Combining Top-down and Bottom-up Segmentation

Eran Borenstein

Eitan Sharon

Shimon Ullman[1]

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

Division of Applied
Mathematics
Brown University
Providence, RI 02912
eitans@dam.brown.edu

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

Abstract

In this work we show how to combine bottom-up and top-down approaches into a single figure-ground segmentation process. This process provides accurate delineation of object boundaries that cannot be achieved by either the top-down or bottom-up approach alone. The top-down approach uses object representation learned from examples to detect an object in a given input image and provide an approximation to its figure-ground segmentation. The bottom-up approach uses image-based criteria to define coherent groups of pixels that are likely to belong together to either the figure or the background part. The combination provides a final segmentation that draws on the relative merits of both approaches: The result is as close as possible to the top-down approximation, but is also constrained by the bottom-up process to be consistent with significant image discontinuities. We construct a global cost function that represents these top-down and bottom-up requirements. We then show how the global minimum of this function can be efficiently found by applying the sum-product algorithm. This algorithm also provides a confidence map that can be used to identify image regions where additional top-down or bottom-up information may further improve the segmentation. Our experiments show that the results derived from the algorithm are superior to results given by a pure top-down or pure bottom-up approach. The scheme has broad applicability, enabling the combined use of a range of existing bottom-up and top-down segmentations.

1. Introduction

The goal of figure-ground segmentation is to identify an object in an 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. Relying mainly on continuity principles, this approach groups pixels according to the grey level or texture uniformity of image regions, as well as the smoothness and continuity of bounding contours. The main difficulty of this approach is that an object may be segmented into multiple regions, some of which may merge the object with its background.

This difficulty as well as evidence from human vision [1, 2], suggests that object recognition facilitates segmentation. A complementary approach, called *top-down segmentation*, is therefore to use prior knowledge about an object, such as its possible shape, color, or texture, to guide the segmentation [3]. The main difficulty in this approach stems from the large variability in the shape and appearance of objects within a given class. Consequently, segmentation results produced by a top-down approach alone may not accurately delineate the object's figure-ground boundary.

The relative merits of bottom-up and top-down approaches are illustrated in Fig. 1. In this paper we describe a scheme that integrates top-down with bottom-up segmentation to achieve optimal segmentation. The method combines two recently developed segmentation approaches: a top-down class-specific approach [3] that effectively addresses high variability within object classes and automatically learns the class representation from unsegmented training images [4]; and a bottom-up approach [5] that rapidly detects homogeneous image regions. In contrast with previous approaches for combining class-specific knowledge with bottom-up information ([6, 7, 8]), the combined approach presented here is fast (linear in the number of pixels) and takes into account image measurements at multiple scales, converging to a global optimum in just one pass. In addition, the combination is general and can be applied to combine a variety of top-down and bottom-up algorithms.

In section 2 we describe briefly the bottom-up and top-down schemes used. We then focus on the new combined approach.

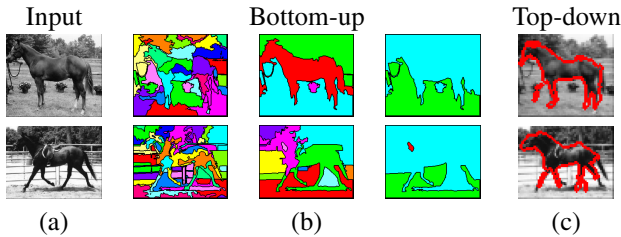| Input	Bottom-up	Top-down

(a)   (b)   (c)

Figure 1: The relative merits of the bottom-up and the top-down approaches. (a.) Input image. (b.) The bottom-up hierarchical segmentation [5] (at three different scales) can accurately detect image discontinuities but may also segment an object into multiple regions and merge object parts with the background. (Each region in uniform color represents a segment.) (c.) The top-down approach [3] provides a meaningful approximation for the figure-ground segmentation of the image, but may not follow exactly image discontinuities. The goal of this paper is to combine these approaches to achieve optimal segmentation.

2. Related work

Bottom-up segmentation

Bottom-up segmentation approaches use different image-based criteria and search algorithms to find homogenous segments within the image. A common bottom-up approach is to use a graph representation of the image (with the nodes representing pixels) and partition the graph into subsets corresponding to salient image regions. A recent bottom-up segmentation [5] (also used in this work) produces a multi-scale, hierarchical graph representation of the image. The resulting segmentation takes into account texture, average intensity and boundary properties of image-regions. The segmentation applies successive, recursive coarsening, in which homogenous segments at a given level are used to form larger homogenous segments at the next level. In this manner, the image is segmented into fewer and fewer segments. This process produces a *segmentation weighted, hierarchical graph* in which each segment is connected with a relating weight to any other segment at a coarser level, if the first was one of the segments used to define the latter. Neighboring segments within each level in the hierarchy are connected with appropriate weights that reflect their similarities. This coarsening process is produced efficiently, linear in the number of pixels.

The algorithm also provides a measure of saliency (related to a normalized-cut function) that ranks segments according to their distinctiveness. The saliency is reflected by an energy measure Γ_i that describes the segment's dissimilarity to its surrounding, divided by its internal homogeneity. Uniform segments that contrast with their surrounding (e.g. a uniform black segment on a white background) will be highly salient, and will therefore have very low energy Γ_i, whereas any segments against similar background will have low saliency and high energy Γ_i.

Top-down segmentation

The top-down segmentation approaches rely on acquired class-specific information, and can only be applied to images from a specific class. These include deformable templates [9], active shape models (ASM) [10] and active contours (snakes) [11]. A recent top-down, class-specific segmentation approach [3] (also applied in this work) deals with the high variability of shape and appearance within a specific class by using image fragments (or patches). These fragments are used as shape primitives for the class. Segmentation is obtained by covering the image with a subset of these fragments, followed by the use of this cover to delineate the figure boundaries. The approach can be divided into two stages – training and segmenting. In the training stage, a set of informative image fragments (*fragment set*) is constructed from training data to capture the possible variations in the shape and appearance of common object parts within a given class. The figure-ground segmentation of each fragment is then learned automatically [4], or can be given manually, and used for the segmentation stage. A set of *classifying fragments* (i.e., fragments that are highly likely to be detected in class images compared with non-class images) are derived from the fragment set. These are used in the segmentation stage to classify a novel input image as well as to detect the approximate location and scaling of the corresponding objects [12, 13, 14, 15].

The entire fragment set also provides an over-complete representation of the class. For instance, for the class of horses, the set contains a large repertoire of fragments representing different options for the appearance and shape of the legs, tail, body, neck and head. Consequently, in a given class image, detected fragments are overlapping, and together they are likely to completely cover the entire object. The same image area can be covered by a few alternative fragments (e.g. different horse heads, legs etc.). The fragments are also free to move with respect to each other as long as they preserve consistency, allowing variety in shape. Each covering fragment applies its figure-ground segmentation (also termed figure-ground labeling) to vote for the classification of the pixels it covers. The overall voting defines a figure-ground segmentation map $T(x, y)$ of the image, which classifies each pixel in the image as figure or background. The map can be given in either a deterministic form (a pixel can be either figure, $T(x, y) = 1$, or background, $T(x, y) = -1$) or a probabilistic form (figure with likelihood $T(x, y)$ and background with likelihood $1 - T(x, y)$). An overview of the approach is illustrated in Fig. 2.

3. Combining Top-down & Bottom-up

In this section we describe in detail how the bottom-up and top-down processes are integrated into a combined segmen-
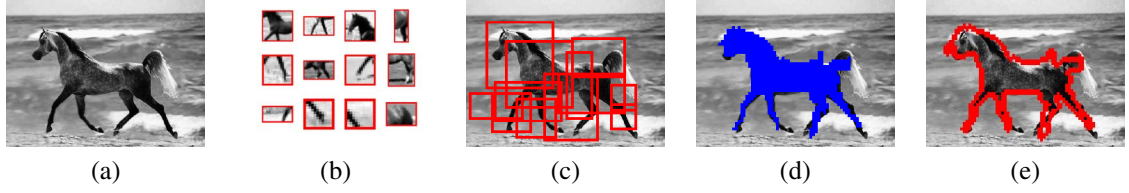
Figure 2: Overview of the top-down approach. (a.) Input image. Fragments stored in memory (b) are used to detect and cover the object (c).(d.) The figure-ground labeling of the covering fragments is used to label the image pixels as figure (blue region) or background. (e.) The boundary defined by the figure-ground segmentation (red contour).
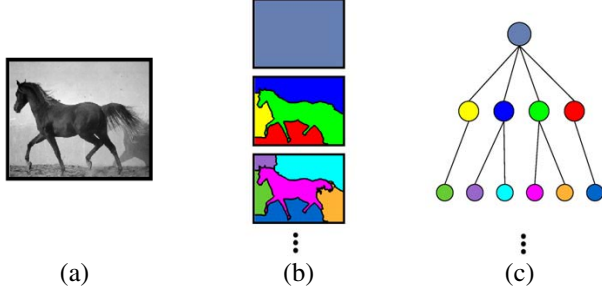

Figure 3: The bottom-up segmentation tree. (a.) Input image. (b.) Bottom-up segmentation of (a) at multiple scales. (c.) Segmentation tree. The bottom-up segments are organized in a tree structure. (The colors of the segments in (b) match those of the corresponding nodes in (c).)

tation scheme. The final goal is to construct a classification map $C(x, y)$ that makes the best possible compromise between a top-down requirement and a bottom-up constraint. The top-down requirement is to make C as close as possible to the initial top-down classification map T. The bottom-up constraint requires C to match the image structure, so that pixels within homogenous image regions, as defined by the bottom-up process, are likely be segmented together into either the figure or background part of the image. We deduce from the hierarchical graph constructed by [5] a *segmentation tree* (See Fig. 3). We also use the energy measure Γ_i provided by this bottom-up algorithm. This measure is modified by a monotonic function to obtain another measure of saliency $h(\Gamma_i)$, such that $h_i \in [0, 1)$ and $h_i \to 1$ as the segment S_i becomes more and more salient.

The combined segmentation algorithm proceeds by assigning each segment in the segmentation tree to either "figure" or "background," as indicated by the segment's corresponding label $s_i = +1/-1$. A configuration vector $\bar{s}$ represents the labeling s_i of all the segments in the segmentation tree and the configuration space Ω is defined as $\Omega = \{-1, +1\}^N$ where N is the number of segments. Each configuration vector $\bar{s}$ defines a figure-ground classification at all levels of the segmentation tree. A segment S_i at a given level classifies its pixels by the corresponding label s_i. The label of a segment s_i can be different from the label

s_i^- of its parent, and therefore the labeling at a finer level can change and overrule coarser level labeling. The classification map $C(\bar{s})$ is therefore determined by the labeling at the finest level of the tree, but, as explained below, the labeling at all levels is important to define the quality, or cost, of a proposed segmentation. In this manner, uniform regions created by the bottom-up process can be split into figure and background parts by smaller segments, and the final segmentation can reach pixel and even sub-pixel accuracy.

We define a cost function that evaluates the classification map defined by each configuration vector, with the aim of obtaining an optimal compromise between top-down and bottom-up penalty terms. The top-down term penalizes configuration vectors that result in classifications far away (in terms of Euclidean distance) from the initial top-down approximation. The bottom-up term penalizes configuration vectors that separate homogenous image regions into figure and background parts. The full cost function is defined in such a manner that its global minimum can be computed efficiently, using only a single bottom-up pass of the segmentation tree followed by a single top down pass. The cost function $P : \Omega \to \Re$ is defined as follows:

$$P(\bar{s}) = \sum_i p_i(s_i, s_i^-) \qquad (1)$$

The overall cost $P(\bar{s})$ is expressed as the sum of local cost for segment S_i to classify its pixels by s_i given that its father classified its pixels by s_i^-. Each local cost term contains a top-down t_i and bottom-up b_i parts:

$$p_i(s_i, s_i^-) = t_i(s_i) + b_i(s_i, s_i^-) \qquad (2)$$

The distance between the final classification $C(\bar{s})$ and the top-down classification T depends only on the labeling of the terminal segments of the tree (they determine the final classification). Therefore, the top-down term t_i is defined as follows:

$$t_i(s_i) = \begin{cases} \sum_{(x,y) \in S_i} (s_i - T(x,y))^2 & \text{if } S_i \text{ is a leaf} \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

The overall contribution of the top-down terms $\sum_i t_i$ is the Euclidean distance $||C(\bar{s}) - T||^2$.

3

The bottom-up part is determined by the segmentation tree. A segment is expected to have the same label as its parent if it is a low-salience segment, but can have its own independent label if it is a high-salience segment. The cost function therefore penalizes a segment if its label is different from its parent's label, unless it is a salient segment. The saliency itself is supplied as a part of the bottom-up segmentation. Specifically, the bottom-up term $b_i(s_i, s_i^-)$ is defined as follows:

$$b_i(s_i, s_i^-) = \lambda |S_i| (1 - h_i)(s_i - s_i^-)^2 \qquad (4)$$

λ is a predetermined constant that controls the tradeoff between the top-down and the bottom-up terms (in our experiments we used $\lambda = 4$), $|S_i|$ denotes the size of the segment (the number of its pixels), and h_i is the segment's saliency.

The parent S_r^- of the root segment is undefined and so are s_r^- and some of the terms in (1),(2). We define $S_r^- = \phi$ and $s_r^- = s_r$ so all the terms involving the root segment become well defined.

3.1. Minimizing the cost function

We find the global minimum of the cost function (1) using the sum-product algorithm. The sum-product algorithm is a generalization of a number of well-known algorithms including Pearl's belief propagation, Forward/backward algorithm, Viterbi algorithm and the Kalman filter. We describe below the application of the algorithm to the segmentation tree. A full description of the general algorithm can be found in [16].

Let $f(x_1, \ldots, x_n)$ be a function of n variables, that can be expressed as the sum of more local functions: $f = \sum_i f_i(\bar{x}_i)$ where $\bar{x}_i$ is a subset of $(x_1, \ldots, x_n)$. We seek a configuration of the variables that will minimize f, that is, $\min_{x_1, \ldots, x_n} f(x_1, \ldots, x_n)$. It turns out that if the decomposition of f forms a so-called "factor tree," then the minimization can be found efficiently by a simple message-passing scheme that requires only two messages between neighboring nodes in the tree.

In our case, the decomposition in (1) forms such a factor tree, which is the segmentation tree. In this tree, the local cost p_i (2) defines a weighted edge between a segment S_i and its parent S_i^-. The global function f is the summation of all these costs as defined by (1).

The computation proceeds by sending messages between neighboring nodes in the tree. This computation is composed of a bottom-up and a top-down phase. During the bottom-up phase, each node sends a message to its parent, and during the top-down phase each node receives a message from its parent. The computation terminates for all nodes when each node has sent to and received one message from its parent. The messages from a node containing variable s consists of two values, one for $s = -1$, the other

for $s = +1$. We denote the bottom-up messages going from s to its parent s^- by $m_s^\uparrow(s)$. Similarly $m_{s_j}^\downarrow(s)$ denotes the top-down messages going from s to one of its descendents s_j. Again, the message consists of two values for $s = -1$ and $s = +1$. The node s and its neighbors are shown in Fig. 4 for the bottom-up and top-down messages.

The bottom-up message from node s to its parent s^- is computed locally from the messages arriving from its descendents s^+, and is given by:

$$m_s^\uparrow(x) = \qquad (5)$$
$$\sum_{i=1}^{k} \min[p_i(-1, x) + m_{s_i}^\uparrow(-1), p_i(+1, x) + m_{s_i}^\uparrow(+1)]$$

This formula defines a recursive computation for the bottom-up messages. Once a node s receives all the messages coming from its descendents, it produces an outgoing message to its parent. The recursive computation of these messages starts at the leaf nodes lacking descendents, and propagates all the way up to the root.

The top-down message $m_{s_j}^\downarrow(x)$ arriving to s_j from its parent s is also computed locally, using the neighbors of s (its parent s^- and all its descendents s^+, except for s_j):

$$m_{s_j}^\downarrow(x) = \qquad (6)$$
$$\sum_{i \neq j} \min[p_i(-1, x) + m_{s_i}^\uparrow(-1), p_i(+1, x) + m_{s_i}^\uparrow(+1)]$$
$$+ \min[p_s(x, -1) + m_s^\downarrow(-1), p_s(x, +1) + m_s^\downarrow(+1)]$$

After all the bottom-up messages are sent, a recursive computation of the top-down messages can start at the root and propagate towards the leaves. Once the message passing is complete, we combine at each node the message to and from the parent in the following way:

$$m_s(-1) = m_s^\uparrow(-1) + min_x[p_s(-1, x) + m_s^\downarrow(x)]$$
$$m_s(+1) = m_s^\uparrow(+1) + min_x[p_s(+1, x) + m_s^\downarrow(x)]$$
$$(7)$$

It can be shown that $m_s(-1) = \min f(s_1, \ldots, s = -1, \ldots, s_n)$ (that is, the minimum of f when variable s is held at -1), and similarly $m_s(+1) = \min f(s_1, \ldots, s = +1, \ldots, s_n)$. The minimal of these two values is the selected label of node s in the configuration $\bar{s}$ minimizing f. This is also used in deriving a confidence map as explained in the next section. For additional details of this message passing algorithm and its properties see [16].

3.2. Confidence map

The combined segmentation scheme is also used to derive a confidence map that serves to indicate regions in $C(x, y)$

4

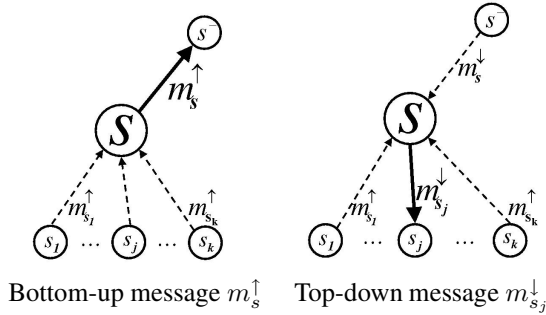Bottom-up message $m_s^\uparrow$ Top-down message $m_{s_j}^\downarrow$

Figure 4: Messages to and from a node s. Each message sent from a node to one of its neighbors is computed locally using all the messages coming from the other neighbors.

where segmentation is uncertain. Changing the classification of low confidence regions will not affect the cost function significantly, while such a change in high confidence region will increase the cost function significantly. The certainty or confidence in a region's classification is therefore expressed using the difference between the minimal cost when the region is classified as figure versus the minimal cost of its classification as background. A large difference (normalized by the segment's size) between the two values indicates high confidence, and a small difference indicates low confidence. The confidence r_s, for a region defined by a segment s in the segmentation tree, is given by:

$$r_s = \frac{|m_s(+1) - m_s(-1)|}{|S|} \tag{8}$$

These two terms $m_s(+1), m_s(-1)$ are given for every segment, without any additional computational cost, by the sum-product algorithm (7). In the results below we constructed the confidence map using the confidence of the terminal segments.

Two different sources can contribute to such segmentation uncertainty. The first, which may be termed bottom-up uncertainty, arises in regions where there is no salient bottom-up segment matching the top-down classification map $T(x, y)$. The second uncertainty, top-down uncertainty, arises in regions where the top-down classification is ambiguous. These are usually regions where the object shape is highly variable (such as the tail or the legs region in the class of horses). In these regions, overlapping fragments may be inconsistent with each other, resulting in inconsistent and therefore ambiguous classification of the pixels they are covering. The source for the low confidence, bottom-up vs. top-down uncertainty, can be determined automatically, and be used for selecting appropriate additional processing to improve the final segmentation.

4. Results

The top-down process briefly described in Sec. 2 was trained for the class of horse images (side views) using 340 horse images (class) and 430 non-horse images (non-class). A set of 425 informative fragments was extracted in the training stage from 54 class images. The bottom-up segmentation used is robust to different image properties. Therefore, in all our experiments we used identical parameters (weighting various image-based measurements, such as textural properties) for all images.

We tested 200 novel horse images, randomly chosen from various sites on the web (average image size is 190x245 pixels, object area ranges between 2722 to 17918 with average of 11447 pixels). These images are highly challenging – they include horses in different positions (running, standing, eating etc.) with different textures (e.g. zebra-like horses) on highly cluttered different backgrounds. The average distance (in pixel units) between a given segmentation contour and a benchmark contour (produced manually) was used to evaluate the figure-ground segmentation by the bottom-up method (17 pixels) [1], the top-down method (5.21 pixels) and the combination method (1.68 pixels). The average distance was further reduced (to 1.3 pixels) by removing from the average all contour points having a confidence measure less than 0.1 (5% of the points). The resulting confidence map efficiently separated regions of high and low consistency. The algorithm and benchmark consistently classified 95% of the pixels having high confidence (> 0.1) and 66% of the pixels having low confidence (≤ 0.1). The combined scheme improved the top-down contour by over 67% on average. This improvement was even larger in object parts with highly variable shape.

Figure 5 shows some examples. In some regions, the top-down 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). Salient bottom-up segments can correct these errors and delineate precise region boundaries (e.g. 1, 6). In other regions, the bottom-up segmentation may be insufficient in detecting the figure-ground contour, and the top-down process completes the missing information (e.g. the horse head in 4). Even the combined process cannot always compensate, however, for cases where the top-down completely misses a part of the object (e.g. horse legs in 3). The confidence map may be helpful in identifying such cases, since it effectively identifies image regions where additional

---

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

bottom-up and top-down information is needed. For example, parts of the horse legs and part of its head (in 3) are classified as background. However, as indicated by the confidence map, changing the classification of these parts would not significantly change the cost function.

5. Conclusion

The main advantage of our approach is the ability to use top-down information to group together segments belonging to the object despite image-based dissimilarity, and to break apart homogenous segments that contain both figure and background regions. The scheme can be used to combine other top-down and bottom-up processes, providing these are able to produce a hierarchical segmentation tree and an initial classification map, as defined in sections 2,3. For example, a rough manual approximation for the figure-ground segmentation of an image can be used by the algorithm to provide a new version of intelligent scissors [17].

The combined segmentation takes into account salient segments at multiple scales. Salient segments may detect image discontinuities that appear at one resolution but disappear at coarser resolutions (and vice versa). By using the entire segmentation tree, all image discontinuities at all scales are taken into account, to provide a final figure-ground segmentation. This segmentation provides an optimal compromise between the image content (bottom-up constraint) and the model (top-down requirement). This compromise is found through the minimization of a cost function defined over all possible segmentations. The global minimum of this cost function is deduced in only two passes of the segmentation tree.

The bottom-up and top-down components of the computation are also efficient: the bottom-up process provides the segmentation tree in complexity linear in image size, and the top-down process provides the initial classification map in complexity linear in the image size (low resolution in this case) and the number of fragments.

The results show that the combination successfully uses the relative merits of the two complementary approaches and significantly improves the figure-ground segmentation results within a learned class of images. The algorithm also provides a reliable confidence map indicating the regions of residual ambiguity, with no additional computational cost. In the future it will be of interest to examine the additional application of specialized processing to improve the segmentation of difficult, highly variable regions detected by this confidence map.

References

[1] A. Needham and R. Baillargeon, "Effects of prior experience in 4.5-month-old infants' object segregation," *Infant Behaviour and Development*, vol. 21, pp. 1–24, 1998.

[2] M. Peterson and B. Gibson, "Shape recognition contributions to figure-ground organization in three-dimensional displays," *Cognitive Psychology*, vol. 25, pp. 383–429, 1993.

[3] E. Borenstein and S. Ullman, "Class-specific, top-down segmentation," in *ECCV (2)*, 2002, pp. 109–124.

[4] E. Borenstein and S. Ullman, "Learning to segment," in *ECCV*, 2004.

[5] E. Sharon, A. Brandt, and R. Basri, "Segmentation and boundary detection using multiscale intensity measurements," in *CVPR (1)*, 2001, pp. 469–476.

[6] X. Chen, Z. Tu, A. Yuille, and S. Zhu, "Image parsing: Segmentation, detection and recognition," in *Proc. ICCV*, Nice, France, 2003, pp. 18–25.

[7] L. Liu and S. Sclaroff, "Region segmentation via deformable model-guided split and merge," in *ICCV (1)*, 2001, pp. 98 – 104.

[8] S. X. Yu, R. Gross, and J. Shi, "Concurrent object recognition and segmentation by graph partitioning," in *NIPS*, 2002.

[9] A. Yuille and P. Hallinan, "Deformable templates," in *Active Vision*, A. Blake and A. Yuille, Eds. MIT press, 1992, pp. 21–38.

[10] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models — their training and application," *CVIU*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

[11] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *IJCV*, vol. 1, pp. 321–331, 1987.

[12] S. Ullman, E. Sali, and M. Vidal-Naquet, "A fragment based approach to object representation and classification," in *Proc. of 4th international workshop on visual form*, Capri, Italy, 2001.

[13] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," in *ECCV (1)*, 2000, pp. 18–32.

[14] S. Agarwal and D. Roth, "Learning a sparse representation for object detection," in *ECCV (4)*, 2002, pp. 113–130.

[15] B. Leibe and B. Schiele, "Interleaved object categorization and segmentation," in *BMVC*, 2003.

[16] F. Kschischang, B. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[17] E. N. Mortensen and W. A. Barrett, "Interactive segmentation with intelligent scissors," *Graphical Models and Image Processing*, vol. 60, no. 5, pp. 349–384, Sept. 1998.
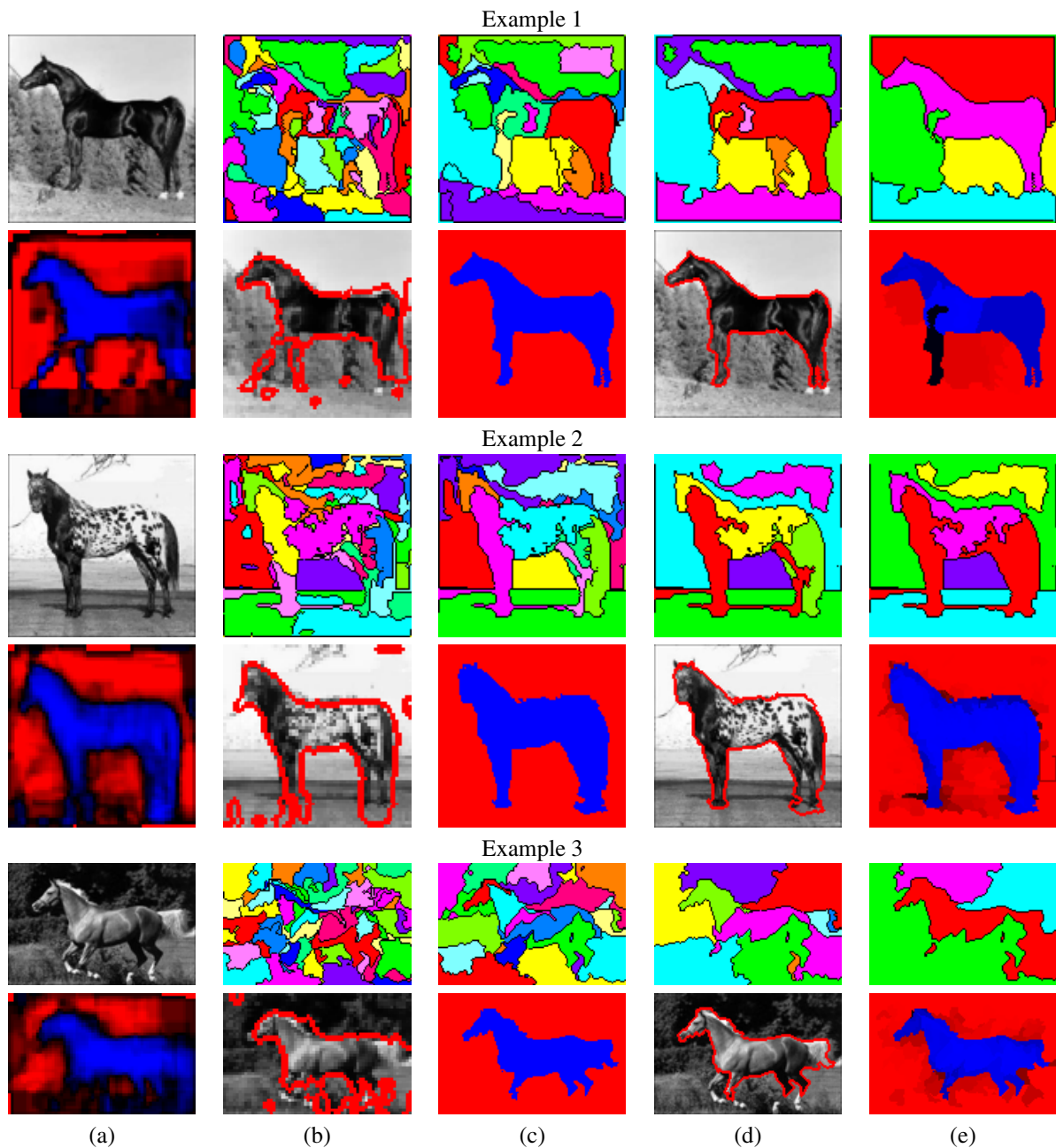
Figure 5: The first row in each example shows the input image and bottom-up segments at four scales (from fine to coarse). The segments are represented by different colors with their boundaries marked by a black contour. The second row shows: (a.) The initial classification map $T(x, y)$ (b.) The figure-ground contour derived from (a) superimposed on a low resolution version of the image (which constitutes the input to the top-down process). (c.) Final figure-ground segmentation $C(x, y)$ as constructed by the algorithm. (d.) The figure-ground contour derived from (c) superimposed on the input image. (e.) Confidence in the final classification. Blue/Red represents figure/ground classification and brightness represents classification confidence.

Example 4

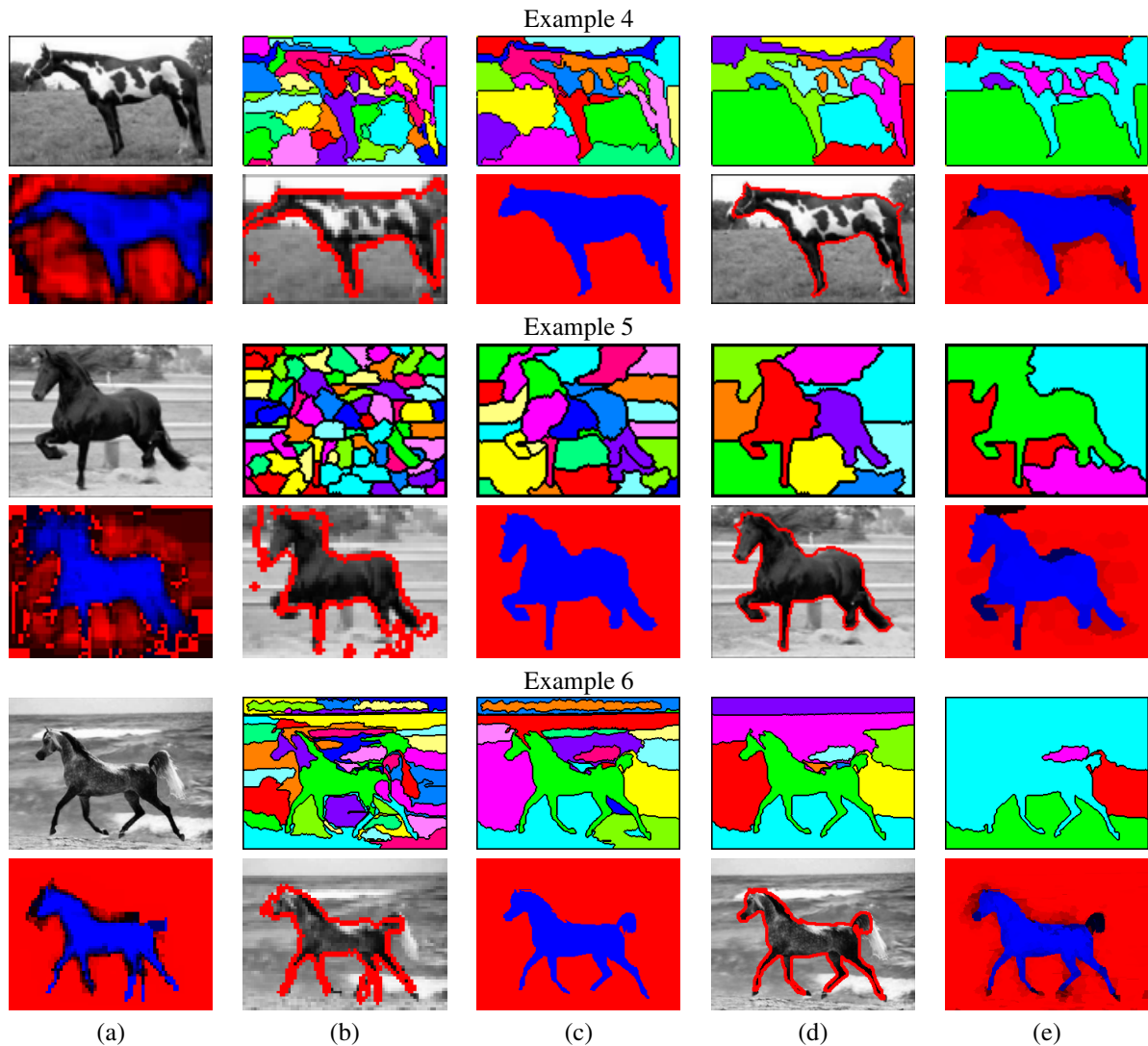Example 5

Example 6

(a) (b) (c) (d) (e)

Figure 6: The first row in each example shows the input image and bottom-up segments at four scales (from fine to coarse). The segments are represented by different colors with their boundaries marked by a black contour. The second row shows: (a.) The initial classification map $T(x, y)$ (b.) The figure-ground contour derived from (a) superimposed on a low resolution version of the image (which constitutes the input to the top-down process). (c.) Final figure-ground segmentation $C(x, y)$ as constructed by the algorithm. (d.) The figure-ground contour derived from (c) superimposed on the input image. (e.) Confidence in the final classification. Blue/Red represents figure/ground classification and brightness represents classification confidence.