7]), or a combination of these region and boundary properties (for example, [8], [9], and [10]). In this work, we use the Segmentation by Weighted Aggregation (SWA) algorithm described in Section 4 to identify a hierarchy of homogenous image regions at multiple scales.

### 2.2 Top-Down Approaches

Directly related to our TD approach are *deformable templates* [11] that also apply a part-based representation model. The template parts are allowed to move with respect to each other, and also change their scale and orientation to some degree, to optimally fit the image content. In contrast to our method, the parts are usually simple geometrical elements (for example, ellipses, lines, and arcs); their spatial relationships, as well as their shape deformations are usually set manually; and their fitting often requires difficult optimization in a high-dimensional parameter space.

*Active contours or Snakes* [12], [13], [14], [15] constitute another group of nonparametric deformable models. In this case, an initial contour is formed on the image, guided by external (higher level shape information) and internal (for example, elasticity and smoothness) forces, such that the contour is eventually delineated by image discontinuities. As with deformable templates, the main difficulty of this approach is the attempt to fit an entire shape to the image, leading to difficulty in segmenting highly variable objects. In addition, these models require accurate initialization to compensate for a difficult minimization problem.

Leibe and Schiele [16] use a similar approach to our TD segmentation [1], applying a "codebook" consisting of image patches and their figure-ground organization to detect and segment pedestrians and other objects. The method is able to detect multiple overlapping objects using a minimal descriptor-length criterion. The POP model by Bernstein and Amit [17] is able to handle multiple objects and occlusions using a relatively small number of positive training examples. However, shape information is represented by a global template, and there is no sharing of object parts derived from different training examples.

### 2.3 Combined Segmentation

Related to our combined scheme is OBJ CUT by Kumar et al. [18]. The method automatically learns an object representation called Pictorial Structures (PS) from video sequences. The PS is combined with a contrast dependent Markov Random Field (MRF) that biases the segmentation to follow image boundaries. Winn and Jojic [19] use unsegmented images to learn a global figure/ground mask and a global edge mask that represent the "average" shape and edges of objects in the class. Shape and edge variations are constrained solely by a smoothness constraint. The global shape approach is limited in its ability to address rigid objects whose shape largely deviate from the "average." Additionally, the assumption of different object and background regions may be violated, especially in gray-level images.

The layout consistency criterion, suggested by Winn and Shotton [20] addresses occlusions and can identify their types (for example, self occlusion, background/object occlusion, etc.). However, it requires a manually segmented training set. It also assumes simple transformations that can align each object instance with a canonical grid. This assumption makes it hard to handle object classes with high shape variability. Yu et al. [21] combine a TD segmentation derived from a patch-based recognition system with a BU segmentation derived from normalized cuts. The object representation is set manually and the part configurations are restricted by spatial criterion alone. The representation used is limited in addressing highly variable objects.

Chen et al. [22] combine deformable templates and eigenfaces with a BU segmentation algorithm to detect and segment human faces and street signs in cluttered scenes. Liu and Sclaroff [23] also combine deformable templates with BU segmentation criteria. The image is first oversegmented, and then, various groupings and splitting are considered to best match a shape represented by a deformable template. Mori et al. [24] take a similar approach to segment baseball players from their background. A primary challenge facing the last two schemes is that of determining the appropriate level of initial segmentation. On one hand, dividing the image into a large number of small subregions reduces the likelihood of incorrectly merging figure regions with their background. On the other hand, increasing numbers of subregions makes their grouping more difficult; in the extreme, the grouping decision will be close to the problem of segmenting the initial image. This "optimal" balance may vary from image to image and even for different regions within an image. Ren et al. [25] use a MRF model, and Zhao and Davis [26] use a linear combination between a template matching and a color-based segmentation to detect and segment upper torsos. Weiss and Levin [27] propose a CRF scheme that learns to combine TD with BU cues from manually segmented images. The method produces only a partial TD object cover and then propagates the segmented figure using image similarity. The limited TD cover may miss object parts, and the image-based propagation does not effectively cope with highly variable object images. For example, when a white region may appear anywhere on a black horse, a partial cover followed by image-based propagation is likely to miss it. In contrast, our scheme produces a complete cover followed by a representation that takes into account regional image properties (such as color and texture) at multiple scales. The combination scheme is therefore able to address "dalmation" like objects, with body parts being detected based on their texture at coarser scales. A more detailed comparison with related work is given in Section 5.1.

### 2.4 Evidence from Human and Primate Vision

Psychophysical studies and physiological evidence from the primate visual system indicate that in human and primate vision, figure-ground segmentation, and object recognition proceed interactively and concurrently.

Peterson [28] showed that the time it takes subjects to organize an image into its figure-ground content depends on the orientation of the stimulus: segmentation occurs faster when objects are presented at familiar orientations, indicating that the differentiation of background from figure involves recognition. A study with camouflaged objects [29] used novel objects embedded in highly cluttered scenes. Subjects were initially unable to segment these scenes; yet, segmentation emerged spontaneously following an unsupervised learning experience, suggesting that the acquired object model guided the segmentation. Experiments with 4.5-month-old babies [30], [31] also indicate that figure-ground segmentation is influenced by prior familiarity. Zemel et al. [32] showed that in adult observers as well, spontaneous segmentation is determined by prior experience with specific shapes.

Evidence from neurophysiology shows contextual modulation of neurons, in which neurons at low-level visual areas (V1, V2) are influenced by higher level neurons, depending on figure-ground relationships [33], [34], [35], [36], [37]. In particular, many edge-specific units in low-level visual areas respond differently to the same edge, depending on the overall figure-ground relationships in the image.

## 3 Top-Down Segmentation

The full TD segmentation scheme consists of two main stages: learning and segmenting (Fig. 2, top). In the learning stage, a fragment bank is constructed automatically from a training set of unsegmented class and nonclass images. The bank consists of image fragments representing common object parts together with their figure-ground labeling. Unlike previous fragment or patch-based approaches [38], [39], the representation is designed to produce a complete covering of the object rather than a partial cover of its most discriminative parts. Section 3.1 summarizes how common image fragments are extracted from training images, and Section 3.2 describes how the fragment points are then labeled as figure or background. Subsets of these fragments are then used to optimally cover an object in a novel image (Section 3.3), and the figure-ground labeling of these covering fragments induces the final TD segmentation. The flowchart in Fig. 3 demonstrates the learning and segmentation algorithm.

### 3.1 Learning a Fragment-Based Class Representation

The learning stage begins with the construction of the fragment bank: a stored set of primitive shapes or "building blocks," for use in identifying regions containing common object parts (such as wheels, doors, and rooftops in cars). To find such components, we collect a large random (in terms of size and location) set of fragment candidates from the class images. A modified version of a *max-min* selection algorithm [40], [41] is then used to select a smaller subset of more informative fragments, based on the mutual information between the fragments and the class in question. High mutual information criteria is a good indication of the fragment being both selective for the class and general within it. These fragments are called *class-specific fragments*.

Let $f_i$ be a binary random variable representing the detection of a fragment $F_i$ in a given image ($f_i = 1$ if the fragment is detected, 0 otherwise); and let $C$ be a random variable representing the class of the image (C can have multiple values, for simplicity, we define it here as binary, $C = 1$ if the image belongs to a selected class). The mutual information between the fragment detection and the class is $I(f_i; C)$. Each candidate fragment $F_i$ is detected ($f_i = 1$) in an image if the similarity measure between the fragment and some image region $I_R$ exceeds a detection threshold $\Theta_i$. This threshold is set automatically to maximize the mutual information $I(f_i; C)$ measured in the training images. The similarity between a fragment and an image region is measured using *Normalized Correlation*, NC, denoted by $\rho(F_i, I_R)$:

$$\rho(x, y) = \frac{cov(x, y)}{\sqrt{var(x)var(y)}}. \tag{1}$$

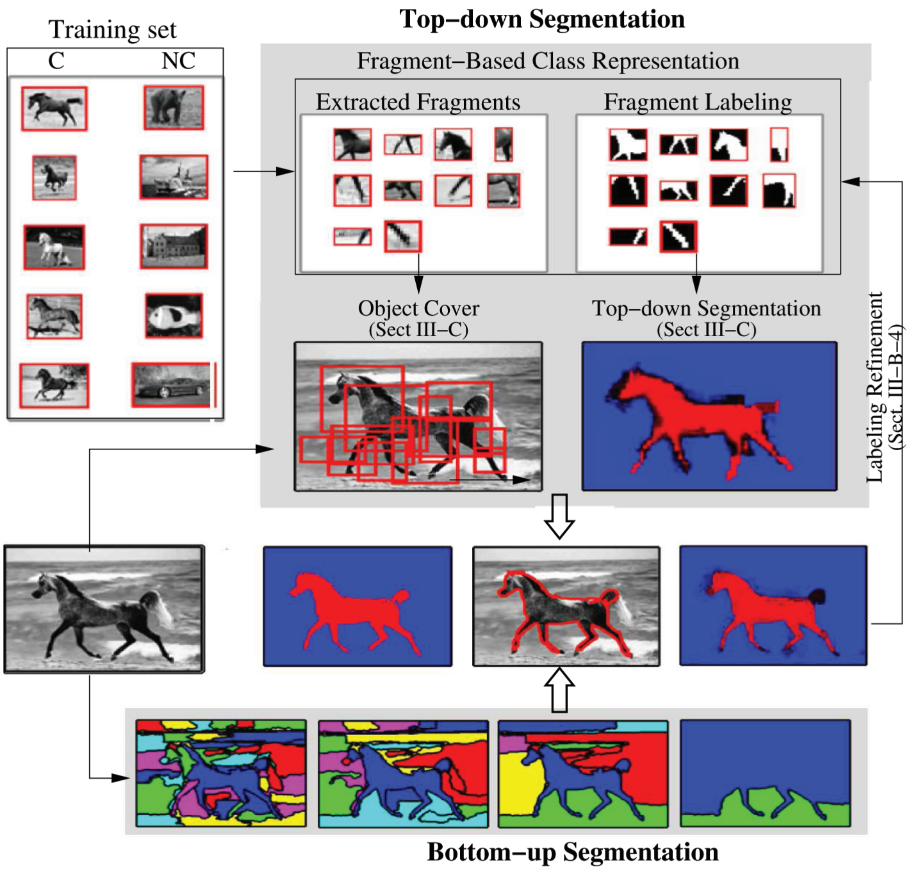


Fig. 2. Schematic diagram: The input to the learning stage consists of class (C) and nonclass (NC) unsegmentated images. Fragment extraction: Fragments containing common object parts (for example, horse head and leg) are extracted from the training images. Fragment labeling: The extracted fragments are segmented into figure and background parts. Object cover: Given a novel image, the extracted fragments are used to cover the object. The figure-ground labeling of these covering fragments then induces a TD segmentation. Combining the TD segmentation (top) with a BU process (bottom) provides the final segmentation (middle).

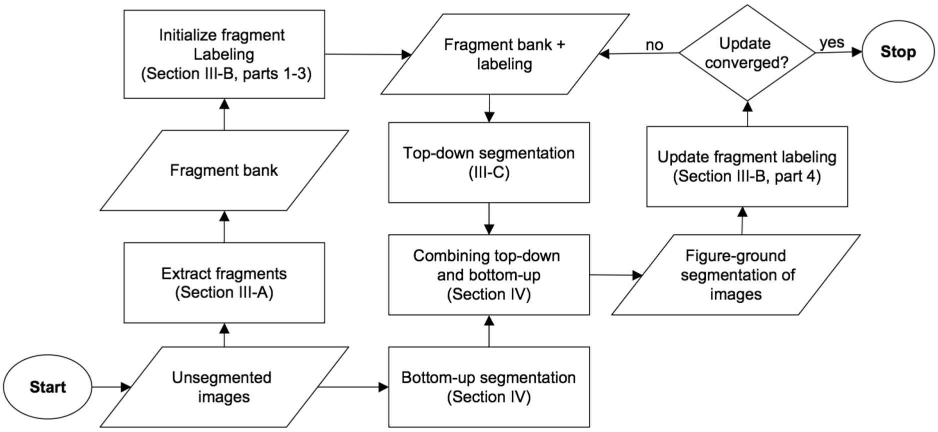

Initialize fragment Labeling (Section III-B, parts 1-3)
Fragment bank + labeling
no
Update converged?
yes
Stop
Fragment bank
Top-down segmentation (III-C)
Update fragment labeling (Section III-B, part 4)
Extract fragments (Section III-A)
Combining top-down and bottom-up (Section IV)
Figure-ground segmentation of images
Start
Unsegmented images
Bottom-up segmentation (Section IV)

Fig. 3. Algorithm flowchart: Rectangles represent processes and parallelograms represent their input and output. The learning process starts with a set of unsegmented images as input to extract a fragment bank. The figure-ground labeling of the extracted fragments is then estimated using the procedure described in Sections 3.2.1-3.2.3. The TD segmentation uses the labeled fragments to provide an object cover for each input image (Section 3.3). The TD segmentation is then combined with a BU process (Section 4) to derive the final segmentations. The resulting segmentations are then used to refine the figure-ground labeling of the extracted fragments (Section 3.2.4). If necessary, this process is repeated for further refinement.

Other similarity measures, that may be more invariant to viewing conditions, can be used instead of the NC measure [42], [43], [44].

Candidates $F_j$ are added one by one to the fragment set $F^n$ to maximize the gain in mutual information $I(F^n \cup F_j; C) - I(F^n; C)$:

$$F_j = \arg\max_F (I(F^n \cup F; C) - I(F^n; C)). \quad (2)$$

This expression is the information gained by adding a fragment $F$ to the set selected by the $n$th step. However, evaluating the last expression is impractical, since it requires estimating the joint probability $p(f_1, f_2, \ldots, f_n, C)$.

Therefore, the information gain is approximated using pairwise probabilities. A candidate is added to the set if the minimum information gain measured with respect to each one of the already selected fragments is maximal:

$$F_j = \arg\max_F (min_{F_i \in F^n} I(F_i, F; C) - I(F_i; C)). \tag{3}$$

The above expression requires estimating $p(f_i, f_j|C)$ rather than multivariate probabilities. This approximation provides a set of fragments with high mutual information and good classification performance [45], [41]. A detailed discussion on feature selection using mutual information measures can be found in [41]. Note that "antifragments," fragments likely to be detected in the background rather than in object images, also provide high information gain. We found that such antifragments tend to be unreliable, and their information gain is reduced as the number of nonclass examples increases. A simple filtering stage is therefore applied to remove these fragments from the candidate pool.

Unlike previous fragment-based schemes, our current goal is to use fragments not only to classify an image, but also to differentiate between its figure and background. This requires a fragment set large enough to cover the entire object, rather than its characteristic parts alone. The selected fragments must be well distributed over objects, thus effectively distinguishing between as many figure and background pixels as possible. The mutual information measure $I$ in (3) is therefore defined to measure the average contribution of individual pixels rather than images. For a single pixel $q$ sampled from the set of all training image pixels, define $C(q) = 1$ if $q$ is a figure pixel and $C(q) = 0$ if it is a background pixel. The probability $p(C = 1)$ is now defined as the probability that a pixel $q$ drawn at random from an input image is a figure pixel. In other words, $p(C = 1)$ is measured by the fraction of pixels rather than images in the training set for which $C = 1$. Similarly, for a fragment $F_j$, we define a random variable $f_j$, and $p(f_j = 1)$ is the probability that this fragment covers a pixel $q$ drawn at random from an image. To estimate $p(f_j, f_i, C)$, needed to evaluate (3), we count the number of pixels classified as $C(q) = 1$ or 0 (by the process described below) and also covered/uncovered by the first and second fragments $f_i$, $f_j = 0$ or 1 (producing eight different possibilities). The Max-Min selection (3) is then applied to the pixel-based mutual information criteria to derive the fragment bank: Candidates are added until the gain in mutual information becomes negligible. The bank selected in this manner provides an overcomplete class representation so that whole objects are likely to be covered by highly overlapping fragments. Note that every fragment can be used to cover at most one region per object. Our experiments obtained complete cover in most examples using a bank of $\approx 150$ fragments. The object covers are more efficient than those obtained using the previous selection algorithm (Results, Fig. 8). This difference was not the result of selecting larger fragments but was due to the use of the pixel-based MI, which favors a larger cover by the fragment set.

Initially, the objects in the training set are not segmented, and all pixels in the class images are therefore assigned with $C(q) = 1$, whereas pixels in the nonclass images are assigned with $C(q) = 0$. This assignment is subsequently refined during learning: The figure-ground labeling of the extracted fragments (Section 3.2) enables the figure-ground segmentation of objects in the training images (Section 3.3) and thus a refinement of the $C(q)$ labeling. The fragment extraction process can then be repeated for the new refined $C(q)$.

## 3.2 Automatic Figure-Ground Labeling

The second step in constructing the fragment bank is learning its *figure-ground labeling* $L$. The labeling determines every point within an individual fragment as either figure or background. (The fragment labeling can also take a continuous value between zero and one to represent the point's likelihood of being figure or background.) This fragment labeling is subsequently used to induce full TD segmentation of novel object images. Fig. 9 shows typical fragments extracted by our method, together with their figure-ground labeling, learned automatically from unsegmented training images.

The main idea for fragment segmentation is that of the consistency of figure versus background parts: To differentiate between these parts, we use the fact that a fragment's figure part covers regions containing features consistently repeated in class images, whereas its background part covers regions that may change significantly from one image to another.

To make use of this consistency, the labeling uses two related measurements: *degree of cover* and *border consistency*. The degree of cover measures the likelihood of each fragment point to be either figure or background, and the border consistency identifies fragment edges likely to separate figure from background parts. Empirical testing showed that the combination of these measures, described below, is sufficient for a reliable segmentation of the class fragments. Fig. 4 shows an overview of the fragment labeling initialization. Details of these measures and their use are explained next.

### *3.2.1 Degree of Cover*

When segmented images are available, the figure-ground likelihood of every fragment point $q$ can be calculated by counting the number of times the point covers regions segmented as figure versus background. However, at the beginning of the process, segmented images are not available, and this likelihood is therefore estimated using the *degree of cover*. The number of fragments covering (or overlapping on) a region in the image can serve to indicate whether this region belongs to the figure or background—for two reasons. First, as described, the fragment selection process provides a set of fragments that are more likely to be detected in class rather than nonclass images, and thus, figure regions are more likely than background regions to be covered. Second, the set of extracted fragments is sufficiently large to cover the object area several times (4.3 on average in our experiments) so that covering fragments highly overlap.

Therefore, a sample of $N$ training images is used to count the number of times each fragment point overlaps with other detected fragments:

$$D(q) = \frac{1}{Z_F} \sum_{q' \in Q(q)} d(q'), \tag{4}$$

where $Q(q)$ is the set of all the pixels $q'$ in the training images covered by fragment point $q$; $d(q')$ is the number of other fragments covering pixel $q'$; and $Z_F = \max_{q \in F} D(q)$ normalizes $D$ to satisfy $\max_{q \in F} D(q) = 1$

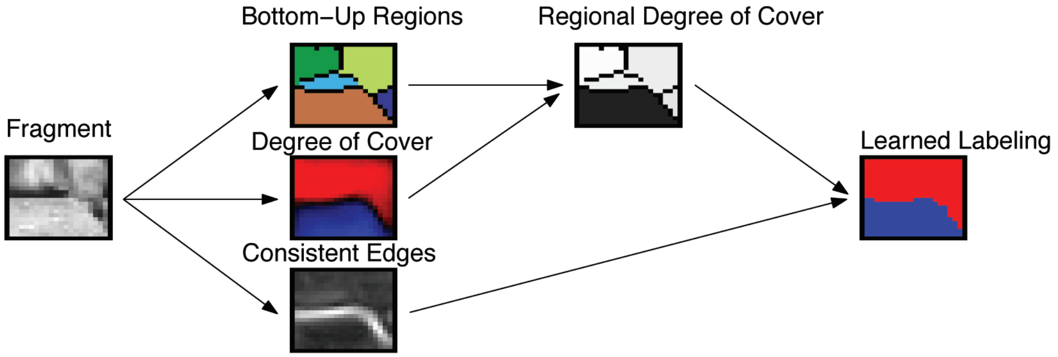


Fig. 4. Initializing the figure-ground labeling. The degree of cover ranks the figure likelihood of fragment regions, detected as homogenous by a BU process. This rank is combined with the fragment's consistent edges to determine the labeling. Note that the process can group together dissimilar subregions and that the consistent edges detect missing and low-contrast parts of the figure-ground borders.

(giving $D(q) \in [0,1]$). When a fragment never overlaps with other fragments, we get $Z_F = 0$. In this case, the fragment is identified as a background fragment and is removed from the fragment bank. Training images in which objects occupy at least 30 percent of the entire image were used to reduce the number of such background fragments and, hence, reduce computational cost. Using such images for training is not critical: Background fragments provide little information about the class and are therefore unlikely to be selected to the fragment bank. Additionally, the figure-ground labeling of those selected erroneously is likely to be inconsistent with the labeling of overlapping fragments.

The degree of cover, defined for pixels, is then averaged within homogeneous subregions contained in the fragment. This averaging makes use of the fact that points belonging to a homogenous (for example, sharing similar gray levels) subregion $R_i$ within the fragment are likely to have the same figure-ground labeling. Therefore, if $q_1$, $q_2 \in R_j$ are in the same region $R_j$, then they share the same average degree of cover: $\bar{D}(q_1) = \bar{D}(q_2)$. Averaging within a local uniform region will therefore increase the measure's signal-to-noise ratio. A simple BU process described in [46] is used to determine these subregions[1] and then use their regional degree of cover to impose a homogeneity constraint on their labeling and reduce computational cost.

The fragment labeling $L$ can consequently be determined by selecting a single threshold $\theta_d$ on these regional degrees to obtain:

$$L^{figure}(\theta_d) = \{q : \bar{D}(q) \geq \theta_d\}. \tag{5}$$

This means that all fragment points $q$ contained in the figure part $L^{figure}(\theta_d)$ have a regional degree of cover $\bar{D}(q)$ higher or equal to $\theta_d$, whereas all other points are chosen as the background part. The number of possible segmentations of a fragment $F$ is reduced in this manner from $2^n$ to $n$, where $n$ is the number of BU regions $R_i$ in $F$. Section 3.2.3 describes how to determine this threshold $\theta_d$.

This degree of cover alone already provides a highly informative cue for the labeling process. In our experiments, the average degree of cover for all pixels labeled manually as figure was 0.8 with $(\text{standard deviation} = 0.17)$ compared with 0.31 $(\text{standard deviation} = 0.2)$ for all pixels labeled manually as background.

### 3.2.2 Border Consistency

We next determine a boundary in the fragment that optimally separates figure from background regions. This is obtained by combining a measure of edge consistency with the degree of cover. Image patches in which a fragment is detected are collected from the training images. Denote this collection by $H_1, H_2, \ldots, H_k$. Each patch in this collection, $H_j$, is called a *fragment hit*, and $H_j(q)$ denotes the gray-level value of the hit pixel corresponding to the fragment point $q$. In each one of these hits, we apply an edge detector. The class-specific edges will be consistently present among hits, whereas other edges are arbitrary and change from one hit to another. The fragment's consistent edges are learned by averaging the edges detected in these hits (Fig. 5). Pixels residing on or close to consistent edges will get a higher average value than pixels residing on noise or background edges, as defined by

$$E(q) = \frac{1}{k}\sum_{j=1}^{k} \|\nabla H_j(q)\|^2. \tag{6}$$

The gradient operator can be replaced by other edge detectors. By the end of this process, $E(q)$ is normalized with a linear transformation so that $E(q) \in [0,1]$ with $E(q) = 1$ for the most consistent edge point and $E(q) = 0$ for the least consistent point.

There are three different types of edges. The first is the *border edge*, which separates the fragment's figure part from its background and is found within the fragment unless the fragment is contained entirely within the object. The second, the *interior edge*, lies within the figure part of the object. For instance, a human eye fragment contains interior edges at the pupil or eyebrow boundaries. The last type, a *noise edge*, is arbitrary and can appear anywhere in the fragment hit; it usually results from background texture or from artifacts coming from the edge detector. The first two types of edges are the consistent edges that consistently appear within the fragment hits. These consistent edges are used to segment the fragments, as next described. Consistent edges were found to be a better cue than consistent gray levels: With this measure, background regions are unlikely to contribute consistency even when the same type of background is

[1] These are determined by selecting a scale in the multiscale hierarchy. In our experiments, scales that gave on average 3-14 homogenous subregions produced almost identical results.

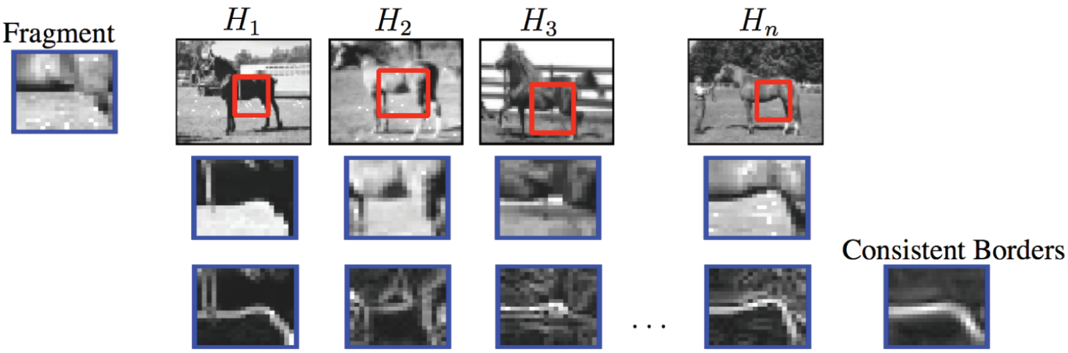


Fig. 5. Learning consistent edges: Fragment (top left) and the consistent boundary extracted from it (bottom right). To detect the consistent boundary, fragment hits $(H_1, \ldots H_n)$ are extracted from the training images (top row shows the hit location inside a red rectangle, middle row shows the hits themselves). An edge detector is used to detect the edge map of these hits (bottom row). The average of these edge maps gives the consistent edge (bottom right).

present (for example, grass in horse images). Note, however, that background edges, which are consistently aligned with the object parts, would be learned as object parts. For example, in our experiments, shadows appearing consistently below cars in the training images were learned as an object part. A more variable database is needed to prevent such cases.

### 3.2.3 Initial Figure-Ground Labeling

Having determined a fragment's degree of cover $D$ (4) and border consistency $E$ (5), these are then combined to determine its figure-ground labeling. The goal is to find a figure part $L^{figure} = \{q|L(q) = 1\}$ and a background part $L^{ground}$ such that the degree of cover measured within the figure part is higher than the degree measured within the background part, whereas the boundary between them is supported by consistent edges. The following labeling score expresses this objective:

$$f_l(L) = f_{deg}(\bar{D}, L) + \lambda f_{cons}(E, L), \tag{7}$$

with $\lambda$ determining the relative contribution of the two criteria. To compensate for the different units ($f_{deg}$ measures area, whereas $f_{cons}$ measures length), we used $\lambda = 15$ pixels, which is the ratio between the average area of the fragments and their average contour length. The degree of cover term $f_{deg}(\bar{D}, L)$ is minimized when points labeled as figure $(L(q) = 1)$ have a high degree of cover, and background points $(L(q) = 0)$ have a low degree:

$$f_{deg}(\bar{D}, L) = \sum_q (L(q) - \bar{D}(q))^2. \tag{8}$$

The border consistency term $f_{cons}(E, L)$ is minimized when the boundary between the figure and background parts is supported by consistent edges (pixels with high $E(q)$):

$$f_{cons}(E, L) = -\sum_{q,p \in N_q} (L(q) - L(p))^2 (E(q) + E(p)). \tag{9}$$

The $N_q$ set denotes a neighborhood of pixels adjacent to $q$ (we use 4-pixel neighborhoods). The above term decreases as the edge consistency of boundary pixels $(L(q) \neq L(p))$ increases.

Finding a fragment labeling that minimizes this score is straightforward. The search is constrained to labeling that preserve the degree of cover: All regions labeled as figure must have a degree of cover higher than regions labeled as background. It is therefore sufficient to check which of the $n$ such labeling (5) minimizes (7). This procedure produces good initial labeling (Section 5), which is further improved through an iterative process, as next explained.

### 3.2.4 Final (Iterative) Figure-Ground Labeling

The fragment's figure-ground labeling is now improved through an iterative refinement process. Complete objects are segmented using the labeling of individual fragments (Section 3.3). In turn, this segmented set is used to improve the labeling of the individual fragments: The figure-ground labeling of every pixel $q'$ in the training set provides a better figure-ground labeling likelihood than its degree of cover defined in (4). Instead of counting the average number of times a fragment point overlaps with other detected fragments, we count the number of times it covers pixels segmented as figure versus background:

$$D(q) = \frac{1}{|Q(q)|} \sum_{q' \in Q(q)} X^0(q'), \tag{10}$$

where $X^0(q')$ is the figure-ground labeling of pixel $q'$ (Section 4). This refined likelihood is then used to update the fragment's figure-ground labeling, as described in Section 3.2.3. With this increasing accuracy, we can use the pixels' likelihood $D(q)$ rather than their regional likelihood $\bar{D}(q)$ to define the fragment labeling (when the regions become pixels we have $D(q) = \bar{D}(q)$). Note that the TD segmentation is determined by the consistent labeling of overlapping fragments (Section 3.3), and therefore, when a significant fraction of these fragments is correctly labeled, very few image points are likely to be labeled incorrectly. In this case, the iterative process is unlikely to change correct labeling and likely to change incorrect labeling (inconsistent with the majority). Indeed, more than 80 percent of the fragment points are already labeled correctly at the first iteration (Section 5) resulting in a stable and fast refinement within two iterations. The figure-ground labeling of the fragments is then used to segment new images, as next described.

### 3.3 Segmentation by Optimal Cover

The main stage of the TD segmentation consists of covering an object in the image with a subset of the stored fragments. The figure-ground labeling of these fragments then identifies the object's figure-ground segmentation. An object cover should optimize three main criteria: *individual match*, *figure-ground consistency*, and *spatial consistency*. Individual match requires that covering fragments are similar to the image regions they cover. The figure-ground consistency requires consistency between the figure-ground labeling of overlapping fragments and the spatial consistency limits the fragments' position to maintain spatial relationships. This section describes a cover score that represents these criteria followed by an optimization technique that determines an optimal cover with respect to this score. Finding a cover optimizing this score is difficult and is therefore done in two main stages. The first stage quickly determines Regions Of Interest (ROIs) that specify locations within the image likely to contain objects of interest (from classes represented in the system) together with their initial partial cover. The partial covers are then completed and refined in the second stage.

An object ROI is represented by specifying a coordinate system $\vec{pos}$ (position, orientation, and scale), and an object cover is represented by a vector $\vec{f}$ in which $f_i = q$ if the center of the fragment $F_i$ covers image point $q$ and $f_i = 0$ if it is not used for the cover (A fragment can therefore be used at most once for each cover.). A cover score $E(\vec{f}, \vec{pos})$ is assigned for every ROI $\vec{pos}$ and object cover $\vec{f}$ in a given image $y$. As noted above, this score takes into account three criteria: *individual match*, *figure-ground consistency*, and a *spatial constraint*. The individual match measures the individual contribution of each covering fragment. It is proportional to the similarity between the fragment and the image region it covers. The figure-ground consistency criterion requires pairs of fragments to consistently overlap in terms of their figure-ground labeling. The spatial constraint limits the fragments to cover only specific subregions by specifying for each fragment a corresponding receptive field $RF_i(\vec{pos})$ within the ROI. In other words, a receptive field $RF_i$ is a set specifying valid values for $f_i$. This set is represented by a center and a radius parameters. Only positions $f_i$ whose distance to the receptive field's center is less than the radius are valid. A method for learning these receptive field parameters is described later.

The overall score for a cover $\vec{f}$ given an image $y$ is therefore expressed as

$$(\vec{f}, \vec{pos}) = \sum_i E_{ind}(f_i) + \lambda \sum_{i,j} E_{cons}(f_i, f_j) \tag{11}$$

subject to

$$f_i \in RF_i, \text{ or } f_i = 0, \ \forall i.$$

The $\lambda$ coefficient determines the relative contribution of the figure-ground consistency $E_{cons}$ relative to the individual contributions $E_{ind}$ of the covering fragment. We found that both individual similarity and fragments consistency play a useful role in evaluating the quality of a cover. The specific form of each factor was optimized empirically.

The first term $E_{ind}$ in (11) is given by the individual contribution of a covering fragment:

$$E_{ind}(f_i) = \frac{|F_i|}{|RF_i|} \rho_i(f_i), \tag{12}$$

which is proportional to the number of pixels $|F_i|$ covered by the fragment and the similarity $\rho_i(f_i)$ between the fragment and the region it is covering (1). It is also inversely proportional to the spatial uncertainty of the fragment position, as determined by the size of its receptive field $RF_i$: Fragments with larger receptive fields $RF_i$ contribute larger spatial uncertainty.

The figure-ground consistency term $E_{cons}$ is determined by the pairwise contribution of overlapping fragments:

$$E_{cons}(f_i, f_j) = \frac{1}{2} \sum_{i,j} \{E_{ind}(f_i) + E_{ind}(f_j)\} C_{ij}(f_i, f_j), \tag{13}$$

where the consistency measure $C_{ij}(f_i, f_j)$ above is proportional to the difference between the number of pixels labeled consistently and inconsistently by the pair $(F_i, F_j)$:

$$C_{ij}(f_i, f_j) = \frac{\#\text{ consistent} - \#\text{ inconsistent}}{\#\text{ consistent} + \#\text{ inconsistent}}. \tag{14}$$

This measure is 1 when there is a perfect consistency between the fragments, $-1$ when they are totally inconsistent and zero when the fragments do not overlap. A fragment contribution to the score may be negative when its overall consistency with other overlapping fragments is negative. In this case, removing or moving this fragment to cover a different region increases the score.

The spatial constraints $RF_i(\vec{pos})$ are specified with respect to the object ROI $\vec{pos}$. Therefore, maximizing the score in (11) requires identifying an object ROI $\vec{pos}$ together with an object cover $\vec{f}$. We found that this maximization problem can be approached efficiently by dividing the cover construction into two successive stages: detection and segmentation (Fig. 6). The detection stage aims to maximize the first part of the covering score $E_{ind}$, thus providing an ROI that is likely to contain the object together with a set of covering fragments. The decomposition of the $E_{ind}$ term into a factor tree (each $E_{ind}$ term depends on the ROI parameter $\vec{pos}$ and exactly one fragment variable $f_i$) makes it possible to find its exact global maximum (with respect to ROI and fragments variables), using a max-sum algorithm [47] (see also Section 4).

This stage is similar to other recognition schemes [48], [49], [50], [39] that use informative fragments or features for object detection. Similar to previous schemes, we found that this stage is sufficient for initial detection: When the maximum obtained is sufficiently large, we assume object detection and proceed to the second stage, namely, segmentation.

The segmentation stage aims to maximize the full cover score (11), taking into account also the figure-ground consistency part $E_{cons}$. A simple relaxation-like technique is applied, starting with the ROI and the initial cover found in the first step. The variables $\vec{f}$ and $\vec{pos}$ are tested successively and changed to maximize the cover score, whereas all other variables are held fixed at their current values. Let $\vec{f}^n$ and $\vec{pos}\ n$ be a realization of the fragment and ROI variables at the $n$th iteration. The state of every fragment is hence changed (one at a time) to increase the covering score:

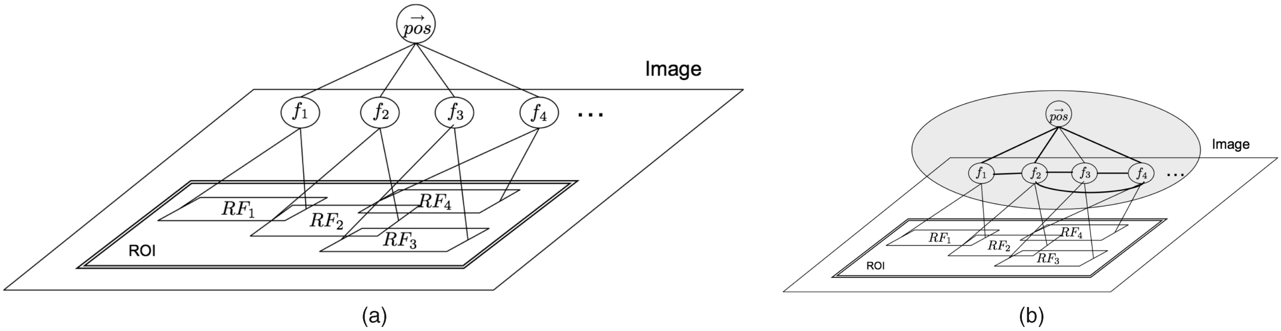


Fig. 6. Cover construction is divided into two successive stages: (a) Object detection: partial object detection is obtained by maximizing (11) using its factorization into a factor tree. A sufficiently high maximum provides an initial estimate for an ROI that is likely to contain the object, as well as a set of covering fragments. Each covering fragment $F_i$ is limited to cover pixels within its receptive fields $RF_i$ as designated by the ROI (spatial constraint). (b) Segmentation: The complete object cover and segmentation is provided through the maximization of the whole cover score (11). Note that the consistency term between overlapping fragments introduces lateral connections between fragment nodes, which makes maximization hard. A relaxation like technique is therefore used to find a local maximum using the initial cover found at stage (a).

$$f_i^{n+1} = \arg\max_{f_i} E(\vec{f}^n, \vec{pos}\ n). \tag{15}$$

This change depends on the states of the other fragments and the ROI. It will add the fragment to the cover (zero to nonzero state) or remove (nonzero to zero) it from the cover or change its cover position (nonzero to nonzero) to improve its consistency with overlapping fragments.

Similarly, the value of the ROI variable is updated by

$$\vec{pos}\ n+1 = \arg\max_{\vec{pos}} E(\vec{f}^n, \vec{pos}). \tag{16}$$

Since the score is increased at each step, and the maximal score is bounded, the cover construction is guaranteed to converge, terminating at some local maximum. In our experiments, the convergence was quick, typically after 3-4 sweep iterations (updating the value of each variable 3-4 times).

The fragment receptive field parameters (position within the ROI and radius) are learned during training in a similar iterative manner. The receptive field position of a fragment is initialized by the position within the source image it was extracted from, and the radius is initialized to 7 pixels. After this initialization, the positions and radii of all fragments are iteratively refined to maximize (11), just like in (15) and (16), whereas the $\vec{pos}$ and $\vec{f}$ parameters are held constant.

The consistency term added to (12) also improves object recognition, since it is easier to produce a consistent cover inside ROIs containing an object than in those that do not. To demonstrate this, we compared the classification performance of the segmentation score with and without the consistency part $E^c$. The ROC curves showed significantly improved performance when the figure-ground consistency was taken into account (Section 5 and Fig. 13).

Once an ROI and object cover are determined, the TD segmentation $S(\vec{f})$ of the detected object is derived from the figure-ground labeling of the covering fragments: Each fragment applies its labeling to the pixels it covers, and the final segmentation $S_q(\vec{f})$ of each pixel $q$ is given by the fraction of covering fragments that label it as figure:

$$S_q = \frac{1}{|A(q)|} \sum_{i \in A(q)} L_i(q), \tag{17}$$

where $A(q)$ is the set of all fragments covering pixel $q$, and $L_i(q)$ is the labeling given by fragment $F_i$ to $q$.

## 4 Combining Top-Down and Bottom-Up Segmentation

This section describes a method for combining the TD segmentation with BU processing to draw on their relative merits. The TD information is used to group segments belonging to an object despite possible BU dissimilarity and to break apart salient BU segments containing both figure and background. The BU process precisely delineates the object's boundaries in the image.

The BU process is based on the SWA algorithm [46], [51], which detects a hierarchy of homogenous regions at multiple scales. The hierarchical nature of the BU process is a key aspect of the combined segmentation. Some object parts, like a horse ear, may be detected at fine scales and disappear at coarser scales, whereas other parts, like the horse back, can only be detected at coarser scales. The SWA algorithm used is briefly described below, for details, see [46]. The segmentation hierarchy is identified by a recursive coarsening process, in which homogeneous regions called "segments" at a given level are used to form larger homogeneous segments at the next level. In this manner, the image is segmented into fewer and fewer segments, producing a hierarchical graph $G(V,E)$, in which each segment $V_i^l$ at a level $l$ is connected with a relating weight $E_{ij}^l$ to another segment $V_j^{l+1}$ at a coarser level $l+1$, providing that the first was one of the segments used to define the latter (Fig. 7). The weight of an edge connecting two segments represents their similarity, taking into account texture, average intensity, and boundary properties. This connection weight increases as the similarity increases and is normalized such that $\sum_j E_{ij}^l = 1$. Each segment $V$ is connected to a subset of the images pixels. These connections are recursively determined by

$$V_i^l(q) = \sum_j E_{ji}^{l-1} V_j^{l-1}(q). \tag{18}$$

A segment $V_q^0(q)$ at the terminal level $l = 0$ is connected to a single pixel $i = q$. Note that at any level $l > 0$, a pixel may be connected with different weights to multiple parent segments, providing a soft segmentation of the image. To

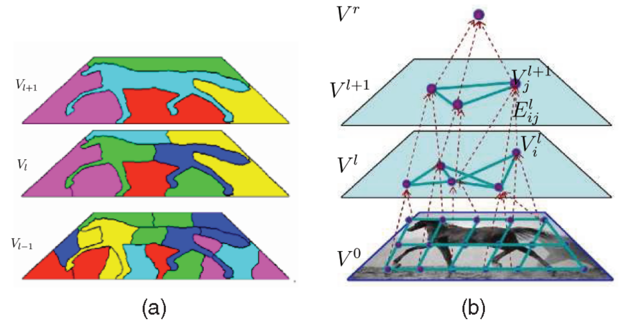


Fig. 7. Hierarchical BU segmentation of the image. Segments at multiple scales, identifying salient homogenous regions (a), are represented by a graph $G = (V, E)$ (b). The graph is constructed recursively, starting at the pixel level $l = 0$: segments $V^l$ at a given level $l$ are used to form larger segments $V^{l+1}$ at the next level, as indicated by their connecting edges $E$.

increase the computation efficiency in the combined segmentation, a segmentation tree $T$ is extracted from the graph, where every segment $V_i^l$ has only a single parent $V_j^{l+1} = par(V_i^l)$ at the next coarser level. A segment's parent is chosen as that having the highest degree of overlap with it ($|V_i^l \cap par(V_i^l)|$ is maximal).

The combined segmentation algorithm proceeds by labeling each segment in the segmentation tree as either "figure" ($X_i^l = 1$) or "background," ($X_i^l = 0$). The labeling of all segments, represented by $X = (\vec{X}^0, \ldots, \vec{X}^r)$, starts at the root segments $\vec{X}^r$ and progresses recursively until the finest level $\vec{X}^0 = (X_1^0, \ldots, X_n^0)$. This level represents the labeling of the image pixels and thus the final segmentation. We seek an optimal labeling of these segments that results in a segmentation $X$ that is as close as possible to the TD segmentation $S$ while minimizing the separation of *salient* segments into figure and background parts. A salient segment is defined in terms of its homogeneity and contrast with its surrounding. For example, a uniform black segment surrounded by a white background is highly salient.

Using the above criteria, an optimal combination $X$ between a given BU and TD segmentation $S$ is derived by maximizing the following combination score:

$$\psi^s(X, S) = -\psi^t(X, S) + \psi^b(X). \tag{19}$$

The first term $\psi^t(X, S)$ models the agreement between the TD segmentation $S$ and the combined segmentation $X$. It penalizes segmentations $X$ that are far away in terms of euclidean distance (squared) from the TD segmentation $S$:

$$\psi^t(X, S) = \sum_q (S_q - X_q^0)^2, \tag{20}$$

where $S_q$ and $X_q^0$ represent the TD and final labeling of pixel $q$.

The second term $\psi^b(X)$ depends on parent-child (pairwise) labeling. It penalizes labeling $X$ in which a segment and its parent are labeled inconsistently:

$$\psi^b(X) = \sum_{l,i} \phi^b(X_i^l, par(X_i^l)), \tag{21}$$

where $(X_i^l, par(X_i^l))$ is the labeling of segment $i$ at level $l$ and its parent, and the pairwise term $\phi^b$ depends on the segment's saliency:

$$\phi^b(X_i^l, par(X_i^l)) = \begin{cases} \log(2 - \Gamma_i^l) & \Big| \; X_i^l = par(X_i^l) \\ \log \Gamma_i^l & \Big| \; X_i^l \neq par(X_i^l). \end{cases} \tag{22}$$

The segment's saliency $\Gamma_i^l$ is defined in [46], (2) and renormalized here to be in (0, 1] and proportional rather than inversely proportional to the segment's saliency. The penalty for inconsistent parent-child labeling is therefore dependent on the child's saliency: It is smaller for salient than nonsalient segments, making the labeling of salient segments more independent of their parents. Note, for example, that changing the labeling of a segment whose saliency is maximal ($\Gamma_i^l = 1$) does not change the penalty. This captures the notion that if a segment is very different from its surrounding, the two are not required to have the same label.

The tree structure of the segmentation score (19) enables global optimization using a simple version of the max-sum algorithm [47]. In our case, $\psi(X, S)$ is decomposed through the segmentation tree $T$ into the local terms of $(X_q^0 - S_q)^2$ and $\phi^b(X_i^l, par(X_i^l))$. The optimization can consequently be obtained by a dynamic programming computation, which requires only one BU and one TD message between neighboring nodes. Each node sends one message $m_{X_i^l}^{\uparrow}(x)$ to its parent during the BU phase and receives one message $m_{X_i^l}^{\downarrow}(x)$ from its parent during the TD phase. Each message consists of two values, one for $X_i^l = 0$ and the other for $X_i^l = 1$. The optimal labeling of every node with respect to (19) is given by combining the message it sent with the message received from its parent:

$$m_i^l(x) = m_{X_i^l}^{\uparrow}(x) + m_{X_i^l}^{\downarrow}(x). \tag{23}$$

By the end of this process, $m_i^l(x)$ is the max of $\psi(X, S)$ with respect to $X$, when variable $X_i^l$ is held at $X_i^l = x$. The maximum of these two values ($x = 0$ or 1) therefore provides the label for $X_i^l$ in the configuration $X$ maximizing (19). The $m_i^l$ messages are also used to produce a confidence map that indicates labeling confidence: Changing the classification of a low confidence segment will not significantly affect $\psi(X, S)$ in comparison to a similar change in a high confidence segment. The certainty or confidence in a segment's labeling is therefore expressed using the difference between the maximal value of $\psi$ when the region is classified as a figure versus its maximal value when classified as a background. A large difference between the two values indicates high confidence, and vice versa. The confidence $r_i^l$, for a region defined by a segment $X_i^l$ in the segmentation tree, is given by $r_i^l = |m_i^l(1) - m_i^l(0)|$. Two different factors can produce low-confidence segmentation regions. The first, which may be termed BU uncertainty, arises in regions where there is no salient BU segment matching the TD segmentation $S$. The second, TD uncertainty, arises in regions where the TD classification votes are ambiguous ($S_q \to 0.5$). These are usually regions where the object shape is highly variable (such as the tail or legs in horses). In highly variable regions, the covering fragments may be inconsistent with each other, resulting in opposite labeling. These BU and TD uncertainties can be used to automatically identify regions that may benefit from additional processing. Section 5, Fig. 11 demonstrates the use of this confidence map.

## 5 Results

We tested our segmentation scheme on four classes: horses (320 images), cars (150), pedestrians (200), and runners (180). The horse, runner, and car databases were collected by an image search from the Internet and are available online.[2] The runner images contained one to five runners. Those with more than one were cropped to create a database of one runner per image. The pedestrian MIT database [52] is also available online. The images were highly variable and difficult to segment, as indicated by the BU segmentation results below. The horse images, for example, include horses in different poses (running, standing, eating, etc.), with different textures (for example, "Dalmation-like" horses) on highly cluttered different backgrounds.

Each segmentation experiment consisted of two main steps: fragment extraction (as described in Section 3.1) and four iterations of image segmentation and fragment labeling (Sections 4 and 3.2, respectively).

To evaluate the fragment labeling, we compared the number of fragment pixels labeled consistently by the automatic scheme versus a benchmark obtained through the manual figure-ground labeling of these fragments by two observers.

To evaluate the image segmentation (specifically, the relative contribution of its BU, TD, and combination components), the entire horse and runner databases and a smaller subset of 50 cars were manually segmented, thus providing a comparison benchmark. Three different measures were then examined using this benchmark:

The first, related to accuracy along the object boundary, was the *average distance* (in pixel units) between a given segmentation contour and the benchmark contour. The second was a *figure-consistency* regional measure (sometimes referred to as the *Jaccard coefficient*) that qualitatively compares the agreement between the figure regions produced by the algorithm $S$ and the figure regions judged manually $F$ using the following ratio $r = \frac{|F \cap S|}{|F \cup S|}$. The maximal ratio is 1.0, obtained only for perfect segmentation. The third was an *overall consistency* measure that counts the total number of pixels given the same label by the algorithm and the benchmark, divided by the total number of pixels.

Table 1 and Fig. 8 summarize the results obtained for the four classes.[3] Note that the TD segmentation improves the BU results and that these two approaches are further improved by their combination. For example, in horses, the combined segmentation was, on average, 67 percent more accurate than the TD contour alone (in terms of average distance). This improvement was even larger in object parts with highly variable shape. Note also that our algorithm had more difficulties learning and segmenting the runner than horse images.

The improvement offered by a multiscale combination can be demonstrated by its comparison with that of a single scale scheme. In the horse database, TD alone gives 0.58 figure consistency; the combined scheme limited to a single scale (selected as scale with best performance) gives 0.65 figure consistency; and the combined scheme with multiple scales gives 0.71 on the entire database. The overall improvement is 0.13, where 0.07 is contributed by a single scale, whereas additional 0.06 accuracy is added by multiscale combination. In the runners database, TD alone gives 0.23, combined scheme limited to a single scheme gives 0.3, and the combined scheme with multiple scales gives 0.43. The overall improvement in this case is therefore 0.2, where single scales contribute 0.07 improvement and multiscale contributes 0.13 to the improvement. The overall contribution of a single scale combination in these experiments is therefore in the range of 30-54 percent of the maximal contribution gained by using multiscale information.

The overall number of figure pixels covered versus the size of the fragment set provides a measure of the fragment selection efficiency. The plots in Fig. 8c compare the efficiency of the original max-min selection algorithm [40] with that of the selection method proposed here. This comparison shows that our method derives more efficient fragment sets and thus improved complete coverage of class objects. For example, the first 150 fragments selected using the pixel-based information criteria cover the same total area as the first 344 fragments selected by the previous measure.

Fig. 9 shows the first 12 fragments extracted for each class together with their automatic labeling. We also compared the TD segmentation produced using the automatically labeled fragments with the segmentation produced using the same fragments, labeled manually. The results were of equal accuracy, illustrating the accuracy of the automatic labeling (Fig. 10).

Fig. 11 shows some examples of BU, TD, and combined segmentations, demonstrating the relative merits of these approaches. Using the TD process, the objects are detected correctly and covered as complete contiguous entities in all images, despite their high variability in shape and cluttered background. However, this process may produce a figure-ground approximation that does not follow the image discontinuities and may also miss or produce erroneous figure regions (especially in highly variable regions such as the horse legs and tail). This is expected from pure TD segmentation, especially when fragments are extracted from as few as 50 training images. Salient BU segments can correct these errors and delineate precise region boundaries. However, they face difficulty in grouping relevant regions and identifying figure-ground boundaries. In these cases, the TD process completes the missing information. Even the combined process cannot always compensate, however, for cases where the TD completely misses a part of the object (for example, the horse legs in the third row).

Fig. 12 shows more examples for the other classes. The combination stage is similar to the TD stage but with significantly improved accuracy along the boundaries (Table 1).

[2] http://www.dam.brown.edu/people/eranb/databases/.

[3] The figure-ground segmentation constructed by the BU segmentation was restricted to segments taken from one level below the coarsest level (with an average of 3.4 segments). The resulting BU segments were then labeled manually as figure or background to maximize their consistency with the benchmark.

TABLE 1
Class Results

Fragment Selection & Labeling

| | Horses (320) | Pedestrians (200) | Runners (180) | Cars (150) |
|---|---|---|---|---|
| Number of source images | 50 | 19 | 30 | 50 |
| Number of candidates | 5496 | 1956 | 2706 | 1132 |
| Number of class-specific fragments | 123 | 86 | 122 | 99 |
| Number of covering fragments | 336 | 200 | 250 | 221 |
| Labeling consistency – $1^{st}$ iteration | 92% | 87% | 65% | 89% |
| Labeling consistency – last iteration | 94% | 90% | 84% | 91% |

Segmentation

| | **Horses** | **Runners** | **Cars** |
|---|---|---|---|
| Number of segmented images | 320 | 180 | 50 |
| **Distance measure (pixels)** | | | |
| BU | 17 | 16.3 | |
| TD | 5.21 | 7.28 | 12.27 |
| Combination (BU + TD) | 1.68 | 6.67 | 8.05 |
| **Figure consistency** ($\lvert F \cap S \rvert / \lvert F \cup S \rvert$) | | | |
| BU | 0.51 | 0.13 | |
| TD (entire database) | 0.58 | 0.26 | 0.4 |
| TD + BU single scale (entire database) | 0.65 | 0.3 | |
| TD + BU multi-scale (entire database) | 0.71 | 0.43 | 0.8 |
| TD (best 200 horses, 120 runners) | 0.71 | 0.31 | |
| TD + BU single scale (best 200 horses, 120 runners) | 0.78 | 0.37 | |
| TD + BU multi-scale (best 200 horses, 120 runners) | 0.8 | 0.53 | |
| **Overall consistency (%)** | | | |
| BU | 67% | 78% | |
| TD (entire database) | 84% | 81% | 89% |
| TD + BU single scale (entire database) | 85% | 82% | |
| TD + BU multi-scale (entire database) | 87% | 82% | 93% |
| TD (best 200 horses, 120 runners) | 88% | 84% | |
| TD + BU single scale (best 200 horses, 120 runners) | 90% | 85% | |
| TD + BU multi-scale (best 200 horses, 120 runners) | 92% | 86% | |

*(a) Fragment selection and labeling. (b) Segmentation.*

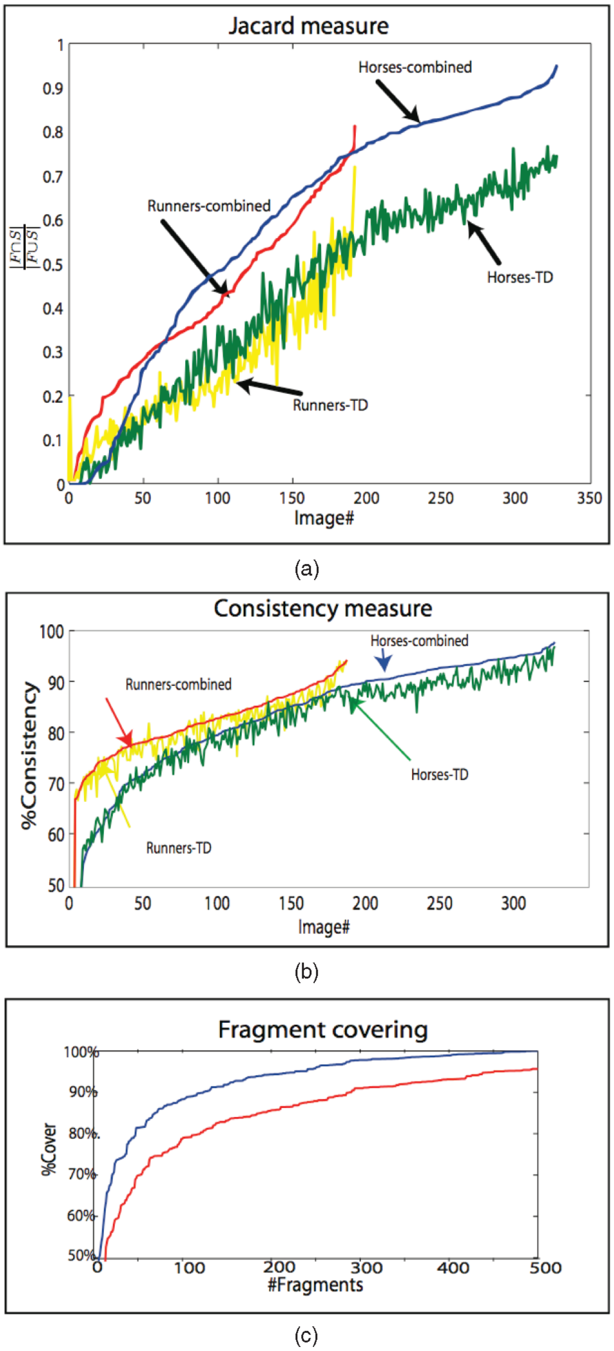
The contribution of the resulting segmentation to object detection was also tested and evaluated by ROC curves (detection rate versus false alarm rate). We compared the ROC curves of two detection methods (Fig. 13). The first uses a standard likelihood ratio test in the form of (11) without the segmentation consistency term ($\lambda = 0$) and the second with it ($\lambda = 15$). As seen in these curves, the segmentation consistency term significantly improves detection performance. Overlapping detected fragments are more likely to be figure-ground consistent when detected on the correct object. A consistency criterion therefore helps reduce false alarm rates by rejecting inconsistent detections resulting from a naive bayes assumption.

### 5.1 Comparison with Related Work

A quantitative comparison with related work is problematic due to the lack of standardized experimentation of such algorithms across the community—specifically, the choice of different databases, as well as subset of images used; the choice of information sources (for example, gray-level versus color or texture) and the evaluation measures used (overall consistency, figure-consistency, etc.). For instance, overall consistency, which compares the total number of pixels given the same label by the algorithm and the benchmark, divided by the total number of pixels, is the most popular measure used for reporting segmentation results. However, it does not necessarily provide a true indication of the segmentation accuracy: The baseline of this measure depends on the number of background pixels in the image. For example, if 90 percent of the image is background, then an algorithm that labels everything as background will perform a 90 percent consistency on this image. For the same reason, it may not be significantly affected by large deviations of the boundary. Therefore, to better compare this work with related approaches, we present here a number of comparisons, both quantitative and qualitative.

As described, our segmentation scheme was tested on four classes: horses (320), cars (150), pedestrians (200), and runners (180). The experiments use only gray-level information, thus posing more challenging segmentation than experiments using color information or experiments involving select images alone. The overall consistency for our combined approach in the horse database was 87 percent for the entire database and 92 percent for selected 200 horse and 82 percent for the entire runner database and 86 percent for selected 120 runners. (The selected examples were images in which the combined segmentation was most successful.) See Table 1 for additional results.

Kumar et al. [18] cite results with various cow and horse images (the numbers are not noted). For the cow images,



(a)

(b)

(c)

Fig. 8. The plots compare the TD segmentation and the final combined segmentation results for each image in the horse (320) and runner (180) databases. The results are given in terms of the (a) figure-consistency measure $\frac{|F \cap S|}{|F \cup S|}$ and (b) overall consistency and are ordered from the worst to best final combined segmentation. Note that the combined segmentations are usually above the TD segmentations. (c) The fragment covering plot shows selection efficiency. It compares the average cover area of objects versus the number of selected fragments obtained by two different selection methods. The red curve (lower) is for the original max-min selection and the blue curve (upper) is for the modified version proposed in this paper (Section 3.1).

95.82 percent of the foreground pixels and 98.72 percent of the background pixels were labeled consistently. For the horse images, 89.39 percent of the foreground pixels and 99.5 percent of the background pixels were labeled consistently. Winn and Jojic [19] report two types of results for their approach. The first takes into account color information, the second texture: The consistency for 200 of the horse images was 93.1 percent (color) and 93.0 percent (texture) and for the side view of 20 cars 91.4 and 94 percent, respectively. Winn and Shotton [20] report an overall consistency of 96.5 percent and figure consistency of 0.67 on 34 manually labeled cars.

In summary, our scheme is unique compared with other approaches (reviewed in Section 2) in the following combination of properties: The algorithm learns the segmentation on its own using a small training set of still images thus minimizing manual intervention. This property stands in contrast to other methods requiring manual parameter of structure setting (for example, [11], [22], and [20]), as well as methods requiring large video sequences (for example, [18]). The fragment-based representation addresses high variability in object shape and appearance. This property stands in contrast to methods that are limited in their ability to account for high object variability (for example, [20], [19], and [17]). The TD part adds a novel labeling consistency constraint (13) that improves segmentation and detection results. It also provides a full rather than a partial object cover (for example, [27]). The combination with hierarchical BU segmentation takes into account object boundaries at multiple scales. This is in contrast to the combination methods that take into account boundaries at a single level, usually the pixel level (for example, [23], [24], and [18]).

## 6 Conclusions

This paper describes an image segmentation scheme that combines fragment-based TD segmentation with a multiscale hierarchical BU segmentation. When applied to a training set of nonsegmented class images, the method automatically learns a fragment bank consisting of image patches containing common object parts. This bank is used to segment the training images, as well as novel class objects. The method is able to address complex images characterized by high object variability and cluttered backgrounds. This is because each object part can be covered by a variety of fragments representing its different possible appearances (for example, a large repertoire of horse heads) and because the fragments themselves are allowed to move with respect to each other as long as they preserve figure-ground and spatial consistency. The method therefore produces a full cover of the object, and the BU is only required to make a final adjustment of the precise boundary location.

The scheme represents an integrated segmentation and recognition system: Recognition and segmentation are intertwined rather than proceeding in a serial or feedforward manner. Detection of characteristic object parts initializes the segmentation, and the resulting segmentation is used to improve object detection.

Similar representations have been used in the past for object classification, but for the purpose of segmentation, this representation was extended in two basic ways. The first is a figure-ground label that segments each stored fragment into its figure and background parts. This figure-ground labeling makes it possible to use the stored fragments for figure-ground segmentation of images. A

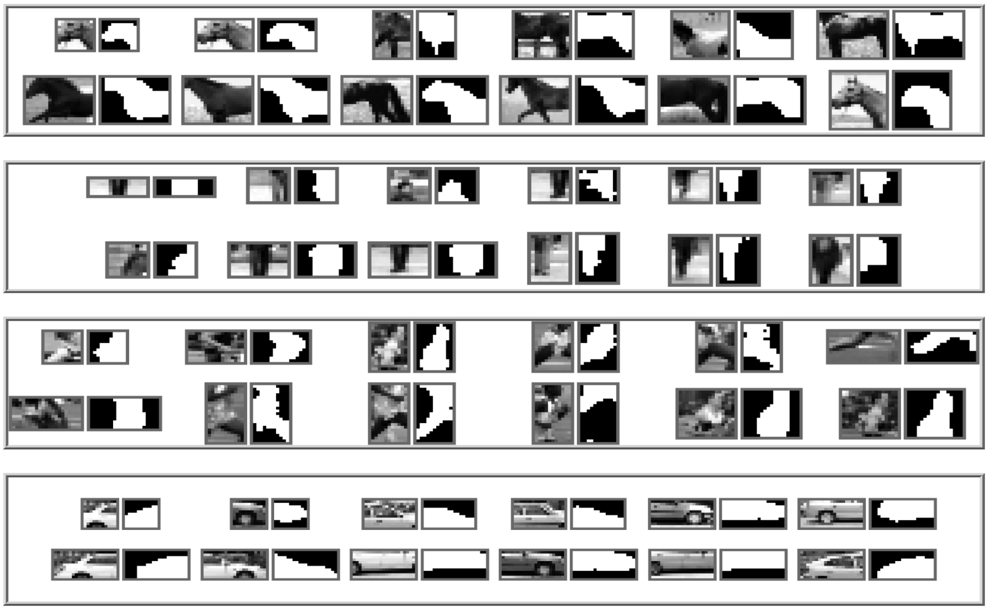


Fig. 9. Fragments and their automatic labeling. The first 12 fragments for the horse, pedestrian, runner, and car classes, together with their figure-ground labeling, as learned by the algorithm.

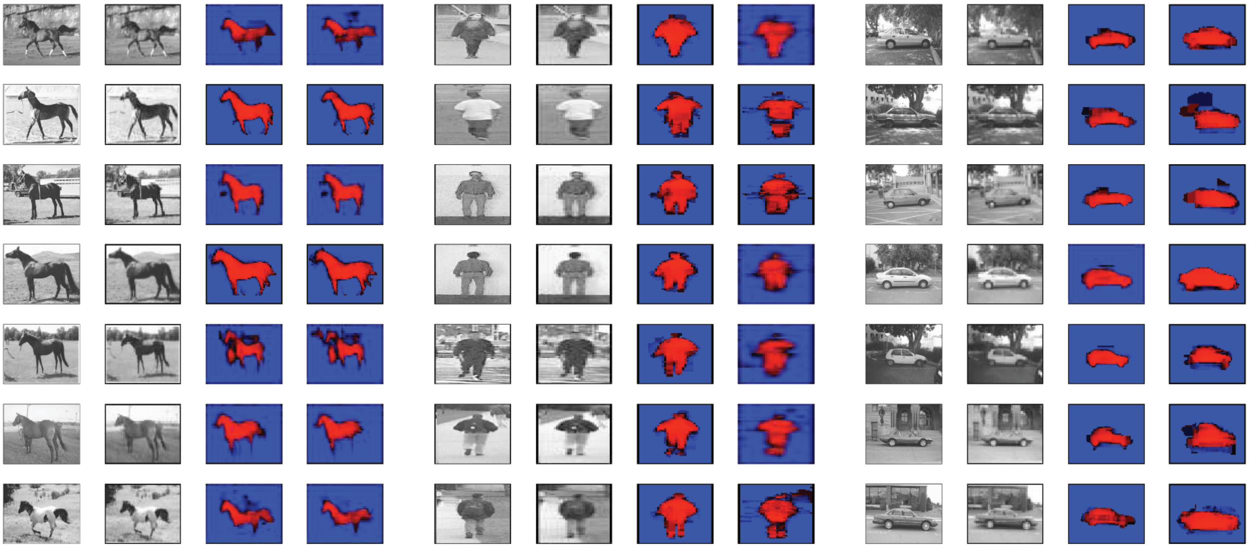


Fig. 10. TD segmentation produced by manually labeled fragments versus those labeled automatically. Left to right: input image, low-resolution input for our segmentation scheme, figure-ground segmentation map produced by manually labeled fragments, and figure-ground segmentation map produced by automatic fragment labeling.

"figure-ground consistency measure" between overlapping fragments was also introduced to improve cover consistency, leading to an overall improvement in both segmentation and recognition. Second, the use of fragments for segmentation requires an overcomplete object representation, in which fragments overlap and are thus able to produce complete rather than partial object covers. To do so, a new algorithm was designed to extend the fragment set, such that it included both class specific detection fragments and covering fragments.

In Section 3.2, we presented a method that is able to learn the figure-ground segmentation of complicated objects using difficult sets of unsegmented training images. This learning stage was motivated in part by evidence indicating that the human visual system is capable of learning to segment unfamiliar objects even from complex training images [29]. The learning principle is that the figure part of the fragment is likely to contain features that repeat consistently in class images, whereas its background part is far more arbitrary and variable. The learning improves through iterative refinement (bootstrapping): As the system segments more and more images, the segmentation of individual fragments improves the segmentation of entire objects, which in turn refines the segmentation of individual fragments.

The combined segmentation integrates the predicted object shape, given by the (TD) stored object representations and the (BU) image content. The combination applies an efficient message-passing algorithm using a single BU and single TD pass within the segmentation tree. The tree itself defines salient image regions (in terms of color, intensity, and texture) from multiple scales. This is a key point, since

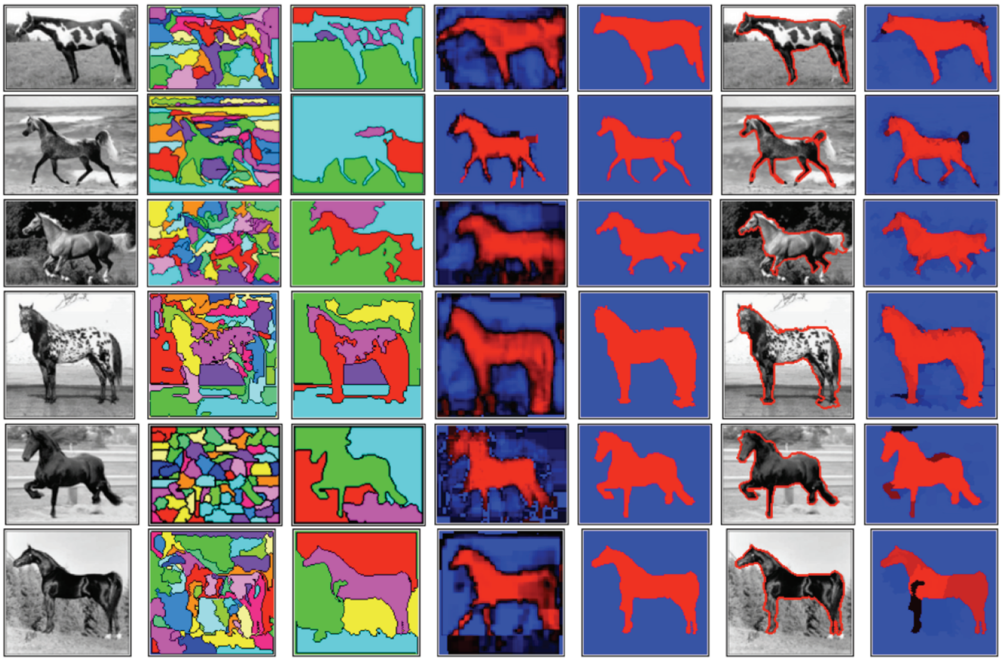


Fig. 11. Examples. Left to right: input image, BU segments (represented by different colors) at two scales (fine and coarse), TD segmentation $S(q)$, final (combined) segmentation $X(q)$, the figure-ground contour superimposed on the input image, and confidence in the final classification. Red/blue represents figure/ground classification, and brightness represents classification confidence.

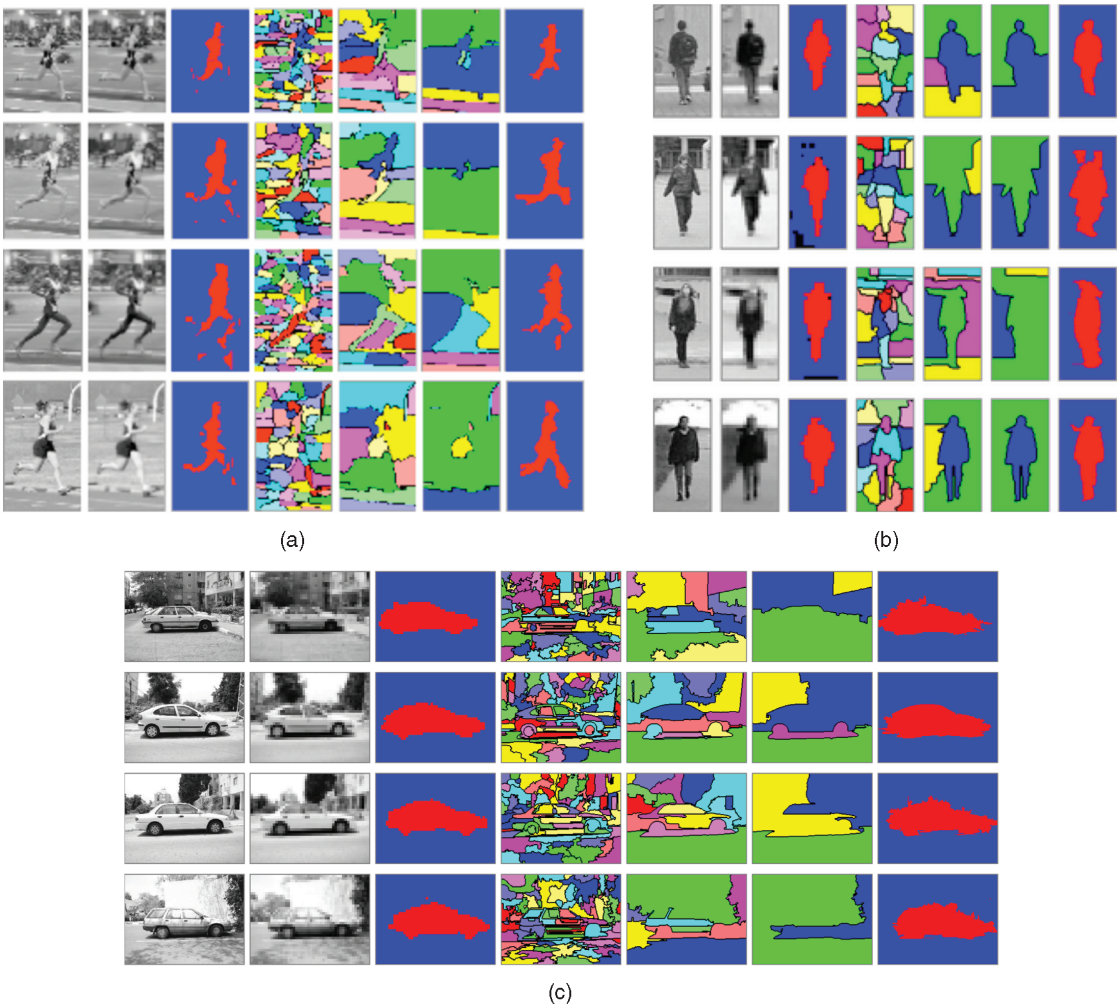
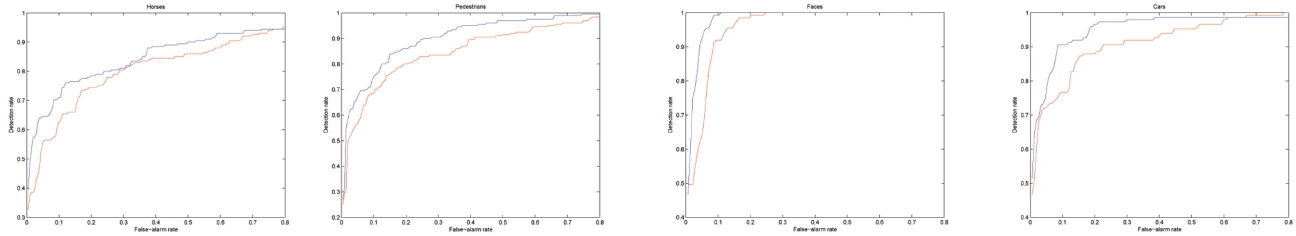


Fig. 12. More examples. Left to right: input image, TD input (low-resolution), TD cover, BU segmentation (three different scales), and combination output.



Horses Pedestrians Faces (CMU face database) Cars

Fig. 13. Segmentation is used to improve classification. ROC curve showing the contribution of the figure-ground labeling consistency between overlapping fragments. The values of the $y$-axis represent the detection rate for a given false alarm rate (as given by the $x$-axis). Note that for any false alarm rate, the detection rate is higher for the classifier with consistency of figure-ground labeling.

different object parts may be detected at different image scales. The combination also provides a confidence map that identifies image regions where the segmentation may not be accurate and might thus be improved by further BU or TD information. The hierarchical combination method has broad applicability, making it possible to combine a range of TD and BU approaches.

A remaining difficulty of the current cover construction scheme is that the TD part is performed on a single scale alone. The scheme takes into account only low-resolution images, which impedes the detection of narrow regions with high variability. To address this, we propose a future multiresolution scheme along the following lines: The ROI of a given subset of fragments in a fine scale would be determined by the detection of a single fragment at the next coarser level. For example, a blob at a coarse scale may be decomposed into ears, nose, and neck in the class of cows. Once this hierarchy of fragments (or ROIs) is defined (or learned automatically), a similar process to that described in Section 3.3 would be applied to produce a hierarchical object cover.

The object cover method can be applied to a given image several times until all class objects are detected and segmented. However, we did not evaluate the performance of the method on images containing more than a single class object.

In the future, the scheme may be extended in several directions. The first, as mentioned above, is to construct hierarchical fragment representation for TD segmentation. Another important extension would be to handle multiple, possibly occluding objects in the same image. The object cover could be applied to a given image several times, until all class objects are detected and segmented. An intriguing but challenging extension is the possible use of semantic information. Figure-ground and spatial consistency by themselves may not be enough to produce optimal object covers. For example, sometimes, both the spatial and consistency constraints that we applied allow the algorithm to cover horses with more (or less) than four legs. It will be useful to find how semantic information (for example, semantic meaning of fragments) can be learned from training data and incorporated into the current scheme to address these difficulties.

## Acknowledgments

The authors would like to thank Eitan Sharon for useful discussions and for his BU segmentation code. This work was supported by ISF Grant 7-0369 and EU IST Grant FP6-2005-015803.

## References

[1] E. Borenstein and S. Ullman, "Class-Specific, Top-Down Segmentation," *Proc. European Conf. Computer Vision (ECCV '02),* vol. 2, pp. 109-124, 2002.

[2] E. Borenstein and S. Ullman, "Learning to Segment," *Proc. European Conf. Computer Vision (ECCV),* 2004.

[3] E. Borenstein, E. Sharon, and S. Ullman, "Combining Top-Down and Bottom-Up Segmentation," *Proc. Computer Vision and Pattern Recognition Workshop Perceptual Organization in Computer Vision,* 2004.

[4] A. Weeks and G. Hague, "Color Segmentation in the HSI Color Space Using the k-Means Algorithm," *Proc. SPIE,* vol. 3026, pp. 143-154, Feb. 1997.

[5] J.D. Buf, M. Kardan, and M. Spann, "Texture Feature Performance for Image Segmentation," *Pattern Recognition,* vol. 23, 1990.

[6] U. Montanari, "On the Optimal Detection of Curves in Noisy Pictures," *Comm. ACM,* vol. 14, 1971.

[7] A. Shashua and S. Ullman, "Structural Saliency: The Detection of Globally Salient Structures Using a Locally Connected Network," *Proc. Int'l Conf. Computer Vision,* 1988.

[8] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, Nov. 1984.

[9] D. Mumford and J. Shah, "Boundary Detection by Minimizing Functionals," *Proc. Computer Vision and Pattern Recognition (CVPR '85),* 1985.

[10] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR),* 1997.

[11] A. Yuille and P. Hallinan, "Deformable Templates," *Active Vision,* A. Blake and A. Yuille, eds., pp. 21-38, MIT Press, 1992.

[12] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. Computer Vision,* vol. 1, pp. 321-331, 1987.

[13] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A Geometric Model for Active Contours in Image Processing," *Numerische Mathematik,* vol. 66, pp. 1-31, 1993.

[14] R. Malladi, J. Sethian, and B. Vemuri, "Shape Modeling with Front Propagation: A Level Set Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 2, pp. 158-175, 1995.

[15] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic Active Contours," *IEEE Trans. Pattern Analysis and Machine Intelligence,* 1995.

[16] B. Leibe and B. Schiele, "Interleaved Object Categorization and Segmentation," *Proc. British Machine Vision Conf. (BMVC),* 2003.

[17] E. Bernstein and Y. Amit, "Part-Based Statistical Models for Object Classification and Detection," *Proc. Computer Vision and Pattern Recognition (CVPR '05),* vol. 2, 2005.

[18] M.P. Kumar, P.H.S. Torr, and A. Zisserman, "Obj cut," *Proc. Computer Vision and Pattern Recognition (CVPR '05),* vol. 1, pp. 18-25, 2005.

[19] J. Winn and N. Jojic, "Locus: Learning Object Classes with Unsupervised Segmentation," *Proc. Int'l Conf. Computer Vision (ICCV '05),* vol. 1, pp. 756-763, 2005.

[20] J. Winn and J. Shotton, "The Layout Consistent Random Field for Recognizing and Segmenting Partially Occluded Objects," *Proc. Computer Vision and Pattern Recognition (CVPR),* 2006.

[21] S.X. Yu, R. Gross, and J. Shi, "Concurrent Object Recognition and Segmentation by Graph Partitioning," *Proc. Ann. Conf. Advances in Neural Information Processing Systems (NIPS),* 2002.

[22] X. Chen, Z. Tu, A. Yuille, and S. Zhu, "Image Parsing: Segmentation, Detection and Recognition," *Proc. Int'l Conf. Computer Vision (ICCV),* 2003.

[23] L. Liu and S. Sclaroff, "Region Segmentation via Deformable Model-Guided Split and Merge," *Proc. Int'l Conf. Computer Vision (ICCV),* 2001.

[24] G. Mori, X. Ren, A. Efros, and J. Malik, "Recovering Human Body Configurations: Combining Segmentation and Recognition," *Proc. Computer Vision and Pattern Recognition (CVPR),* 2004.

[25] X. Ren, C. Fowlkes, and J. Malik, "Scale-Invariant Contour Completion Using Conditional Random Fields," *Proc. Int'l Conf. Computer Vision (ICCV),* 2005.

[26] L. Zhao and L.S. Davis, "Closely Coupled Object Detection and Segmentation," *Proc. Int'l Conf. Computer Vision (ICCV '05),* vol. 1, 2005.

[27] A. Levin and Y. Weiss, "Learning to Combine Bottom-Up and Top-Down Segmentation," *Proc. European Conf. Computer Vision (ECCV),* 2006.

[28] M. Peterson, "Object Recognition Processes Can and Do Operate Before Figure-Ground Organization," *Current Directions in Psychological Science,* vol. 3, pp. 105-111, 1994.

[29] M.J. Brady and D. Kersten, "Bootstrapped Learning of Novel Objects," *J. Vision,* vol. 3, no. 6, pp. 413-422, 2003.

[30] A. Needham, "Object Recognition and Object Segregation in 4.5-Month-Old Infants," *J. Experimental Child Psychology,* vol. 78, pp. 3-24, 2001.

[31] P.C. Quinn and P.G. Schyns, "What Goes Up May Come Down: Perceptual Process and Knowledge Access in the Organization of Complex Visual Patterns by Young Infants," *Congnitive Science,* vol. 27, no. 6, pp. 923-935, 2003.

[32] R.S. Zemel, M. Behrmann, M.C. Mozer, and D. Bavelier, "Object Recognition Processes Can and Do Operate Before Figure-Ground Organization," *Experimental Psychology,* vol. 28, no. 1, pp. 202-217, Feb. 2002.

[33] K. Zipser, V. Lamme, and P.H. Schiller, "Contextual Modulation in Primary Visual Cortex," *J. Neuroscience,* vol. 16, no. 22, pp. 7376-7389, 1996.

[34] V. Lamme, "The Neurophysiology of Figure-Ground Segregation in Primary Visual Cortex," *J. Neuroscience,* vol. 15, no. 2, pp. 1605-1615, Feb. 1995.

[35] G.C. Baylis and J. Driver, "Shape-Coding in IT Cells Generalizes over Contrast and Mirror Reversal, but Not Figure-Ground Reversal," *Nature Neuroscience,* vol. 4, no. 9, pp. 937-942, 2001.

[36] J. Hupe, A. James, B. Payne, S. Lomber, and J. Bullier, "Cortical Feedback Improves Discrimination between Figure and Background by v1, v2 and v3 Neurons," *Nature,* vol. 394, pp. 784-787, Aug. 1998.

[37] H. Supper, H. Spekreijse, and V. Lamme, "Contextual Modulation in Primary Visual Cortex as a Neuronal Substrate for Working Memory," *J. Vision,* vol. 1, no. 3, p. 345, 2001.

[38] M.C. Burl, M. Weber, and P. Perona, "A Probabilistic Approach to Object Recognition Using Local Photometry and Global Geometry," *LNCS 1407,* 1998.

[39] S. Agarwal and D. Roth, "Learning a Sparse Representation for Object Detection," *Proc. European Conf. Computer Vision (ECCV '02),* vol. 4, pp. 113-130, 2002.

[40] S. Ullman and E. Sali, "Object Classification Using a Fragment-Based Representation," *Proc. First IEEE Int'l Workshop Biologically Motivated Computer Vision (BMCV '00),* pp. 73-87, 2000.

[41] F. Fleuret, "Fast Binary Feature Selection with Conditional Mutual Information," *J. Machine Learning Research,* Nov. 2004.

[42] D.N. Bhat and S.K. Nayar, "Ordinal Measures for Image Correspondence," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 4, pp. 415-423, Apr. 1998.

[43] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[44] K. Mikolajczyk and C. Schmid, "An Affine Invariant Interest Point Detector," *Proc. European Conf. Computer Vision (ECCV '02),* vol. 1, p. 128, 2002.

[45] S. Ullman, E. Sali, and M. Vidal-Naquet, "A Fragment Based Approach to Object Representation and Classification," *Proc. Fourth Int'l Workshop Visual Form,* 2001.

[46] M. Galun, E. Sharon, R. Basri, and A. Brandt, "Texture Segmentation by Multiscale Aggregation of Filter Responses and Shape Elements," *Proc. Int'l Conf. Computer Vision (ICCV '03),* pp. 716-723, 2003.

[47] F. Kschischang, B. Frey, and H. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Trans. Information Theory,* vol. 47, pp. 498-519, Feb. 2001.

[48] E. Sali and S. Ullman, "Detecting Object Classes by the Detection of Overlapping 2-D Fragments," *Proc. British Machine Vision Conf. (BMVC),* 1999.

[49] Y. Amit, "A Neural Network Architecture for Visual Selection," *Neural Computation,* vol. 12, no. 5, pp. 1141-1164, 2000.

[50] H. Schneiderman and T. Kanade, "Object Detection Using the Statistics of Parts," *Int'l J. Computer Vision,* vol. 56, no. 3, pp. 151-177, 2004.

[51] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, "Hierarchy and Adaptivity in Segmenting Visual Scenes," *Nature,* vol. 442, pp. 810-813, 2006.

[52] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian Detection Using Wavelet Templates," *Proc. Computer Vision and Pattern Recognition (CVPR '97),* pp. 193-199, 1997.

**Eran Borenstein** received the BSc degree (*cum laude*) in electrical engineering from the Israel Institute of Technology (Technion) in 1998 and the MSc and PhD degrees from the Department of Computer Science and Applied Mathematics, Weizmann Institute of Science in 2004. He then pursued a parallel postdoctoral fellowship at the Mathematical Sciences Research Institute (MSRI), Berkeley, California, and at the Department of Electrical Engineering and Computer Science, UC Berkeley from 2004 to 2005, followed by a postdoctoral fellowship at the Department of Applied Mathematics, Brown University from 2005 to 2007. He is currently a visiting research scientist at Brown University. His research focuses on image segmentation and recognition, computational modeling of the human vision, and face detection.

**Shimon Ullman** received the BSc degree from the Hebrew University, Jerusalem, and the PhD degree from MIT. He is the Samy and Ruth Cohn professor of computer science in the Department of Computer Science and Applied Mathematics, the Weizmann Institute of Science, Israel. Prior to the current position, he was a professor at the Brain and Cognitive Science and the Artificial Intelligence Laboratory, MIT. His research interests include vision and brain modeling, with emphasis on object classification and recognition. He is the author of the books: *The Interpretation of Visual Motion* (MIT Press, 1979) and *High-Level Vision* (MIT Press, 1996). He is the recipient of the 2008 David E. Rumelhart Award.