Learning to Segment

Eran Borenstein and Shimon Ullman*

Faculty of Mathematics and Computer Science
Weizmann Institute of Science
Rehovot, Israel 76100
{eran.borenstein,shimon.ullman}@weizmann.ac.il

Abstract. We describe a new approach for learning to perform class-based segmentation using only unsegmented training examples. As in previous methods, we first use training images to extract fragments that contain common object parts. We then show how these parts can be segmented into their figure and ground regions in an automatic learning process. This is in contrast with previous approaches, which required complete manual segmentation of the objects in the training examples. The figure-ground learning combines top-down and bottom-up processes and proceeds in two stages, an initial approximation followed by iterative refinement. The initial approximation produces figure-ground labeling of individual image fragments using the unsegmented training images. It is based on the fact that on average, points inside the object are covered by more fragments than points outside it. The initial labeling is then improved by an iterative refinement process, which converges in up to three steps. At each step, the figure-ground labeling of individual fragments produces a segmentation of complete objects in the training images, which in turn induce a refined figure-ground labeling of the individual fragments. In this manner, we obtain a scheme that starts from unsegmented training images, learns the figure-ground labeling of image fragments, and then uses this labeling to segment novel images. Our experiments demonstrate that the learned segmentation achieves the same level of accuracy as methods using manual segmentation of training images, producing an automatic and robust top-down segmentation.

1 Introduction

The goal of figure-ground segmentation is to identify an object in the image and separate it from the background. One approach to segmentation – the *bottom-up approach* – is to first segment the image into regions and then identify the image regions that correspond to a single object. The initial segmentation mainly relies on image-based criteria, such as the grey level or texture uniformity of image regions, as well as the smoothness and continuity of bounding contours. One of the major shortcomings of the bottom-up approach is that an object may be segmented into multiple regions, some of which may incorrectly merge the object

* This research was supported in part by the Moross Laboratory at the Weizmann Institute of Science.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3023, pp. 315–328, 2004.
© Springer-Verlag Berlin Heidelberg 2004

with its background. These shortcomings as well as evidence from human vision [1,2] suggest that different classes of objects require different rules and criteria to achieve meaningful image segmentation. A complementary approach, called *top-down segmentation*, is therefore to use prior knowledge about the object at hand such as its possible shape, color, texture and so on. The relative merits of bottom-up and top-down approaches are illustrated in Fig. 1.

A number of recent approaches have used fragments (or patches) to perform object detection and recognition [3,4,5,6]. Another recent work [7] has extended this fragment approach to segment and delineate the boundaries of objects from cluttered backgrounds. The overall scheme of this segmentation approach, including the novel learning component developed in this paper, is illustrated schematically in Fig. 2. The first stage in this scheme is fragment extraction (F.E.), which uses unsegmented class and non-class training images to extract and store image fragments. These fragments represent local structure of common object parts (such as a nose, leg, neck region etc. for the class of horses) and are used as shape primitives. This stage applies previously developed methods for extracting such fragments, including [8,4,5]. In the detection and segmentation stage a novel class image is covered by a subset of the stored fragments. A critical assumption is that the figure-ground segmentation of these covering fragments is already known, and consequently they induce figure-ground segmentation of the object. In the past, this figure-ground segmentation of the basic fragments, termed the fragment labeling stage (F.L.), was obtained manually. The focus of this paper is to extend this top-down approach by providing the capacity to learn the segmentation scheme from unsegmented training images, and avoiding the requirement for manual segmentation of the fragments.

The underlying principle of our learning process is that class images are classified according to their figure rather than background parts. While figure regions in a collection of class-image samples share common sub-parts, the background regions are generally arbitrary and highly variable. Fragments are therefore more likely to be detected on the figure region of a class image rather than in the background. We use these fragments to estimate the variability of regions within sampled class images. This estimation is in turn applied to segment the fragments themselves into their figure and background parts.

1.1 Related Work

As mentioned, segmentation methods can be divided into bottom-up and top-down schemes. Bottom-up segmentation approaches use different image-based uniformity criteria and search algorithms to find homogenous segments within the image. The approaches vary in the selected image-based similarity criteria, such as color uniformity, smoothness of bounding contours, texture etc. as well as in their implementation.

Top-down approaches that use class-based (or object-specific) criteria to achieve figure-ground segmentation include deformable templates [10], active shape models (ASM) [11] and active contours (snakes) [12]. In the work on deformable templates, the template is designed manually for each class of objects.

Fig. 1. Bottom-up and Top-down segmentation (two examples): Left – input images. Middle – state-of-the-art bottom-up segmentation ([9]). Each colored region (middle-left) represents a segment and the edge map (middle-right) represents the segments' boundaries. Right – class-specific segmentation (white contour) as learned automatically by our system. The bottom-up approach may segment objects into multiple parts and merge background and object parts as it follows prominent image-based boundaries. The top-down approach uses stored class-specific representation to give an approximation for the object boundaries. This approximation can then be combined with bottom-up segmentation to provide an accurate and complete segmentation of the object.

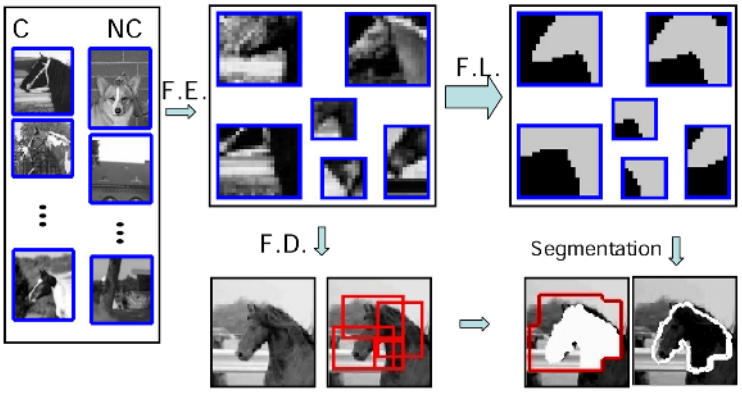

Fig. 2. The approach starts from a set of class (C) and non-class (NC) training images. The first stage is fragment extraction (F.E.) that extracts a set of informative fragments. This is followed by fragment-labeling (F.L.), the focus of this work, in which each fragment is divided into figure and background. During recognition, fragments are detected in input images (fragment detection, F.D.). The fragments' labeling and detection are then combined to segment the input images.

In schemes using active shapes, the training data are manually segmented to produce aligned training contours. The object or class-specific information in the active contours approach is usually expressed in the initial contour and in the definition of the external force. In all of the above top-down segmentation schemes, the class learning stage requires extensive manual intervention.

In this work we describe a scheme that automatically segments shape fragments into their figure and ground relations using unsegmented training images, and then uses this information to segment class objects in novel images from their background. The system is given a set of class images and non-class images and requires only one additional bit for each image in this set ("class" / "non-class").

2 Constructing a Fragment Set (Fragment Extraction)

The first step in the fragment-based approach is the construction of a set of fragments that represents the class and can be used to effectively recognize and segment class images. We give below a brief overview of the fragment extraction process. (further details of this and similar approaches can be found in [8,4,5].) The construction process starts by randomly collecting a large set of candidate fragments of different sizes extracted from images of a general class, such as faces, cars, etc. The second step is to select from the initial pool of fragments a smaller subset of the more useful fragments for detection and classification. These fragments are selected using an information measure criterion. The aim is that the resulting set be highly informative, so that a reliable classification decision can be made based on the detection of these fragments. Detected fragments should also be highly overlapping as well as being well-distributed across the object, so that together they are likely to cover it completely. The approach in [8] sets for each candidate fragment a detection threshold selected to maximize the mutual information between the fragment detection and the class. A fragment is subsequently detected in an image region if the similarity measure (absolute value of the normalized linear correlation in our case) between the fragment and that region exceeds the threshold. Candidates f_j are added to the fragment set F^s one by one so as to maximize the gain in mutual information $I(F^s;C)$ between the fragment set and the class:

$$f_j = \arg\max_f \left(I(F^s \cup f;C) - I(F^s;C)\right) \tag{1}$$

This selection process produces a set of fragments that are more likely to be detected in class compared with non-class images. In addition, the selected fragments are highly overlapping and well distributed. These properties are obtained by the selection method and the fragment set size: a fragment is unlikely to be added to the set if the set already contains a similar fragment since the mutual information gained by this fragment would be small. The set size is determined in such a way that the class representation is over-complete and, on average, each detected fragment overlaps with several other detected fragments (at least 3 in our implementation).

3 Learning the Fragments Figure-Ground Segmentation

To use the image fragments for segmentation, we next need to learn the figure-ground segmentation of each fragment. The learning process relies on two main criteria: *border consistency* and the *degree of cover*, which is related to the variability of the background. We initialize the process by performing a stage of bottom-up segmentation that divides the fragment into a collection of uniform regions. The goal of this segmentation is to give a good starting point for the learning process – pixels belonging to a uniform subregion are likely to have the same figure-ground labeling. This starting point is improved later (Sect. 5). A

number of bottom-up segmentation algorithms were developed in the past to identify such regions. In our implementation we use the algorithm developed by [9], which is fast (less than one second for an image with $240x180$ pixels) and segments images on several scales. We used scales in which the fragments are over-segmented (on average they divide the fragments into 9 subregions) providing subregions that are likely to be highly uniform. The algorithm was found to be insensitive to this choice of scale (scales that give on average $4-16$ subregions produce almost identical results). We denote the different regions of a fragment F by $R_1, R_2, \ldots, R_n$. Each region in the fragment (R_j) defines a subset of fragment points that are likely to have the same figure-ground label.

3.1 Degree of Cover

The main stage of the learning process is to determine for each region whether it is part of the figure or background. In our fragment-based scheme, a region R_j that belongs to the figure, will be covered on average by significantly more fragments than a background region R_i, for two reasons. First, the set of extracted fragments is sufficiently large to cover the object several times (7.2 on average in our scheme). Second, the fragment selection process extracts regions that are common to multiple training examples and consequently most of the fragments come from the figure rather than from background regions. Therefore, the number of fragments detected in the image that cover a fragment's region R_j can serve to indicate whether R_j belongs to the figure (high degree of cover) or background (low degree of cover). The average degree of cover of each region over multiple images, (denoted by r_j), can therefore be used to determine its figure-ground label. The value r_j is calculated by counting the average number of fragments overlapping with the region over all the class images in the training set. The higher r_j, the higher its likelihood to be a figure region (in our scheme, an average of 7.0 for figure points compared with 2.2 for background points). The degree of cover therefore provides a powerful tool to determine the figure-ground segmentation of the fragments. Using the degree of cover r_j $j = 1, \ldots, n$, for the n regions in the fragment, we select as the figure part all the regions with $r_j \geq \bar{r}$ for some selected threshold $\bar{r}$. That is, the figure part is defined by:

$$P(\bar{r}) = \bigcup_{\{j: r_j \geq \bar{r}\}} R_j \tag{2}$$

In this manner, all the regions contained in a chosen figure part $P(\bar{r})$ have a degree of cover higher or equal to $\bar{r}$, while all other regions have a degree of cover lower than $\bar{r}$. The segmentation of the fragment into figure and background parts is therefore determined by a single parameter, the degree of cover $\bar{r}$. Since $\bar{r} = r_k$ for some $k = 1, \ldots n$, the number of possible segmentations is now reduced from 2^n to n. This stage, of dividing the fragment into uniform regions and then ranking them using the degree of cover, is illustrated in Fig. 3. We next show how to choose from these options a partition that is also consistent with edges found in image patches covered by the fragment.

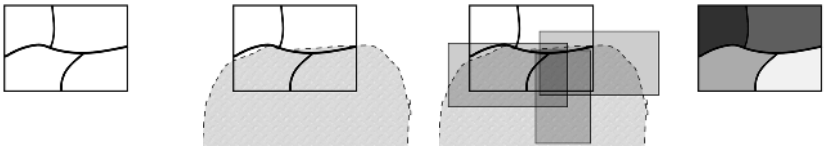

Fig. 3. Degree of cover: a fragment segmented into uniform regions (a) is detected on a given object (b). The degree of cover by overlapping fragments (also detected on the object) indicates the likelihood of a region to be a figure sub-region, indicated in (d) by the brightness of the region.

3.2 Border Consistency

The degree of cover indicates the likelihood of a fragment region to belong to the figure part. We next determine the boundary that optimally separates figure from background regions (such a boundary will exist in the fragment, unless it is an internal fragment). A fragment often contains multiple edges, and it is not evident which of these corresponds to the figure-ground boundary we are looking for. Using the training image set, we detect the fragment in different class-images. We collect the image patches where the fragment was detected, and denote this collection by $H_1, H_2, \ldots, H_k$. Each patch in this collection, H_j, is called a *fragment hit* and $H_j(x, y)$ denotes the grey level value of pixel (x, y) in this hit. In each one of these hits we apply an edge detector. Some edges, the class-specific edges, will be consistently present among hits, while other edges are arbitrary and change from one hit to the other. We learn the fragment's consistent edges by averaging the edges detected in these hits. Pixels residing on consistent edges will get a high average value, whereas pixels residing on noise or background edges will get a lower average, defined by:

$$D(x, y) = \frac{1}{k} \sum_{j=1}^{k} edge(H_j(x, y)) \tag{3}$$

Where $edge(\cdot)$ is the output of an edge detector acting on a given image. By the end of this process $D(x, y)$ is used to define the consistent edges of the fragment (see also Fig. 4).

We differentiate between three types of edges seen in this collection of hits. The first, defined here as the *border edge*, is an edge that separates the figure part of the fragment from its background part. This is the edge we are looking for. The second, defined here as an *interior edge*, is an edge within the figure part of the object. For instance, a human eye fragment may contain interior edges at the pupil or eyebrow boundaries. The last type, *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 and in the next section we show how to use them to segment the fragment.

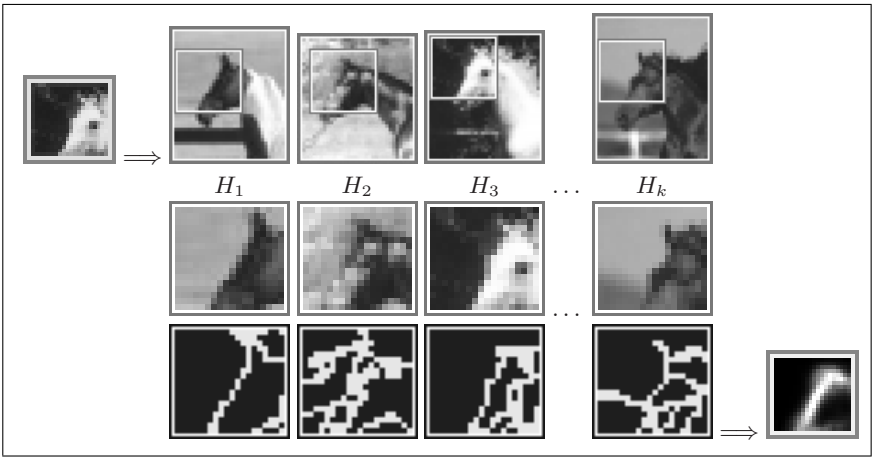

Fig. 4. Learning consistent edges. Fragment (top left) and the consistent boundary located in it (bottom right). To detect the consistent boundary, fragment hits $(H_1, \ldots H_k)$ are extracted from a large collection of training class images where the fragment is detected (Top row shows the hit location in the images, 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).

3.3 Determining the Figure-Ground Segmentation

In this section we combine the information supplied by the consistent edges computed in the last step with the degree of cover indicating the likelihood of fragment regions to be labeled as figure. The goal is to divide each fragment F, into a figure part P, and a complementary background part P^c in an optimal manner. The boundary between P and P^c will be denoted by ∂P. As mentioned, the set of consistent edges includes both the figure-ground boundary in the fragment (if such exists), as well as consistent internal boundaries within the object. Therefore, all the consistent edges should be either contained in the figure regions, or should lie along the boundary ∂P separating P from the background part P^c. A good segmentation will therefore maximize the following functional:

$$P = \arg\max_{P(\bar{r})} \left(\sum_{(x,y)\in P(\bar{r})} D(x,y) + \lambda \sum_{(x,y)\in \partial P(\bar{r})} D(x,y) \right) \tag{4}$$

The first term in this functional is maximized when the fragment's figure part contains as many as possible of the consistent edges. The second term is maximized when the boundary ∂P separating figure from ground in the fragment is supported by consistent edges. The parameter λ ($\lambda = 10$ in our implementation) controls the relative weights of the two terms.

Solving this problem is straightforward. As noted in (2), there are n possible values for $\bar{r}$, and each defines a possible segmentation of the fragment into a figure part $P(\bar{r})$ and background $P^c(\bar{r})$. It is therefore necessary to check which of the n

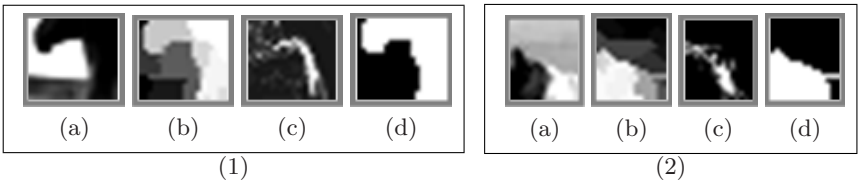

Fig. 5. Stages of fragment figure-ground segmentation (two examples). Given a fragment (a), we divide it into regions likely to have same figure-ground label. We then use the degree of cover to rank the likelihood of each region to be in the figure part of the fragment (b). Next,the fragment hits are used to determine its consistent edges (c). In the last stage, the degree of cover and the consistent edges are used to determine the figure-ground segmentation of the fragment (d).

options maximizes (4). This procedure alone produces good segmentation results, as discussed in the results section. The overall process is illustrated in Fig. 5. The figure depicts the stages of labeling two fragments that are difficult to segment. Note that by using the degree of cover and border consistency criteria it becomes possible to solve problems that are difficult to address using bottom-up criteria alone. Some parts of the contours (Fig. 5(1)) separating the figure from the background are missing in the fragment but are reconstructed by the consistent edges. Similarly, using the border consistency and degree of cover criteria, it is possible to group together dissimilar regions (eg. the black and white regions of the horse head in Fig. 5(2))

4 Image Segmentation by Covering Fragments

Once the figure-ground labels of the fragments are assigned, we can use them to segment new class images in the following manner. The detected fragments in a given image serve to classify covered pixels as belonging to either figure or background. Each detected fragment applies its figure-ground label to "vote" for the classification of all the pixels it covers. For each pixel we count the number of votes classifying it as figure versus the number of votes classifying it as background. In our implementation, the vote of each fragment had a weight $w(i)$. This value was set to the class-specificity of the fragment; namely the ratio between its detection rate and false alarms rate. The classification decision for the pixel was based on the voting result:

$$S(x,y) = \begin{cases} +1 \; if \;\; \sum_i w(i) L_i(x,y) > 0 \\ -1 \; if \;\; \sum_i w(i) L_i(x,y) \leq 0 \end{cases} \tag{5}$$

Where $\sum_i w(i) L_i(x,y)$ is the total votes received by pixel (x,y), and $L_i(x,y) = +1$ when the figure-ground label of detected fragment F_i votes for pixel(x,y) to be figure, $L_i(x,y) = -1$ when it votes for the pixel to be background. $S(x,y)$ denotes the figure-ground segmentation of the image: figure pixels are characterized by $S(x,y) = +1$ and background pixels by $S(x,y) = -1$.

The segmentation obtained in this manner can be improved using an additional stage, which removes fragments that are inconsistent with the overall cover using the following procedure. We check the consistency between the figure-ground label of each fragment L_i and the classification of the corresponding pixels it covers, given by $S(x, y)$, using normalized linear correlation. Fragments with low correlation (we used 0.65 as threshold) are regarded as inconsistent and removed from the cover. In the new cover, the figure-ground labels of covering fragments will consistently classify overlapping regions. The voting procedure (5) is applied again, this time only with the consistent fragments, to determine the final figure-ground segmentation of the image. The construction of a consistent cover can thus be summarized in two stages. In the first stage, all detected fragments are used to vote for the figure or ground labeling of pixels they cover. In the second stage, inconsistent fragments that "vote" against the majority are removed from the cover and the final segmentation of the image is determined.

5 Improving the Figure-Ground Labeling of Fragments

The figure-ground labeling of individual fragments as described in Sect. 3 can be iteratively refined using the consistency of labeling between fragments. Once the labeled fragments produce consistent covers that segment complete objects in the training images, a region's degree of cover can be estimated more accurately. This is done using the average number of times its pixels cover figure parts in the segmented training images, rather than the average number of times its pixels overlap with other detected fragments. The refined degree of cover is then used to update the fragment's figure-ground labeling as described in Sect. 3.3, which is then used again to segment complete objects in the training images. (As the degree of cover becomes more accurate, we can also use individual pixels instead of bottom-up subregions to define the fragment labeling.) This iterative refinement improves the consistency between the figure-ground labeling of overlapping fragments since the degree of cover is determined by the segmentation of complete objects and the segmentation of complete objects is determined by the majority labeling of overlapping fragments. This iterative process was found to improve and converge to a stable state (within 3 iterations), since majority of fragment regions are already labeled correctly by the first stage (see results).

6 Results

We tested the algorithm using three types of object classes: horse heads, human faces and cars. The images were highly variable and difficult to segment, as indicated by the bottom-up segmentation results (see below). For the class of horse heads we ran three independent experiments. In each experiment, we constructed a fragment set as described in Sect. 2. The fragments were extracted from 15 images chosen randomly from a training set of 139 class images (size $32x36$). The selected fragments all contained both figure and background pixels. The selection process may also produce fragments that are entirely interior to

the object, in which case the degree of cover will be high for all the figure regions. We tried two different sizes for the fragment set: in one, we used 100 fragments, which on average gave a cover area that is 7.2 times larger than the average area of an object; in the second, we used the 40 most informative fragments within each larger set of 100 fragments. These smaller sets gave a cover area that was 3.4 times the average area of an object. We initialized the figure-ground labels of the fragments using the method described in Sect. 3. We used the fragments to segment all these 139 images, as described in Sect. 4, and then used these segmentations to refine the figure-ground labels of the fragments, as described in Sect. 5. We repeated this refinement procedure until convergence, namely, when the updating of figure-ground labels stabilized. This was obtained rapidly, after only three iterations.

The fragments selected in these experiments all contained both figure and background pixels. The selection process may also produce fragments that are entirely interior to the object, in which case the degree of cover will be high for all the figure regions.

To evaluate the automatic figure-ground labeling in these experiments, we manually segmented 100 horse head images out of the 139 images, and used them as a labeling benchmark. The benchmark was used to evaluate the quality of the fragments' labeling as well as the relative contribution of the different stages in the learning process. We performed two types of tests: in the first (labeling consistency), we compared the automatic labeling with manual figure-ground labeling of individual fragments. For this comparison we evaluated the fraction of fragments' pixels labeled consistently by the learning process and by the manual labeling (derived from the manual benchmark).

In the second type of test (segmentation consistency), we compared the segmentation of complete objects as derived by the automatically labeled fragments; the manually labeled fragments; and a bottom-up segmentation. For this comparison we used the fraction of covered pixels whose labeling matched that given by the benchmark. In the case of bottom-up segmentation, segments were labeled such that their consistency with the benchmark is maximal . The output of the segmentation (given by using [9]) was chosen so that each image was segmented into a maximum of 4 regions. The average benchmark consistency rate was 92% for the case of automatically labeled fragments, 92.5% for the case of manually labeled fragments and 70% for the labeled bottom-up segments. More detailed results from these experiments are summarized in Table 1. The results indicate that the scheme is reliable and does not depend on the initial choice of fragments set. We also found that the smaller fragment sets (40th most informative within each bigger set) give somewhat better results. This indicates that the segmentation is improved by using the most informative fragments. The automatic labeling of the fragments is highly consistent with manual labeling, and its use gives segmentation results with the same level of accuracy as these obtained using fragments that are labeled manually. The results are significantly better than bottom-up segmentation algorithms.

Another type of experiment was aimed at verifying that the approach is general and that the same algorithm applies well to different classes. This was

Table 1. Results. This table summarizes the results of the first two type of tests we performed (labeling and segmentation consistency).

		Large set			Small set		
		Ex.1	Ex.2	Ex.3	Ex.1	Ex.2	Ex.3
Labeling consistency auto. vs. benchmark	Initial Labeling (sect. 3)	83%	88%	80%	86%	90%	93%
	Final Labeling (Sect. 5)	88%	91%	89%	93%	97%	95%
Segmentation consistency	fragments labeled automatically	90%	90%	92%	90%	90%	90%
	fragments labeled manually	92%	91%	94%	91%	95%	92%
	Bottom-up Segmentation	70%					

demonstrated using two additional classes: human faces and side view images of cars. For these classes we did not evaluate the results using a manual benchmark, but as can be seen in Fig. 6, our learning algorithm gives a similar level of segmentation accuracy as obtained with manually labeled fragments. Examples of the final segmentation results on the three classes are shown in Fig. 6. It is interesting to note that shadows, which appeared in almost all the training class images, were learned by the system as car parts.

The results demonstrate the relative merits of top-down and bottom-up segmentation. Using the top-down process, the objects are detected correctly as complete entities in all images, despite the high variability of the objects shape and cluttered background. Boundaries are sometimes slightly distorted and small features such as the ears may be missed. This is expected from pure top-down segmentation, especially when fragments are extracted from as few as 15 training images. In contrast, bottom-up processes can detect region boundaries with higher accuracy compared with top-down processes, but face difficulty in grouping together the relevant regions and identifying figure-ground boundaries – such as the boundaries of horse-heads, cars and human faces in our experiments.

7 Discussion and Conclusions

Our work demonstrates that it is possible to learn automatically how to segment class-specific objects, giving good results for both the figure-ground labeling of the image fragments themselves as well as the segmentation of novel class images. The approach can be successfully applied to a variety of classes. In contrast to previous class- and object-based approaches, our approach avoids the need for manual segmentation as well as minimizing the need for other forms of manual intervention. The initial input to the system is a training set of class and non-class images. These are raw unsegmented images, each having only one additional bit of information which indicates the image as class or non-class. The

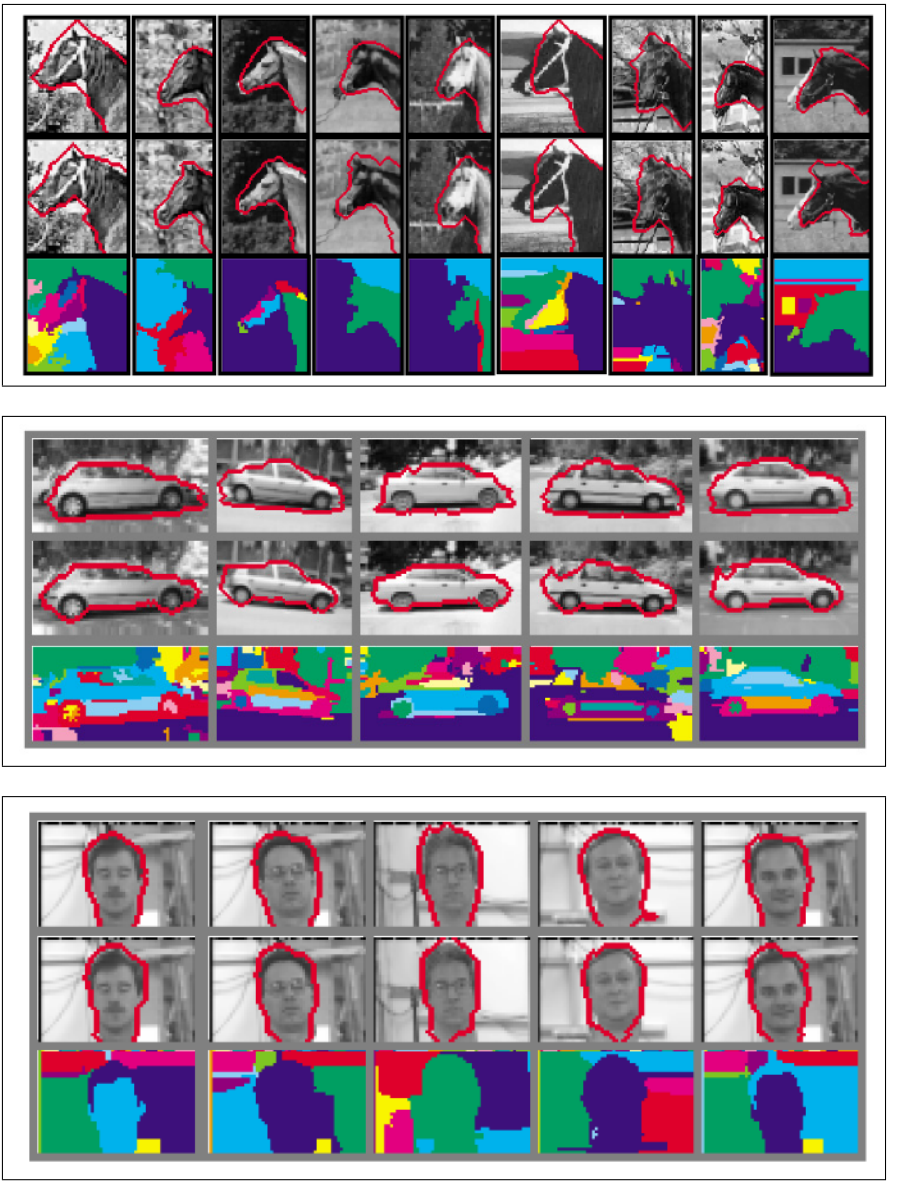

Fig. 6. Results. Rows 1-2,4-5,7-8 show figure-ground segmentation results, denoted by the red contour. The results in rows 1,4,7 are obtained using the automatic figure-ground labeling of the present method. The results in rows 2,5,8 are obtained using a manual figure-ground labeling of the fragments. Rows 3,6,9 demonstrate the difficulties faced in segmenting these images into their figure and background elements using a bottom-up approach [9]: segments are represented by different colors.

system uses this input to construct automatically an internal representation for the class that consists of image fragments representing shape primitives of the class. Each fragment is automatically segmented by our algorithm into figure and background parts. This representation can then be effectively used to segment novel class images.

The automatic labeling process relies on two main criteria: the degree of cover of fragment regions and the consistent edges within the fragments. Both rely on the high variability of background region compared with the consistency of the figure regions. We also evaluated another natural alternative criterion based on a direct measure of variability: the variability of a regions' properties (such as its grey level values) along the fragment's hit samples. Experimental evaluation showed that the degree of cover and border consistency were more reliable criteria for defining region variability – the main reason being that in some of the fragment hits, the figure part was also highly variable. This occurred in particular when the figure part was highly textured. In such cases, fragments were detected primarily based on the contour separating the figure from background region, and the figure region was about as variable as the background region. It therefore proved advantageous to use the consistency of the separating boundary rather than that of the figure part.

Another useful aspect is the use of inter-fragment consistency for iterative refinement: the figure-ground segmentation of individual fragments is used to segment images, and the complete resulting segmentation is in turn used to improve the segmentation of the individual fragments.

The figure-ground learning scheme combined bottom-up and top-down processes. The bottom-up process was used to detect homogenous fragment regions, likely to share the same figure-ground label. The top-down process was used to define the fragments and to determine for each fragment its degree of cover and consistent edges likely to separate its figure part from its background part. This combination of bottom-up and top-down processes could be further extended. In particular, in the present scheme, segmentation of the training images is based on the cover produced by the fragments. Incorporating similar bottom-up criteria at this stage as well could improve object segmentation in the training images and consequently improve the figure-ground labeling of fragments. As illustrated in Fig. 6, the top down process effectively identifies the figure region, and the bottom-up process can be used to obtain more accurate object boundaries.

Acknowledgment. The authors would like to thank Michel Vidal-Naquet for providing the fragment extraction software.

References

1. Needham, A., Baillargeon, R.: Effects of prior experience in 4.5-month-old infants' object segregation. Infant Behaviour and Development **21** (1998) 1–24
2. Peterson, M., Gibson, B.: Shape recognition contributions to figure-ground organization in three-dimensional displays. Cognitive Psychology **25** (1993) 383–429

3. Sali, E., Ullman, S.: Detecting object classes by the detection of overlapping 2-d fragments. In: BMVC, 10th. (1999) 203–213
4. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. In: ECCV. Volume I. (2000) 18–32
5. Agarwal, S., Roth, D.: Learning a sparse representation for object detection. In: ECCV. Volume IV. (2002) 113–130
6. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR. Volume II. (2003) 264–271
7. Borenstein, E., Ullman, S.: Class-specific, top-down segmentation. In: ECCV. Volume II. (2002) 109–124
8. Ullman, S., Sali, E., Vidal-Naquet, M.: A fragment based approach to object representation and classification. In: Proc. of 4th international workshop on visual form, Capri, Italy (2001) 85–100
9. Sharon, E., Brandt, A., Basri, R.: Segmentation and boundary detection using multiscale intensity measurements. In: CVPR. Volume I., Hawaii (2001) 469–476
10. Yuille, A., Hallinan, P.: Deformable templates. In: A. Blake and A. Yuille, editors, Active Vision, MIT press (1992) 21–38
11. Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models — their training and application. CVIU **61** (1995) 38–59
12. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1** (1987) 321–331