Int J Comput Vis (2009) 83: 211–232
DOI 10.1007/s11263-009-0216-2

Shape Based Detection and Top-Down Delineation Using Image Segments

Lena Gorelick · Ronen Basri

Received: 5 May 2008 / Accepted: 16 January 2009 / Published online: 27 February 2009

Abstract We introduce a segmentation-based detection and top-down figure-ground delineation algorithm. Unlike common methods which use appearance for detection, our method relies primarily on the shape of objects as is reflected by their bottom-up segmentation.

Our algorithm receives as input an image, along with its bottom-up hierarchical segmentation. The shape of each segment is then described both by its significant boundary sections and by regional, dense orientation information derived from the segment's shape using the Poisson equation. Our method then examines multiple, overlapping segmentation hypotheses, using their shape and color, in an attempt to find a "coherent whole," i.e., a collection of segments that consistently vote for an object at a single location in the image. Once an object is detected, we propose a novel pixel-level top-down figure-ground segmentation by "competitive coverage" process to accurately delineate the boundaries of the object. In this process, given a particular detection hypothesis, we let the voting segments compete for interpreting (covering) each of the semantic parts of an object. Incorporating competition in the process allows us to resolve ambiguities that arise when two different regions are matched to the same object part and to discard nearby false regions that participated in the voting process.

We provide quantitative and qualitative experimental results on challenging datasets. These experiments demonstrate that our method can accurately detect and segment objects with complex shapes, obtaining results comparable to those of existing state of the art methods. Moreover, our method allows us to simultaneously detect multiple instances of class objects in images and to cope with challenging types of occlusions such as occlusions by a bar of varying size or by another object of the same class, that are difficult to handle with other existing class-specific top-down segmentation methods.

Keywords Shape-based object detection · Class-based top-down segmentation

L. Gorelick (✉)
Dept. of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel
e-mail: lena.gorelick@weizmann.ac.il

R. Basri
Toyota Technological Institute at Chicago, Chicago, IL 60637, USA

1 Introduction

We present a segmentation-based system for detection and figure-ground delineation. Unlike many other methods, see Sect. 2, which use appearance for detection, our method relies primarily on the shape of objects, as is reflected by their (bottom-up) segmentation in the image. Shape information, available in the image in the form of coherent image regions/segments and their bounding contours, provides a powerful cue that can be used for detection and segmentation of objects under a variety of lighting conditions and in the presence of significant deformations. While the robust extraction of adequate shape information for recognition still poses a considerable challenge, recent advances in recognition allow rapid examination of partial shape hypotheses. Statistical tests can then evaluate ensembles of such hypotheses and identify the presence of objects of interest in the image.

In order to utilize shape for recognition we need to identify three main components: (1) a method for extraction of shape hypotheses from images, (2) a representation of

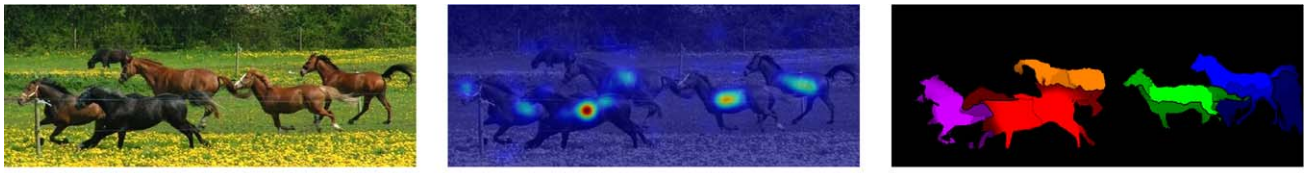

Fig. 1 (**a**) An input image, (**b**) its voting map (overlaid on the original input image), and (**c**) top-down segmentations for the detected object hypotheses. Each hypothesis is marked with a different color. The brighter the color is at a pixel the higher the confidence is in assigning the pixel to the foreground of that particular hypothesis

shapes to be applied to both database shape model and image shapes, and (3) a method for comparing these representations for recognition.

Image segmentation provides a natural means for extracting shape hypotheses. However, since in general existing image segmentation algorithms cannot reliably extract complete objects from an image, we rely on a hierarchical segmentation to provide overlapping partial shape hypotheses. Moreover, to account for over-segmentations, which may occur at intermediate levels of the segmentation hierarchy, our shape representation is designed to rely mainly on the significant (sharp) boundary sections of the segment. These are the locations in the underlying image where sharp transition in color/texture takes place and therefore they convey useful shape information.

We represent shape using two types of local descriptors. The first type encodes the boundaries of the shape. The second type is a regional descriptor, which includes a dense local orientation field derived from the shape by solving a Poisson equation on the shape. We augment this representation by storing in addition a color histogram with each shape.

These shape descriptors are used to compare the segments extracted from the image to a database of object silhouettes stored in memory. Partial matches then vote for potential object locations. Our voting scheme uses the principles outlined in Leibe et al. (2008), but utilizes segments of different sizes and shapes in the voting process. Once an object is detected, we propose a novel *competitive coverage* process to accurately delineate the boundaries of the object. In this process the voting segments compete for interpreting (covering) each of the semantic parts of an object. Our aim in this process is not only to identify the regions in the image that voted for the object, but also to resolve ambiguities that arise when two different regions are matched to the same model part and to discard nearby false regions that participated in the voting process.

Figure 1 shows an example of a challenging input image along with its voting map and top-down segmentations computed with our method. Note that this image contains multiple object instances, some of which are substantially occluded and/or significantly vary in scale.

We provide quantitative experimental results by applying our method to two challenging horse and skate-runner datasets. These experiments demonstrate that our method is capable of accurately detecting and segmenting objects with complex shapes even with as few as ten training examples. While obtaining results comparable to those of existing state of the art methods on images with a single object instance, our method allows us to simultaneously detect multiple instances of class objects in images and to cope with challenging types of occlusions that are difficult to handle with other existing class-specific top-down segmentation methods. To demonstrate this we perform further qualitative tests and show detection maps and top-down segmentations computed for a number of difficult out-of-dataset images containing multiple instances of horses and skate-runners and severe occlusions of various types (see discussion and figures in Sect. 4.3).

The remainder of the paper is divided as follows. Section 2 places our approach in the context of other related work. Our approach is described in detail in Sect. 3. Experimental results are shown in Sect. 4.

2 Related Work

The success of early work, which approached recognition using both independent collections and dependent constellations of local appearances (Burl et al. 1998; Weber et al. 2000; Ullman et al. 2001; Agarwal and Roth 2002), led to a surge of studies that explored different aspects of the problem. Here we only briefly mention a few. Common to most of this work is the use of local descriptors (e.g., raw intensity patches (Ullman et al. 2001; Vidal-Naquet and Ullman 2003; Agarwal and Roth 2002; Leibe et al. 2008), local filter responses (Felzenszwalb and Huttenlocher 2005; Pantofaru et al. 2008), local edge elements and/or their descriptors (Kumar et al. 2004; Mori et al. 2004; Shotton et al. 2005; Ren et al. 2005b; Opelt et al. 2006; Ferrari et al. 2007; Wang et al. 2007), and small regions obtained by over-segmentation (Ren et al. 2005a; Pantofaru et al. 2008; Cao and Fei-Fei 2007), and the reliance on a statistical model to

relate these descriptions to a coherent shape. Such an approach has the advantage that it requires only minimal preprocessing of the image, and so it is less sensitive to errors that may result from such preprocessing.

The position we take in this paper is that larger regions identified in the image as a result of a grouping process (bottom-up segmentation), can nevertheless provide elaborate shape information which is less affected by variations in the color, texture and lighting conditions present in the image and hence may be useful for recognition. Moreover, while segmentations essentially encode the same information as edge elements, they may give rise to sparser sets of boundaries whenever a relatively coarse segmentation is considered, allowing to reduce the number of object hypotheses to be evaluated in the detection stage. Finally, coherent image regions emerging from a bottom-up segmentation can be used by top-down processes to delineate foreground regions that are consistent with the boundaries in the image.

Several recent methods use bottom-up segmentations for discovering object and scene categories. The impressive method of Todorovic and Ahuja (2007) uses co-occurrence, similarity and clustering of hierarchical bottom-up segmentation sub-trees to unsupervisely learn the taxonomy of visual categories present in images. In the method of Wang et al. (2007) top-down object hypotheses are refined by aligning them to any feasible bottom-up segmentation of the input image. The method of Borenstein (2006) constructs a Bayesian model that integrates top-down with bottom-up segmentations to obtain figure-ground class-specific object segmentation. The method of Russell et al. (2006) uses multiple segmentations of each image to increase the chance of obtaining a few "good" segments that will contain potential objects, while each segment is assumed to be a potential object in its entirety. The method of Cao and Fei-Fei (2007) applies to a single level segmentation (over-segmentation) and uses the over-segmented pieces as building blocks for their hierarchical model. The methods of both Russell et al. (2006) and Cao and Fei-Fei (2007) describe image regions using visual "bag of words" (capturing mostly the appearance of objects) and rely on probabilistic latent semantic analysis (pLSA) to simultaneously recognize categories of objects in images. In contrast to that, our method does not rely on robustness and reproducibility of the segmentation structure for different objects of the same class, nor on extracting visual appearance features, but rather on dense regional shape representation of objects exemplars and the segments themselves. Such a representation is less affected by variations in the appearance of the objects belonging to the same class and is suitable for recognition of objects with distinctive shapes.

The system we present also combines and modifies several existing components:

- We use segmentations obtained with the method of Sharon et al. (2006). This method provides in one pass a complete hierarchical decomposition of the image into segments by applying a process of recursive coarsening, in which the same minimization problem is represented with fewer and fewer variables producing an irregular pyramid. During this coarsening process the algorithm generates and combines multiscale measurements of intensity contrast and texture differences, giving rise to regions that differ by fine as well as coarse properties with accurately located boundaries. We extend the segmentation algorithm above to cope with color images.

  It is worth noting, however, that our method is not limited to any particular type of segmentation algorithm as long as it is hierarchical.
- Poisson equation was proposed in Gorelick et al. (2006) to characterize the shape of pre-segmented *complete silhouettes*. The solution to the Poisson equation at a point internal to the silhouette represents the mean time required for a particle undergoing a random walk starting at a point to hit the boundary (see Sect. 3.1.2 and Fig. 5(b)). Such a representation allows to smoothly propagate the contour information of a silhouette to every pixel internal to the silhouette resulting in a dense shape representation. In practice, bottom-up segmentation algorithms often fail to extract complete object silhouettes. Therefore, unlike Gorelick et al. (2006), our method applies to *partial silhouettes* (shapes) formed by segments at intermediate scales of the bottom-up segmentation, possibly with *incomplete* boundaries. To reduce the effect of the missing boundary sections on the computed shape descriptors we modify the Poisson equation in Gorelick et al. (2006) by introducing mixed Dirichlet and Neumann boundary conditions (see Sect. 3.1.2).
- For detection we use probabilistic voting that follows the principles proposed in Leibe et al. (2008). Our voting scheme, however, differs in several important respects. First, as opposed to rectangular intensity patches in Leibe et al. (2008), our voting scheme applies to segments of different sizes and shapes, characterized by dense local shape properties. Second, the segments are matched to a database model, characterized by similar local shape properties, rather than a code book of rectangular appearance patches. Finally, segments emerging from a bottom-up segmentation provides a natural way for object delineation consistent with the boundaries present in the image.

The problem of class specific segmentations was addressed in several papers (Borenstein et al. 2004; Kumar et al. 2005; Winn and Jojic 2005; Levin and Weiss 2006; Wang et al. 2007; Borenstein 2006; Leibe et al. 2008), in many cases independent of detection. These studies further demonstrate the potential benefits of combining

bottom-up with top-down information. Some of these approaches (Borenstein et al. 2004; Borenstein 2006) use image segmentation to guide top-down figure-ground delineation. However, they rely primarily on intensity fragments for the detection phase. Once an object is detected (Borenstein et al. 2004; Borenstein 2006) seek consistent placements of the voting fragments with the results of bottom-up segmentation. Moreover, these approaches do not constrain the relative location of object parts in the detection phase. In contrast to these studies, our method does not use intensity patches but relies exclusively on shape descriptions obtained from the bottom-up segmentation. In addition, the relative location of parts plays a significant role in the detection phase.

Generally in studies that address the problem of class-specific image segmentation, image pixels are classified as either foreground or background based directly on the votes cast by patches containing the pixels (see, e.g., Leibe et al. 2008). Below we propose a new, top-down segmentation method by *competitive coverage*. In our scheme we let the voting segments compete for interpreting the same semantic area of an object. We then choose the most likely segment to explain each part, yielding sharper delineation than that obtained by directly applying (Leibe et al. 2008) to segments, (see discussion in Sect. 3.3 and examples in Sect. 4).

3 Approach

In the following section we describe our detection and the top-down segmentation approach.

The main components of our detection algorithm are shown graphically in Fig. 2. Our algorithm receives as input an image in (a), along with its hierarchical segmentation in (b). We use segmentations obtained with the method of Sharon et al. (2006). This method gradually merges segments according to criteria based on intensity and texture, providing segmentation maps at several scales containing segments of different shapes and sizes. We extend this method to cope with color images by applying all its arsenal of multiscale measurements in a transformed HSV color space (tHSV) defined by $(vs\cos h, vs\sin h, v)$ for a particular (h, s, v) value.

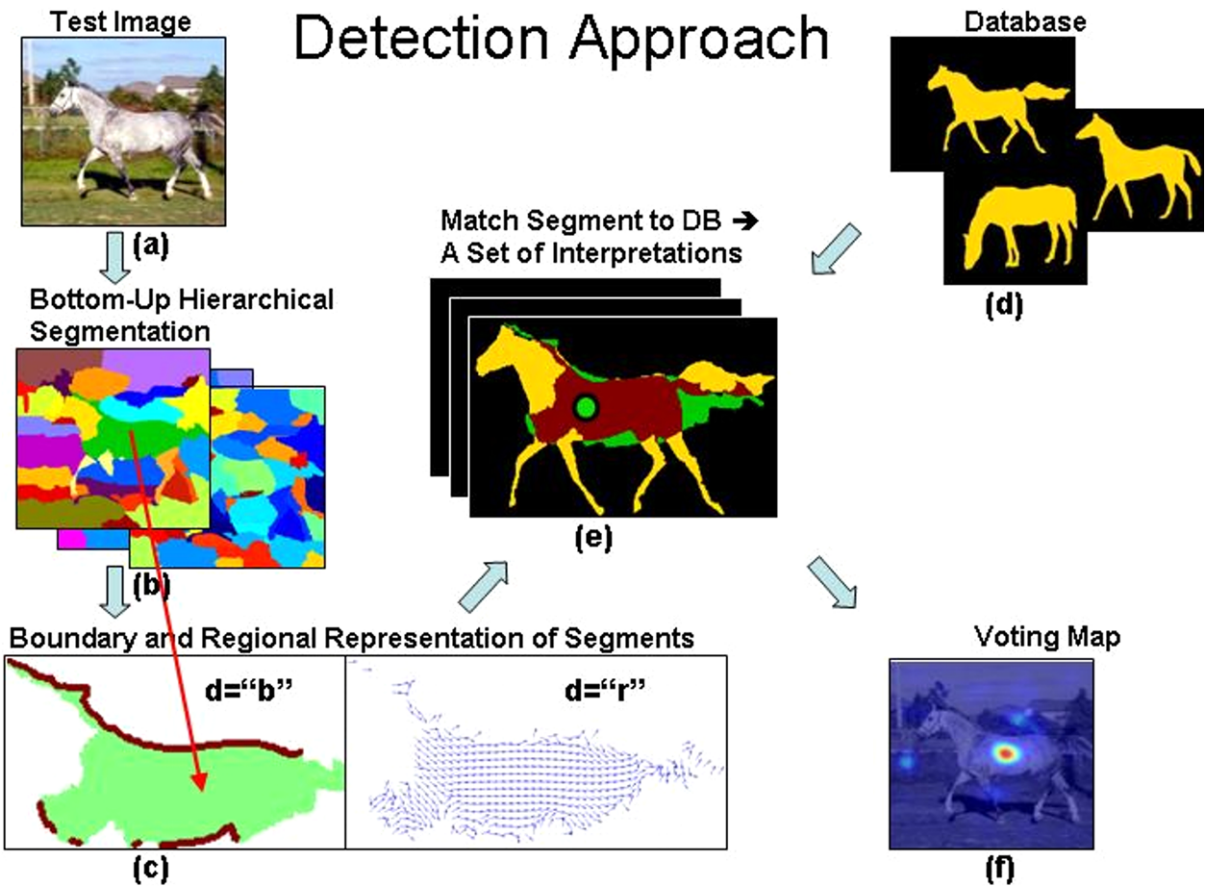

Fig. 2 Given an input image in (**a**), and its hierarchical segmentation in (**b**), each segment is matched to a database of object silhouette exemplars in (**d**) based on its shape information, namely, boundary and regional (interior) representation in (**c**) to obtain a set of segment interpretations in (**e**). Each interpretation then votes for a potential object location in a probabilistic framework, resulting in voting map in (**f**)

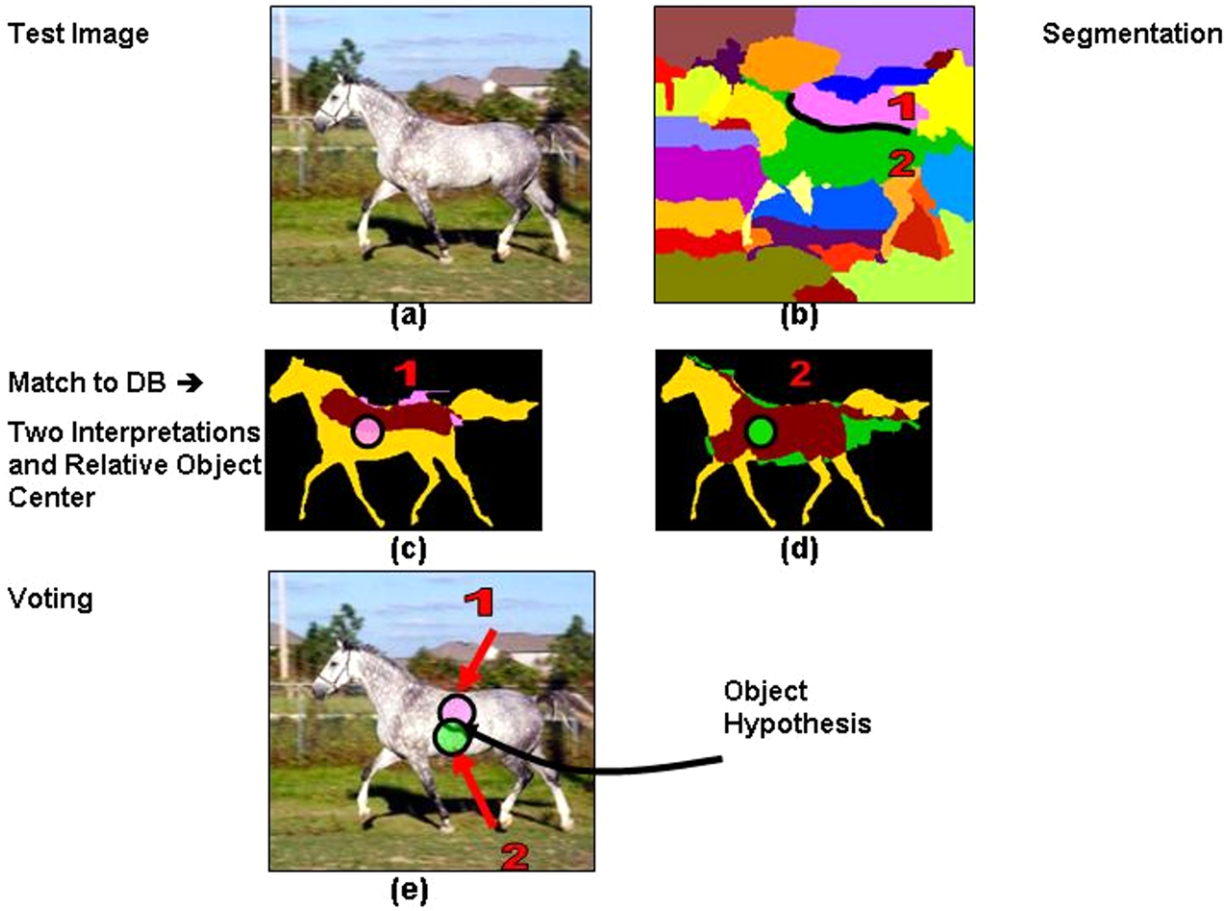

Fig. 3 Top-down, figure-ground segmentation: the need for a competitive coverage process. (**a**) An input image, (**b**) its hierarchical segmentation and an example of two neighboring segments (*pink* and *green* also marked by 1 and 2 for clarity) sharing a significant common boundary (marked in *black*). (**c**, **d**) The above two segments are matched to the database of silhouette exemplars to activate twointerpretations (both as horses's back) along with the relative object centers (marked by the *pink* and *green* circles accordingly). (**e**) A voting map based solely on the votes of the above two interpretations (marked by the *blue* and *green circles*). Both segments contributed to the hypothesis in the overlap area (marked by the *black arrow*) while being interpreted as horse's back, but the "real" horse in the input image has a single back

For each segment we then maintain its boundaries, while identifying the sections where sharp transitions in color or texture occur. These boundary sections convey useful shape information, and therefore are considered as significant or "sharp". Figure 2(c) shows an example of an image segment along with its boundary representation. The significant ("sharp") portions of its boundaries are marked in red. (See Sect. 3.1.1 for more details.) We further store regional properties computed by solving a Poisson equation on the segment. This descriptor assigns to each internal pixel the local orientation of the segment near that pixel. Figure 2(c) shows an example of a segment along with its regional representation (local orientation at every pixel is represented by small arrows). Note that all the body pixels are assigned with horizontal orientation of the body, while the pixels near the neck region show diagonal orientation of the neck. (See Sect. 3.1.2 for more details.) Using these properties we match each segment to a model of an object class represented by a collection of segmented silhouette exemplars stored in memory (shown in Fig. 2(d)), characterized by similar local shape properties, to activate a set of interpretations (shown in Fig. 2(e)). By interpretation we mean the effective part of a database model (silhouette) that was explained by the segment match, e.g., pixels of the back, leg or head lying in the overlap area (the segment is green, the database model is yellow and the overlap/match area is brown). Each activated interpretation augmented with the color information of its pixels votes for an object and its center location resulting in a voting map (shown in Fig. 2(f)). Object hypotheses are then found as local maxima in the voting space.

Finally, for a given object hypothesis we apply a pixel-level figure-ground top-down segmentation process by "competitive coverage" in which the voting segments com-

pete for interpreting each semantic part of the object. The need for this competition is illustrated in Fig. 3. Consider the input image in (a) and the two segments (the pink and green, also marked by 1 and 2) at the particular segmentation scale shown in (b). These two segments share somewhat similar shapes—they are both horizontally elongated, due to their significant common boundary (marked as black line). As a result, during the matching process, both segments might be matched to the same semantic part of the model, e.g., be interpreted as the horse's back (Figs. 3(c) and (d)). Now every such match will vote for an object location, and since the two segments are nearby, their votes will cast at nearby locations and will partly overlap (the pink and green circles, also marked by the red arrows 1 and 2 in Fig. 3(e)). In this overlap area, an object hypothesis obtains votes from both segments, each claims to interpret the horse's back. But the "real" horse in the given test image only has a single back. Our method identifies this situation and chooses the most likely segment to cover each semantic part of the model. It, therefore, resolves the ambiguities that arise when several different regions are matched to the same model part and discards false nearby regions that participated in the voting process. This is our proposed competitive coverage process (see Sect. 3.3).

The rest of this section describes our approach in detail. Section 3.1 explains the extraction of shape properties of the segments. Sections 3.2 and 3.3 describe our probabilistic formulation for object detection and competitive top-down segmentation respectively.

3.1 Representation

Our detection scheme relies on a matching process in which image segments are matched against silhouettes in the database. A good match is one in which a segment overlaps fairly closely with a portion of a silhouette. To evaluate a match we consider two types of representations: boundary and region, denoted by $d =$ "b" and $d =$ "r" respectively. The first one describes the fit between the boundaries of both the segment and the silhouette, and the second one compares a local shape descriptor at each of their internal pixels. Below we describe in detail the representations and how we judge the quality of a match.

3.1.1 Boundary Representation

The first representation we use consists of the segment boundaries. We distinguish between "sharp" (significant) and "soft" (insignificant) sections along the boundaries by measuring the local color contrast at every location along the boundaries of a segment (see Fig. 4).

For each boundary pixel **p** consider a disc of radius r centered at a pixel. Let V_{in} and V_{out} denote the sets of disc pixels falling inside and outside the segment respectively.

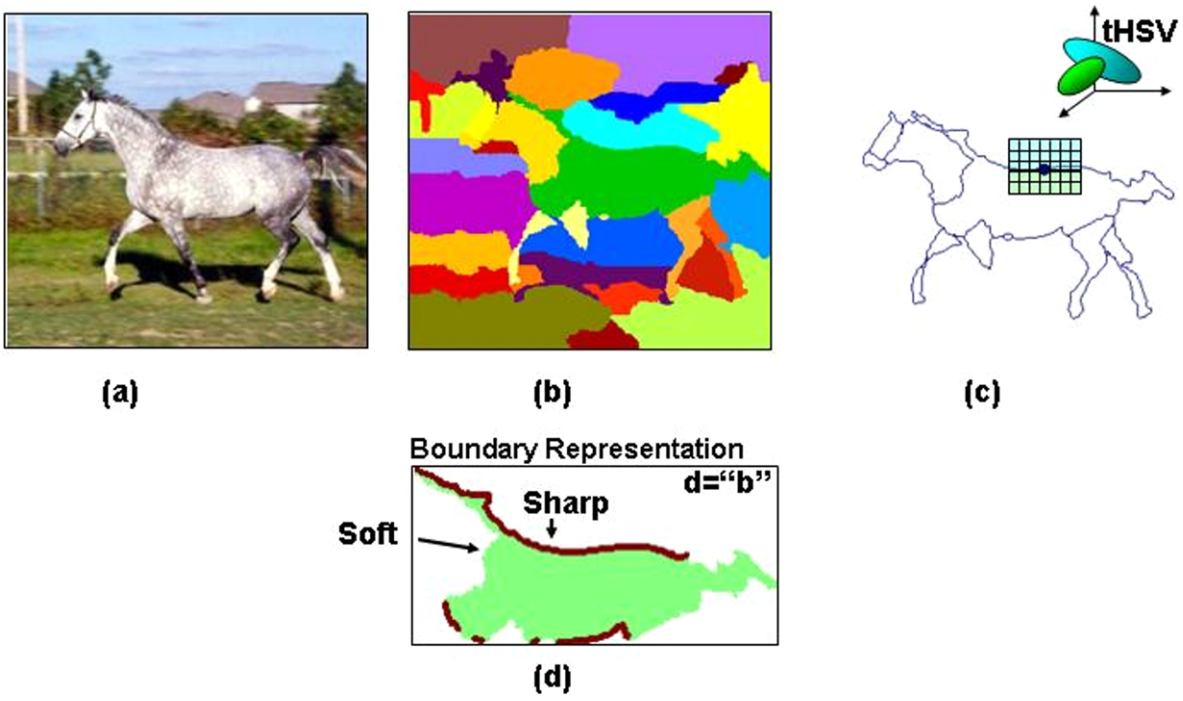

Fig. 4 (**a**) An input image, (**b**) its segmentation at one scale of the hierarchy, (**c**) at each boundary point we consider a rectangular patch and measure the symmetric Mahalanobis Distance in tHSV space between the inner and outer parts of the segment, (**d**) boundary representation, denoted $d =$ "b", for a particular segment (the *green segment* in (**b**))

Let C_{in} and C_{out} be the covariance matrices and μ_{in} and μ_{out} be the mean color vectors of the two sets in tHSV space respectively. We define the local contrast $M(\mathbf{p})$ at the pixel $\mathbf{p}$ to be the symmetric Mahalanobis distance between the color distributions across the segment boundary at that point. Namely,

$$M(\mathbf{p}) = 0.5\sqrt{(\mu_{out}-\mu_{in})^T C_{in}^{-1}(\mu_{out}-\mu_{in})} + 0.5\sqrt{(\mu_{in}-\mu_{out})^T C_{out}^{-1}(\mu_{in}-\mu_{out})}. \quad (1)$$

A sharp change of color (high symmetric Mahalanobis distance in the tHSV color space) between the two sides of a contour suggests that the segment boundary in this location coincides with a consistent edge separating two regions in the image. Conversely, gradual change (low symmetric Mahalanobis distance) in the color values implies uncertainty in the location of the segment boundary. The contour at this location may be arbitrarily affected by the isotropic nature of the segmentation algorithm. Neighboring regions obtained at a particular scale of the segmentation pyramid may still merge at higher levels if there is no sufficient contrast in color or texture present in their common boundary. We, therefore, allow only the "sharp" portions to affect the quality of a match and discard segments with low rate of sharp portion of the boundary (relatively to their boundary length) from participating in the matching and voting procedures. Note that the decision regarding the boundary type ("sharp" or "soft") is made separately for each pair of neighboring segments based on the average contrast (value of $M(\mathbf{p})$) along the boundary between them, and is the same for *all* the pixels belonging to their common boundary.

To compare the boundaries of an image segment with those of a training silhouette we use the one way Chamfer distance. Chamfer matching has proven a capable and efficient method for matching contours (Shotton et al. 2005). The one way Chamfer distance, which allows for partial matches, enables us to detect objects that are split into several segments and to overcome partial occlusion. Consider a training (model) silhouette $\mathbf{t}$ and a segment $\mathbf{s}$ placed in a particular position $\mathbf{x}$ relative to $\mathbf{t}$. (Throughout this manuscript we will use the superscripts $\mathbf{t}$ and $\mathbf{s}$ to denote, respectively, a training silhouette and an image segment.) Denote by $\chi_b^{\mathbf{t}}$ the characteristic function of the silhouette contour and by $\chi_b^{\mathbf{s}}$ the characteristic function of the "sharp" portions of the segment boundary. We define the boundary matching distance $D_{\text{boundary}}(\mathbf{x};\mathbf{s},\mathbf{t})$ as the one way Chamfer distance within a radius D between the boundaries of the segment $\mathbf{s}$ and the training silhouette $\mathbf{t}$ as a function of relative position $\mathbf{x}$,

$$D_{\text{boundary}}(\mathbf{x};\mathbf{s},\mathbf{t}) = \frac{1}{L_b^{\mathbf{s}}}\sum_{\mathbf{p}\in\chi_b^{\mathbf{s}}}\frac{\min(\min_{\mathbf{q}\in\chi_b^{\mathbf{t}}}\|(\mathbf{p}+\mathbf{x})-\mathbf{q}\|_2, D)}{D}, \quad (2)$$

where $L_b^{\mathbf{s}}$ is the total length of the sharp boundary sections in $\mathbf{s}$. This gives the mean distance of boundary pixels of $\mathbf{s}$ to their closest pixels in $\mathbf{t}$ as a function of displacement $\mathbf{x}$. Having the exact Euclidean *distance transform* ($DT^{\mathbf{t}}$) precomputed for every exemplar $\mathbf{t}$ in the database, (2) can be computed simultaneously for all possible superpositions of the segment $\mathbf{s}$ over the training silhouette $\mathbf{t}$ by a convolution.

3.1.2 Regional Representation

We use a second representation in which every pixel internal to a segment is associated with a local shape orientation vector and compared to a similar dense representation of a silhouette in the database. This orientation field is computed by solving a Poisson equation as in Gorelick et al. (2006). To make this paper self contained we briefly describe this approach here.

Consider a silhouette S surrounded by a simple, closed contour ∂S. We seek to represent the silhouette by assigning to every internal point a value that depends on its relative position within the silhouette. A common descriptor is the Distance Transform, which assigns every point within the silhouette a value reflecting its distance to the nearest boundary point and therefore is local in nature. An alternative approach that provides a more global description of shape is to place a set of particles at each internal point and let the particles move in a random walk until they reach the boundaries of the silhouette. We can then characterize the shape using various statistics of this random walk, such as the mean time required for a particle starting at a point to hit the boundaries. Denote this particular measure by $U(x,y)$. Then, $U(x,y)$ can be computed recursively as follows. At the boundary of S, i.e. $(x,y)\in\partial S$, $U(x,y)=0$. At every point (x,y) internal to S, $U(x,y)$ is equal to the average value of its immediate four neighbors plus a constant representing the amount of time required to get to an immediate neighbor, i.e.,

$$U(x,y) = 1+\frac{1}{4}\big(U(x+h,y)+U(x-h,y)+U(x,y+h)+U(x,y-h)\big). \quad (3)$$

We set this constant to one time unit. It is easy to show that (3) above is a discrete form approximation of the Poisson equation

$$\Delta U(x,y) = -\frac{4}{h^2}, \quad (4)$$

where $\Delta U = U_{xx}+U_{yy}$ denotes the Laplacian of U and $4/h^2$ is the overall scaling. For convenience we set $4/h^2=1$ and, therefore, solve

$$\Delta U(x,y) = -1, \quad (5)$$

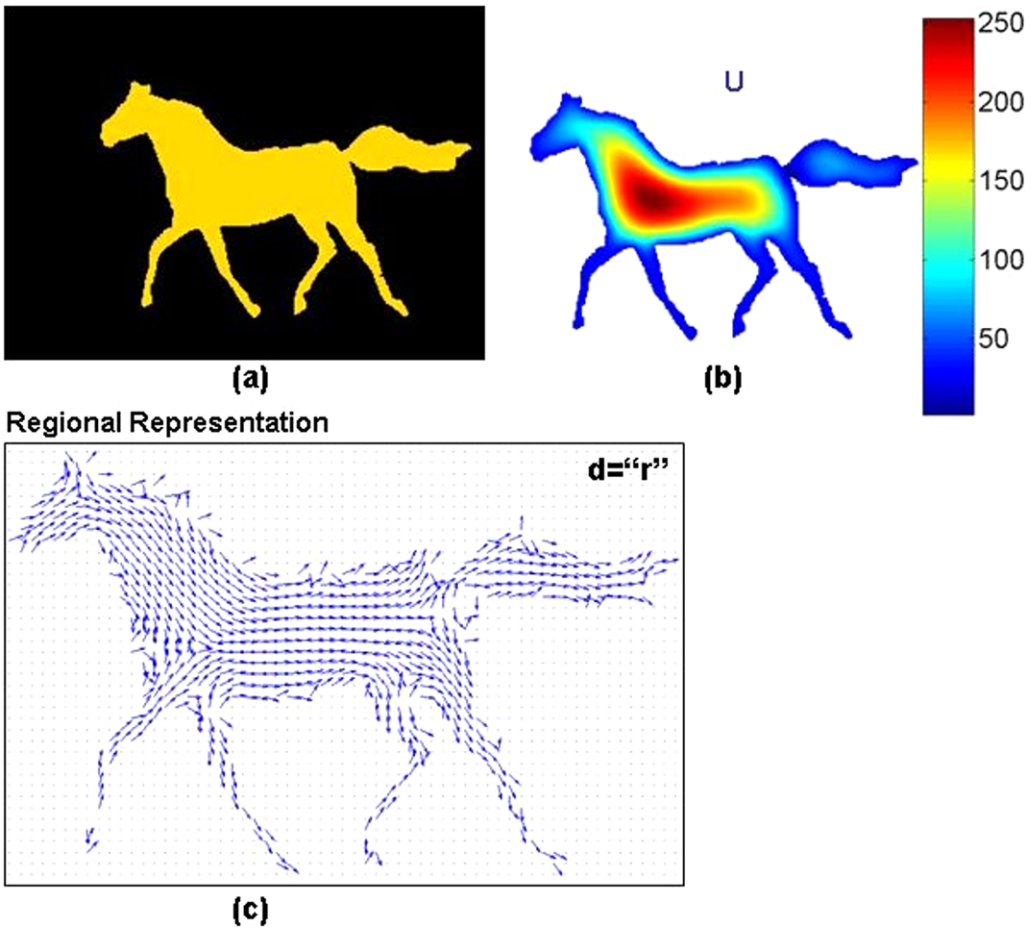

Fig. 5 Interior representation for a complete silhouette. (**a**) An example of a database silhouette, (**b**) the solution to the Poisson equation U and (**c**) the resulting regional representation $d =$ "r"—local shape orientation field (ranging from $-\pi/2$ to $\pi/2$ and shown as *small arrows*)

for $(x, y) \in S$, subject to Dirichlet boundary conditions $U(x, y) = 0$ at the bounding contour ∂S. The level sets of U represent smoother versions of the bounding contour and the contour information is smoothly propagated to every pixel internal the silhouette resulting in a dense shape representation (see Fig. 5(b) for an example of solution U obtained for a complete silhouette of a horse).

The solution U can be used to extract various shape properties of a silhouette, including its part structure and rough skeleton, local orientation and aspect ratio of different parts, and convex and concave sections of the boundaries (Gorelick et al. 2006).

As an example, consider a shape defined by a second order polynomial. Computing the Hessian matrix of U at any given point results in exactly the same matrix, which is in fact the second moment matrix of the entire shape, scaled by a constant. The eigenvectors and eigenvalues of H can then be used to infer the orientation of the shape, its aspect ratio, and will indicate whether the shape is elliptic or hyperbolic. For more general shapes the Hessian will vary continuously from one point to the next, but we can treat it as providing a measure of geometric moments of shape felt locally by the point. In this work we make use of such local shape orientation descriptors at every internal pixel of a silhouette.

For a database silhouette we solve the Poisson equation with zero Dirichlet boundary conditions as in Gorelick et al. (2006). Figure 5 shows an example of a database silhouette in (a), the solution to the Poisson equation U in (b) and the resulting regional representation $d =$ "r", namely, dense local shape orientation field in (c).

Unlike complete database silhouettes, image segments that emerge from the intermediate scales of the bottom-up segmentation may form only partial silhouettes with possibly incomplete (partially insignificant) boundaries. To reduce the effect of the insignificant boundary sections on the computed shape descriptors we solve the Poisson equation with Robin (mixed Dirichlet and Neumann) boundary conditions. Zero Dirichlet boundary conditions are used for the "sharp" boundary sections of the segment, while Neumann boundary conditions ($U_n(x, y) = 0$ with U_n denoting the derivative in the direction perpendicular to the bounding contour at (x, y)) are used for the "soft" boundary sections.

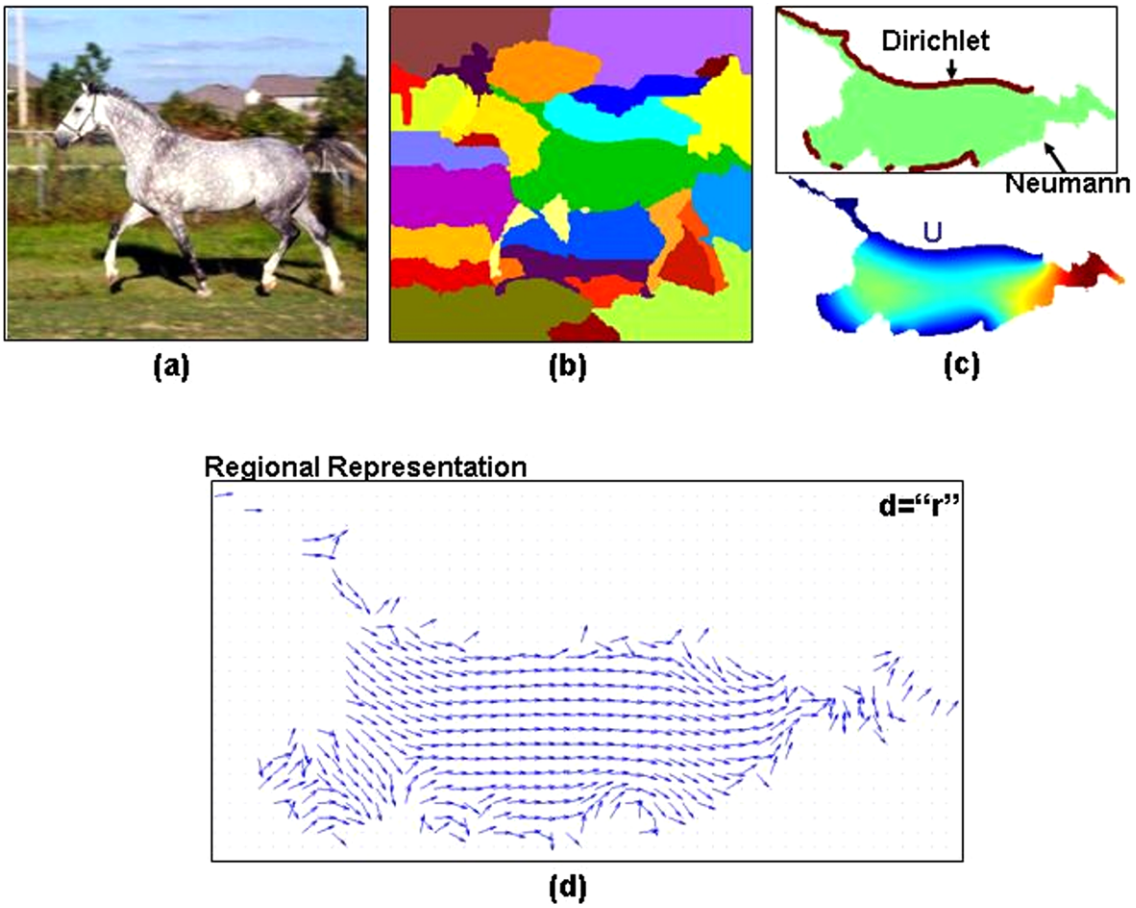

Fig. 6 Interior representation for an image segment. (**a**) An input image, (**b**) a particular scale of its bottom-up segmentation, (**c**) a silhouette of a particular (*green*) image segment with Dirichlet and Neumann boundary conditions introduced at "sharp" and "soft" sections of the segment boundaries respectively along with the solution to the Poisson equation U and (**d**) the resulting regional representation $d =$ "r"—local shape orientation field (ranging from $-\pi/2$ to $\pi/2$ and shown as *small arrows*)

The effect of the Neumann boundary conditions is that of a local mirror in space. If we assume that the "soft" boundary sections of a segment represent a common border with a missing part, then a reasonable assumption about the missing part is that U does not change much across the boundary. (Namely, the mean time to hit the complete shape contour does not change near the location of the "soft" boundary.) The level sets of the solution U represent smoother versions of the Dirichlet bounding contour, and are perpendicular to the Neumann boundary sections. The Fig. 6 shows an example of an input image in (a), a particular scale of its bottom-up segmentation in (b), a silhouette of an image segment with Dirichlet and Neumann boundary conditions introduced at "sharp" and "soft" sections of the segment boundaries respectively along with the solution to the Poisson equation U in (c) and the resulting regional representation $d =$ "r" in (d). Note, for example, that even though the silhouette of the segment is abrupt near the neck region, the correct (diagonal) shape orientation is preserved there.

As there might not be an exact analytical solution to the Poisson equation with the given boundary conditions, we solve it approximately in a least squares error (LSE) sense using a standard linear-time geometric multigrid solver (Trottenberg et al. 2001), modifying it to handle mixed boundary conditions through the use of Robin boundary conditions (Gustafson 1998), characterized by "Neumann coefficients" at the coarse boundary points, since at coarser levels single pixels may need to represent both types of boundaries simultaneously.

Next we match the interior region of the segment with the interior regions of the database silhouettes, using local shape orientation properties. Let $\theta^{\mathbf{s}}(\mathbf{p})$ and $\theta^{\mathbf{t}}(\mathbf{q})$ denote the orientations obtained at every pixel $\mathbf{p}$ of the segment $\mathbf{s}$ and every pixel $\mathbf{q}$ of the training silhouette $\mathbf{t}$ by solving the Poisson equation ($|\theta_i| \leq \pi/2$). By sliding the segment $\mathbf{s}$ over the training silhouette $\mathbf{t}$ we seek for a position in which the difference between the orientations of the matched pixels will be minimal, while the overlap between the segment and the silhouette will be maximal. Let $\chi_r^{\mathbf{s}}$ and $\chi_r^{\mathbf{t}}$ be the character-

istic functions of the segment $\mathbf{s}$ and the training silhouette $\mathbf{t}$ respectively, and let $\mathbf{x}$ be a displacement of $\mathbf{s}$ relative to $\mathbf{t}$. We define the region matching distance as follows,

$$D_{\text{region}}(\mathbf{x};\mathbf{s},\mathbf{t}) = \sum_{\mathbf{p}\in\chi_r^{\mathbf{s}}} d(\theta^{\mathbf{s}}(\mathbf{p}),\theta^{\mathbf{t}}(\mathbf{p}+\mathbf{x}))\cdot\chi_r^{\mathbf{s}}(\mathbf{p})\cdot\chi_r^{\mathbf{t}}(\mathbf{p}+\mathbf{x}) + \sum_{\mathbf{p}\in\chi_r^{\mathbf{s}}}\frac{1}{2}\cdot\chi_r^{\mathbf{s}}(\mathbf{p})\cdot\bar{\chi}_r^{\mathbf{t}}(\mathbf{p}+\mathbf{x}), \tag{6}$$

where $d(\theta_1,\theta_2)=0.5(1-\cos 2(\theta_1-\theta_2))$ ranges from zero (for a perfect match) to one (no match) and $\bar{\chi}_r^{\mathbf{t}}$ is the complement of the characteristic function.

The first term in (6) sums up the difference in orientation of matching pixels in the overlap area, and the second term counts the number of non-overlapping pixels of $\mathbf{s}$, assuming that non-overlapping portions match with $1/2$, the mean value of $d(\theta_1,\theta_2)$. Clearly, $D_{\text{region}}(\mathbf{x};\mathbf{s},\mathbf{t})$ depends on the size of a segment of interest, but it is comparable between the different matches of the same segment. The distance $D_{\text{region}}(\mathbf{x};\mathbf{s},\mathbf{t})$ can be computed simultaneously for all possible superpositions of the segment $\mathbf{s}$ over the training silhouette $\mathbf{t}$ using the identity $\cos 2(\alpha-\beta)=\cos(2\alpha)\cos(2\beta)+\sin(2\alpha)\sin(2\beta)$ by a convolution.

Note, that both our boundary and regional representations are determined by the edges of a segment, but they substantially differ in the type of information they describe. Our boundary representation is local in nature. It marks at every location the distance to the nearby edge element. Our regional representation, in contrast, provides a more global description of the shape, determined for each location through averaging random walk times to all boundary locations. Therefore, for instance, to conclude that a shape is elongated in a certain direction we need information about the presence of two adjacent image contours oriented in roughly the same direction. This information is not available in the boundary representation of the contours.

3.1.3 Interpretations of a Segment

Consider the collection of training silhouettes in the database superimposed one on top the other so that their centers are aligned. We refer to the coordinate system with the origin at their common center location as "*an abstract model*" or "*an abstract coordinate system*" of an object class.

Given a particular segment $\mathbf{s}$, it is matched against each of the training silhouettes in the database resulting in two sets of distance maps, namely $\{D_{\text{boundary}}(\mathbf{x};\mathbf{s},\mathbf{t}_k)\}_{k=1}^{K}$ and $\{D_{\text{region}}(\mathbf{x};\mathbf{s},\mathbf{t}_k)\}_{k=1}^{K}$ (with $\mathbf{t}_k$ denoting the k^{th} training silhouette in the database). We apply non-minimal suppression to each map with a radius proportional to the area of the segment. This results in a collection of preferred locations $\mathbf{x}_i$, $i=1\ldots N_b+N_r$ of the segment $\mathbf{s}$ relative to the object class abstract model, where N_b and N_r are the total number of selected peaks in the two sets of maps accordingly. We refer to each such preferred location as an interpretation and denote it with I_i (or I_{ij} when the segment is denoted by $\mathbf{s}_j$). We further assign a score to each interpretation, reflecting the quality of the match for that interpretation, which will be used as a voting weight.

The quality of a boundary based interpretation $I_i, i=1\ldots N_b$ depends on the relative length of the sharp portion of the segment boundary and its Chamfer distance to the boundary of the related exemplar in the database $\mathbf{t}_i$, namely,

$$Q_b(I_i)=\frac{L_b^{\mathbf{s}}}{L^{\mathbf{s}}}(1-D_{\text{boundary}}(\mathbf{x}_i;\mathbf{s},\mathbf{t}_i)), \tag{7}$$

where L^s is the total length of the segment boundary in pixels and $L_b^{\mathbf{s}}$ is the length of the sharp portion of the segment boundary. We further define the normalized score as

$$\hat{Q}_b(I_i)=\frac{Q_b(I_i)}{\sum_{l=1}^{N_b}Q_b(I_l)}. \tag{8}$$

The quality of a region based interpretation $I_i, i=N_b+1\ldots N_b+N_r$ depends on the similarity in the orientation of the matched pixels and the size of the explained portion of the segment, that is

$$Q_r(I_i)=\frac{1}{A^{\mathbf{s}}}\sum_{\mathbf{p}\in\chi_r^{\mathbf{s}}}\left(1-d(\theta^{\mathbf{s}}(\mathbf{p}),\theta^{\mathbf{t}_i}(\mathbf{p}+\mathbf{x}_i))\right) \times\chi_r^{\mathbf{s}}(\mathbf{p})\cdot\chi_r^{\mathbf{t}_i}(\mathbf{p}+\mathbf{x}_i), \tag{9}$$

where $A^{\mathbf{s}}$ denotes the area of the segment in pixels. Again, we define the normalized score as

$$\hat{Q}_r(I_i)=\frac{Q_r(I_i)}{\sum_{l=N_b+1}^{N_b+N_r}Q_r(I_l)}. \tag{10}$$

Note, that the boundary and region matching distances in (2) and (6) are used to find the preferred interpretations (local minima in the distance map) among the interpretations originating from the same training silhouette, whereas the quality scores of boundary and region interpretations in (8) and (10) are used to weight the interpretation relatively to interpretations originating from other training silhouettes.

3.2 Detection

In this and the next sections we provide a probabilistic formulation for the detection and top-down segmentation.

Our detection scheme follows the principles proposed in Leibe et al. (2008) while being reformulated to handle segments of varying shapes and sizes and to incorporate dense local shape properties into the voting process.

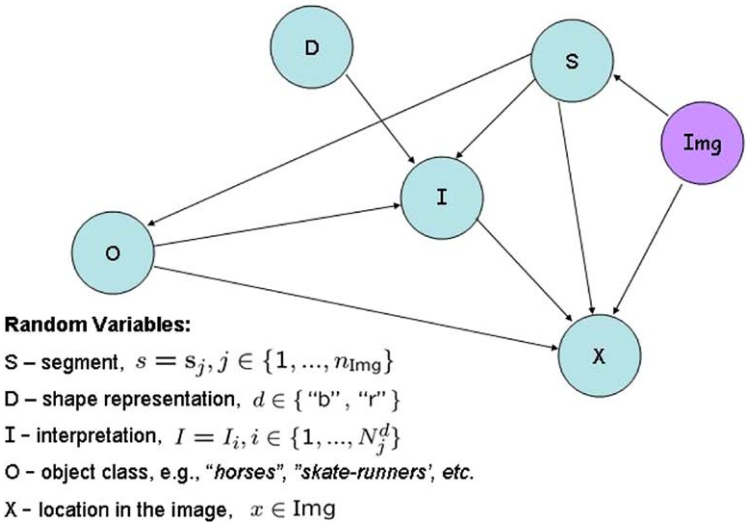

Fig. 7 The graphical model of the detection stage

Given a test image Img, consider a particular scale of the segmentation pyramid. Let $\mathbf{s}_j = (\mathbf{c}_j, \theta_j, l_j, \mathbf{b}_j)$ denote the color and orientation information associated with pixels of a segment j observed at location l_j along with its boundary information. Denote by s the random variable representing the index of an image segment, $s = \mathbf{s}_j, j \in \{1, \ldots, n\}$. Let d be the random variable corresponding to the considered shape representation, ($d \in \{b, r\}$ standing for boundary or region representation). Let x and o denote the random variables corresponding to the location in the image of the object of a certain class accordingly. See Fig. 7 for a depiction of the graphical model and a list of random variables along with their domains. (Note that the identity of the test image Img is observed, and the computations derived from the graphical model and described hereafter are all conditioned on it. For clarity we omit the conditioning throughout this section.) The joint probability of observing an object of the class o at location x given the segment s and the chosen representation d is defined by

$$\begin{aligned} p(o, x|s, d) &= p(o|s, d)p(x|s, o, d) \\ &= p(o|s)p(x|s, o, d) \end{aligned} \tag{11}$$

where we choose the first term, $p(o|s, d)$, to reflect our confidence that the segment s belongs to the class o based solely on its appearance (e.g., its color), and so it is independent of the chosen shape representation d. The second term, $p(x|s, o, d)$, is a probabilistic Hough vote for object position x based on the shape information of the segment.

For the term, $p(o|s)$, we learn a discrete probability distribution over the first two coordinates of a tHSV color space. Using segmented training data we construct two 100×100 bins color histograms providing the probabilities to observe a specific color $\mathbf{c}$ at pixels belonging to class and non-class respectively, namely, $p(\mathbf{c}|o)$ and $p(\mathbf{c}|\bar{o})$. Assuming a uniform prior for $p(o)$ and $p(\bar{o})$, the probability to observe an object class o given the color $\mathbf{c}$ is $p(o|c) = p(c|o)/(p(c|o) + p(c|\bar{o}))$. Finally, for the confidence $p(o|s)$ that a segment s belongs to a class based on its color we then use the mean of these probabilities over the pixels of s.

For the second term, $p(x|s = \mathbf{s}_j, o, d)$ in (11), we use the set of interpretations $\{I_i|d, o, s = \mathbf{s}_j\}$ obtained by matching the segment $\mathbf{s_j}$ to the silhouettes in the database using the shape representation d as described in Sects. 3.1.1 and 3.1.2. Given that s is a part of an object of class o, let the term $p(I = I_i|s = \mathbf{s}_j, o, d)$ reflect our confidence in interpreting the segment s by I using the shape representation d. This confidence is set to be the normalized quality of the match between the segment and the related training silhouette, $\hat{Q}_d(I)$ (see (8), (10)). Each interpretation then votes for possible center locations x of the object with probability $p(x|s, o, I, d)$ which is independent of the representation d chosen. We model this probability by a Gaussian distribution whose mean is determined by the relative location of the interpretation in the database silhouette and with constant variance σ (we used $\sigma = 8$). By marginalizing (11) over the set of interpretations we obtain

$$\begin{aligned} p(o, x|s, d) &= p(o|s)p(x|s, o, d) \\ &= p(o|s)\sum_I p(x, I|s, o, d) \\ &= p(o|s)\sum_I p(x|s, o, I, d)p(I|s, o, d). \end{aligned} \tag{12}$$

The single-segment votes are marginalized to obtain the probability of a hypothesis $h = (o, x)$,

$$\begin{aligned} p(o, x) &= \sum_{d\in\{b,r\}} p(o, x, d) \\ &= \sum_{d\in\{b,r\}} p(d)p(o, x|d) \end{aligned}$$

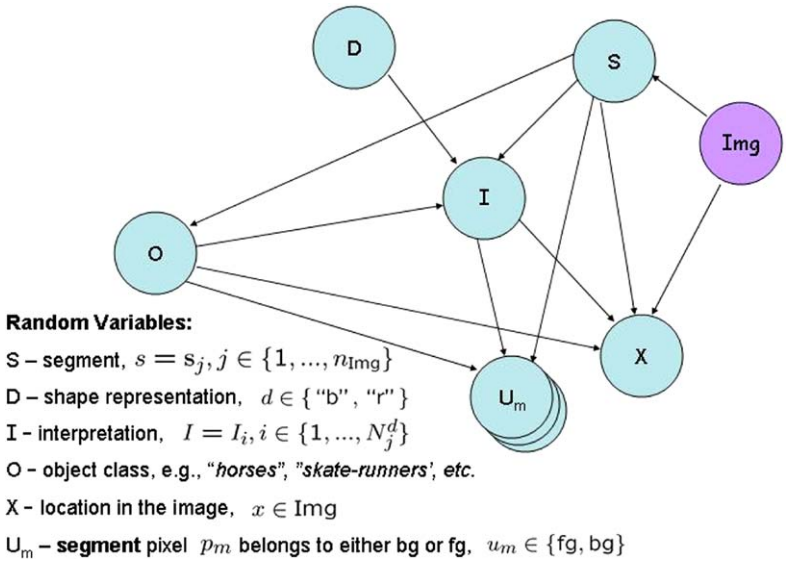

Fig. 8 The graphical model of the back-projection stage

$$= \sum_{d\in\{b,r\}} p(d) \sum_{s} p(o, x, s|d)$$

$$= \sum_{d\in\{b,r\}} p(d) \sum_{s} p(o, x|s, d) p(s|d) \tag{13}$$

with a uniform prior $p(d)$ and probability $p(s|d)$ set to be proportional to the size of the segment s relatively to other segments of the same scale in the segmentation pyramid. The probabilities $p(o, x)$ are computed simultaneously for all x resulting in a "voting map" (see Fig. 10, second row, column (a)). See Table 1 for a pseudocode of the detection algorithm.

3.3 Top-Down Segmentation

Given a detection hypothesis $h = (x, o)$ obtained with the bottom-up detection process, our next goal is to segment the object from its background.

One way to do this is to simply select all the segments that voted with significant probability for this hypothesis along with their interpretations. For each such interpretation I_i, it is possible to only consider the part of the segment $\mathbf{s}_j$ that is matched to the inside of the related database silhouette. Combining the information from all the interpretations of the segment would result in a probability map determining which pixels of the segment belongs to the foreground or the background. This, *back-projection* approach is similar to the top-down segmentation proposed in Leibe et al. (2008). However, it does not distinguish between correct and erroneous votes of different segments that accidentally matched to (or were interpreted by) the same semantic part of the model (e.g., the correct votes of the green segment vs. the erroneous votes of the pink segment shown in Fig. 3). Therefore, such a naive back-projection approach may give rise to a "ghosting effect," whereby the segmentation results include, e.g., "extra" back or limbs, see also Fig. 10, bottom row, column (b). To address this problem we introduce a *competitive coverage* approach, in which we let the segments compete for those parts.

We begin by formulating the back-projection approach.

3.3.1 Segmentation by Back-Projection

Our aim in this part is to identify the pixels that voted strongly to the object in its considered location. Given an object hypothesis $h = (o, x)$ and segment $s = \mathbf{s}_j$, denote by u_m the event that a segment pixel $p_m \in \mathbf{s}_j$ belongs to the foreground object. The relevant graphical model is shown in Fig. 8. The probability of this event is given by

$$p(u_m|o, x, s) = \sum_{d\in\{b,r\}} p(u_m, d|o, x, s)$$

$$= \sum_{d\in\{b,r\}} p(d|o, x, s) p(u_m|o, x, s, d) \tag{14}$$

$$= \sum_{d\in\{b,r\}} p(d) \frac{p(u_m, o, x|s, d)}{p(o, x|s, d)}. \tag{15}$$

The first transition is a simple marginalization over the possible shape representations. Assuming the shape representation d is independent of the class object o, position x and segment s, the last transition follows from the Bayes' rule.

Consider the numerator $p(u_m, o, x|s, d)$ in (15). This is the joint probability for the hypothesis $h = (o, x)$ and for the segment pixel p_m to belong to the foreground of the object, given the segment $s = \mathbf{s}_j$ and using the shape representation d. To compute it, we marginalize over those interpretations of s in which p_m participated as foreground pixel (i.e., was matched to the inside of the related database silhouette) to obtain:

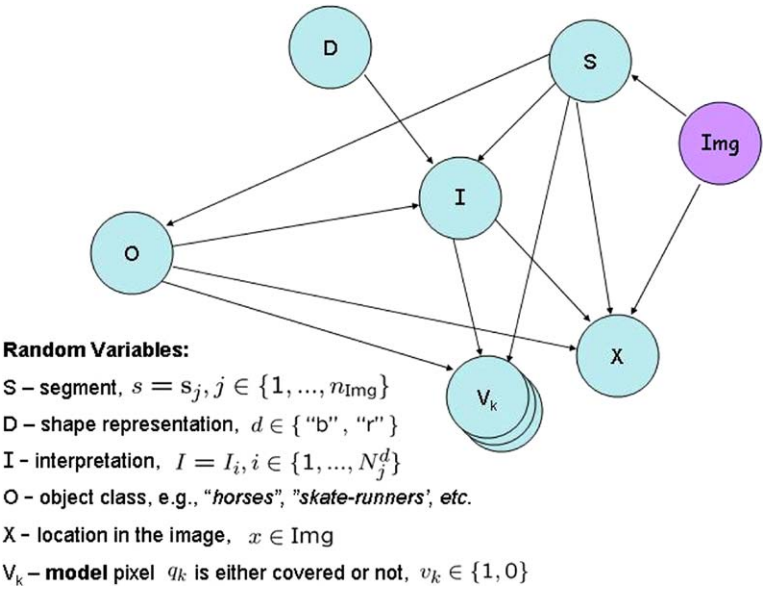

Fig. 9 The graphical model of the competitive-coverage stage

$$p(u_m, o, x|s, d)$$

$$= p(o|s, d)p(u_m, x|s, o, d) \tag{16}$$

$$= p(o|s)\sum_I p(u_m, x, I|s, o, d) \tag{17}$$

$$= p(o|s)\sum_I p(x|s, o, I, u_m, d)p(u_m|s, o, I, d) \times p(I|s, o, d) \tag{18}$$

$$= p(o|s)\sum_I p(x|s, o, I)p(u_m|s, o, I)p(I|s, o, d). \tag{19}$$

The term $p(o|s, d)$ in (16) depends solely on the appearance of $\mathbf{s}_j$ and is independent of the shape representation d. The transition to (17) is due to marginalization of the interpretations I of the segment s. The term $p(x|s, o, I, u_m, d)$ in (18) is the probabilistic Hough vote of the interpretation I for an object location. It is dependent on the location of the segment l and the interpretation I, but is independent of a particular pixel u_m and the representation d chosen. The term $p(u_m|s, o, I, d)$ in (18) is one if p_m is explained by the interpretation I (i.e., matches to the inside of the related database silhouette) and vanishes otherwise. It is, therefore, independent of the shape representation d as well. Finally, the last term $p(I|s, o, d)$ in (19) is the normalized matching quality, $\hat{Q}_d(I)$, computed as in (8) and (10).

The collection of the probabilities $p(u_m|o, x, s = \mathbf{s}_j)$ in (14), computed $\forall p_m \in \mathbf{s}_j$ forms the "back-projection" map of the segment. Figure 10, second row, column (b) shows an example of back-projection map simultaneously for all the segments of a particular segmentation scale.

3.3.2 Segmentation by Competitive Coverage

To incorporate competition in the process we distinguish between the segment pixels p_m, which may or may not belong to the foreground object o in the image, and the pixels in an *abstract* model of an object (defined in Sect. 3.1.3), which may or may not be covered (explained) by the image pixels. We denote by $\mathbf{k} = (x_k, y_k)$ the coordinates of a pixel q_k in an *abstract* model of an object and seek the most probable cover of the model pixel q_k by an image segment. For a given hypothesis $h = (o, x)$, denote by v_k the event that hypothesis h covers the pixel q_k. The graphical model of this process is illustrated in Fig. 9. The probability that (among all the segments) a segment $s = \mathbf{s}_j$ covers model pixel q_k can be written as

$$p(s|o, x, v_k) = \sum_{d\in\{b,r\}} p(d|o, x, v_k)p(s|o, x, v_k, d) \tag{20}$$

$$= \sum_{d\in\{b,r\}} p(d)\frac{p(v_k, o, x|s, d)p(s|d)}{p(v_k, o, x|d)}, \tag{21}$$

where the first transition is a marginalization over the shape representation d and the second transition follows Bayes' rule. Here again, we assume the shape representation d is independent of the class object o, position x and pixel v_k.

The term $p(v_k, o, x|s, d)$ in the numerator of (21) reflects the joint probability of the hypothesis $h = (o, x)$ and the event that the model pixel q_k is covered by the segment s using the shape representation d. Again, we sum only over those interpretations of the segment s that explain the model pixel q_k, (i.e., have the pixel q_k of the related database silhouette in the overlap area),

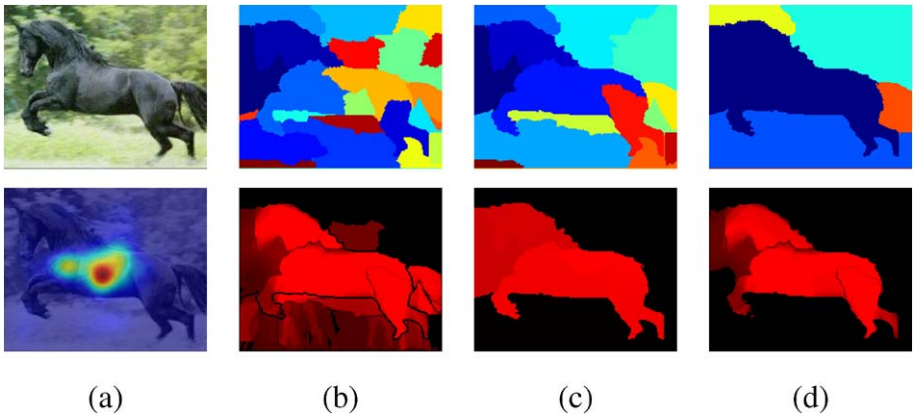

Fig. 10 *Top*: an input image and its three segmentation maps of scales 8–10. *Bottom*: (**a**) voting map, (**b**) back-projection map (14), (**c**) total expected volume of model pixels covered by each segment (26), (**d**) top-down segmentation map (27). (*Bottom*, (**b**–**d**) are coded with the intensity of the *red* color)

$$
\begin{aligned}
&p(v_k, o, x|s, d) \\
&\quad = p(o|s, d)p(v_k, x|s, o, d) \qquad (22)\\
&\quad = p(o|s)\sum_I p(x|s, o, I, v_k, d)p(v_k|s, o, I, d) \\
&\qquad \times p(I|s, o, d) \qquad (23)\\
&\quad = p(o|s)\sum_I p(x|s, o, I)p(v_k|s, o, I)p(I|s, o, d). \qquad (24)
\end{aligned}
$$

The first term $p(o|s, d)$ in (22) depends solely on the appearance of the segment s and is independent of the representation d. The term $p(x|s, o, I, v_k, d)$ in (23) is the probabilistic Hough vote of the interpretation I for an object location. It is dependent on the location of the segment l and the interpretation I but is independent of a particular model pixel q_k and the representation d chosen. The term $p(v_k|s, o, I, d)$ in (23) is one if the model pixel q_k is explained by the interpretation I and vanishes otherwise. It is, therefore, independent of the shape representation d as well. Finally, the last term $p(I|s, o, d)$ in (23) is the normalized matching score $\hat{Q}_d(I)$ computed as in (8) and (10).

Having the notion of an abstract object model allows us to ask the following question: "Given the input image and a particular object hypothesis $h = (o, x)$ what are the parts of the model (i.e., pixels in the abstract coordinate system) that are more likely to be present in the image?". We, therefore, compute for each model pixel q_k the posterior probability that it is covered by *any* of the input image segments given the hypothesis $h = (o, x)$ as follows:

$$
\begin{aligned}
p(v_k|o, x) &= \frac{p(v_k, o, x)}{p(o, x)} \\
&= \sum_{d\in\{b,r\}} p(d)\frac{p(v_k, o, x|d)}{p(o, x)} \\
&= \sum_{d\in\{b,r\}} p(d)\sum_s \frac{p(v_k, o, x|s, d)p(s|d)}{p(o, x)}, \qquad (25)
\end{aligned}
$$

where the term $p(s = \mathbf{s}_j|d)$ is set to be proportional to the size of the segment $\mathbf{s}_j$ relatively to other segments of the same scale in the segmentation pyramid. The first transition follows Bayes' rule, the second transition is a marginalization over the representation d, and the third transition is a marginalization over the image segments.

Figure 11 illustrates the information content that can be captured by taking $p(v_k|o, x)$ into account. Two input images are shown and compared: (a) an image containing a complete figure of a horse and (b) an image containing a horse occluded by a bar. The posterior probabilities of the model pixels to be covered by any of the image segments are shown in (c) and (d). As opposed to the result shown in (c), the central pixels of the abstract model in (d) have low probability to be covered by any of the image segments of the input image in (b) since they represent the pixels that are occluded by the bar.

Our next aim is to choose the most likely segment among the ones competing over the same semantic area. This can be done, for example, by computing for each segment the "volume" of model pixels covered by the segment. Consider a pixel q_k of an abstract model. We say that it has a volume of one, if it is covered by any of the image segments and zero otherwise. (Then, $p(v_k|o, x)$ can be seen as the expected volume of the pixel q_k.) Then, given a segment $\mathbf{s}_j$ the "expected total volume," denoted $E[V(\mathbf{s}_j)]$, of model pixels covered by $\mathbf{s}_j$ is defined as

$$
E[V(\mathbf{s}_j)|o, x] = \sum_k p(s = \mathbf{s}_j|v_k, o, x)p(v_k|o, x). \qquad (26)
$$

We assign as foreground those segments that cover a large volume of the model (relative to their area). Thresholding $E[V(\mathbf{s}_j)|o, x]/|\mathbf{s}_j|$ will result in assigning each *whole* segment to either the foreground or background based on top-down considerations, i.e., segment-level top-down object delineation. See example in Fig. 10, bottom row, column (c). Finally, since segments may contain both foreground and background portions we wish to separate the background portions from the selected segment. To that end,

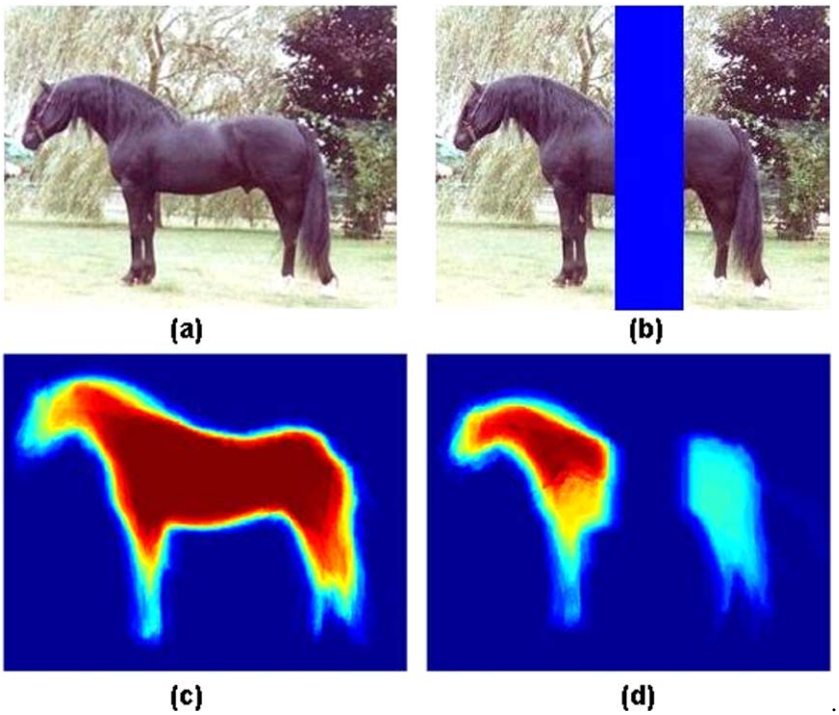

Fig. 11 An abstract model: (**a**) input image of a horse, (**b**) input image of a horse occluded by a bar, (**c**), (**d**) the posterior probability that pixels in the abstract model are covered by any of image segments as in (25) (the values range between zero and one, jet colormap in Matlab). Note that (**c**) and (**d**) are in the coordinate system of the abstract model and not the input images

we further refine the segment-level delineation by combining it with the back-projection map. We therefore define our final top-down map by

$$T(p_m|o,x) = \frac{E[V(\mathbf{s}_j)|o,x]}{|\mathbf{s}_j|} p(u_m|o,x,s=\mathbf{s}_j), \tag{27}$$

where $p_m \in \mathbf{s}_j$ (see Fig. 10, bottom row, column (d)). Notice in particular the segment containing the head which is classified as foreground using (26) and is split into two areas of distinct probabilities in our final competitive coverage derivation (27). Note also the marked improvement over the traditional, back-projection approach in which various background areas are assigned high probabilities. Most of these probabilities are sharply reduced by our competitive coverage process. Additional examples of top-down segmentation by competitive coverage are shown in Sect. 4. Applying a threshold we can obtain a pixel-level top-down object segmentation. See Table 1 for a pseudocode of the detection and top-down delineation algorithms.

4 Experiments and Results

We have tested our method on two challenging datasets, the Weizmann Horse Dataset (Borenstein 2009) and a Skate-Runner Dataset. The horse dataset consists of side-view images of horses and was investigated in Borenstein et al. (2004), Shotton et al. (2005), Levin and Weiss (2006), Winn and Jojic (2005), Kumar et al. (2005). While Borenstein et al. (2004), Levin and Weiss (2006), Winn and Jojic (2005), Kumar et al. (2005) concentrate their efforts mostly on segmentation of the detected objects, Shotton et al. (2005) achieves impressive detection results using contour-based learning. Both datasets have a very high within-class variation in shape, color, and texture and present challenging classes of objects for testing our approach. Note, that while the background in the skate-runner dataset is mostly white, the colorful patterns on the runners' outfits are highly cluttered and the highlights on the ice are highly reflective.

4.1 Detection

We first evaluated the performance of our detection algorithm. During training we computed the Poisson equation for each of the training exemplars and marked object center at the center of mass of the area defined by high values of U (higher than $0.6 \max_{(x,y)\in\mathbf{t}} U(x,y)$). The test set consisted of the remaining images along with the three scales (8–10) of the hierarchical segmentation pyramid. We let each scale of the segmentation pyramid participate in the voting process and combine the resulting voting maps by taking the average at every pixel, giving slightly more weight to the intermediate scale. We then quantify our detection results in terms of recall-precision curve (RPC), where recall (the proportion of the horses in the test set that are detected) is plotted against precision (the proportion of correctly detected horses out of total number of detections) as the global detection threshold is varied. Ground truth object locations and segmentations are known, and a detection is marked as correct when a peak falls within the central part

Table 1 Pseudocode summarizing the detection and top-down delineation steps of the algorithm

Detection Algorithm:

1. For each segment $s \in \{\mathbf{s}_j\}$ in the test image
   - 1.1. Extract boundaries: sharp(s), soft(s)← computeBoundaries(τ) (1)
   - 1.2. If length(sharp(s)) $< \rho$ (length(sharp(s) + soft(s))) → skip and goto 1
   - 1.3. Compute class prob. based solely on appearance: $p(o|s)$ (11)
   - 1.4. For representation $d \in \{b, r\}$
     - 1.4.1. Match s to DB using d to result in set of interpretations $\{I_i\}$ (2) and (6)
     - 1.4.2. For each interpretation $I \in \{I_i\}$
       - 1.4.2.1. Compute $p(I|s, o, d)$ (8) and (10)
       - 1.4.2.2. Compute location vote $p(x|s, o, I, d)$ with uncertainty σ (12)
     - 1.4.3. Compute $p(o, x|s, d)$ (12)
2. Compute final voting map p(o,x) (13)

Top-Down Algorithm: Given an object hypothesis $h = (o, x)$

1. For each segment $s \in \{\mathbf{s}_j\}$ in the test image
   - 1.1. Extract boundaries: sharp(s), soft(s)← computeBoundaries(τ) (1)
   - 1.2. If length(sharp(s)) $< \rho$ (length(sharp(s) + soft(s))) → skip and goto 1
   - 1.3. Compute class prob. based solely on appearance: $p(o|s)$ (11)
   - 1.4. For representation $d \in \{b, r\}$
     - 1.4.1. Match s to DB using d to result in set of interpretations $\{I_i\}$ (2) and (6)
     - 1.4.2. For each interpretation $I \in \{I_i\}$
       - 1.4.2.1. Compute $p(I|s, o, d)$ (8) and (10)
       - 1.4.2.2. Compute $p(u_m|s, o, I)$ (19), $p(v_k|s, o, I)$ (23)
       - 1.4.2.3. Compute location vote $p(x|s, o, I, d)$ with uncertainty σ (19)
     - 1.4.3. Compute $p(u_m, o, x|s, d)$ (16)
     - 1.4.4. Compute $p(v_k, o, x|s, d)$ (22)
   - 1.5. Compute segment back-proj. $p(u_m|o, x, s)$ (14)
   - 1.6. Compute segment competition $p(s|o, x, v_k)$ (20)
2. Compute posterior cover $p(v_k|o, x)$ (25)
3. For each segment $s \in \{\mathbf{s}_j\}$
   - 3.1. Compute expected volume $E[V(\mathbf{s}_j)|o, x]$ (26)
   - 3.2. Compute final top-down $T(p_m|o, x)$ (27)

Remark: τ is the average Mahalanobis distance threshold. We used $\tau = 6.5$, $\rho = 2/3$ and $\sigma = 8$

of the horse torso. Specifically, if the value of the solution U to the Poisson equation (computed for the silhouette) at the peak is higher than $0.6 \max_{(x,y)\in\mathbf{t}} U(x, y)$ or if the peak falls within a radius of 35 pixels from the correct centroid location stored in the database (a circle with a radius of 35 pixels corresponds to 30% of the average object size). (If two peaks fall in this area, only one is counted as correct.)

4.1.1 Weizmann Horse Dataset

For the horse dataset the training consisted of the first 50 images of the database, along with their manually segmented silhouettes, and the test set consisted of the remaining 277 images. 250 out of 277 (90%) images had the highest peak at the location of the correct centroid. We show recall-precision curves in Fig. 12(left) comparing different setting of our system: using only region representation (Region, green curve), boundary representation (Boundary, red), and both (r + b, black). The full model gives a recall-precision equal error rate of 85.8%. Partial models, i.e., based on region representation or on boundary representation achieve 76.1% and 82.4% respectively. The additional two curves portray the performance of our system when the number of training exemplars is varied: 20 (magenta) and 10 (blue) exemplars, obtaining ERR of about 85.3% and 83.7% respectively, showing only minor deterioration.

Our results are somewhat inferior to Shotton et al. (2005) who obtained a detection rate of 92%. However, Shotton et al. (2005) applied their method to a low-resolution (3.5 times as small) version of the same database, but with detection radius of 25 pixels, corresponding to 54% of the average object size. This is in comparison to 35 pixels in our higher resolution implementation, corresponding to 30% of the average object size. Our EER increases to 88.4% when we increase the detection radius in our implementation to 47 pixels, which corresponds to 54% of the average object size in our higher resolution database. Nevertheless, inspecting our results we noticed that failures often occurred in cases that segmentations given as input were poor. To verify this, we analyzed the segmentations used as input separately for two sets of images. The first one (TD) consisted of test images in which the highest peak was marked as correct, (re-

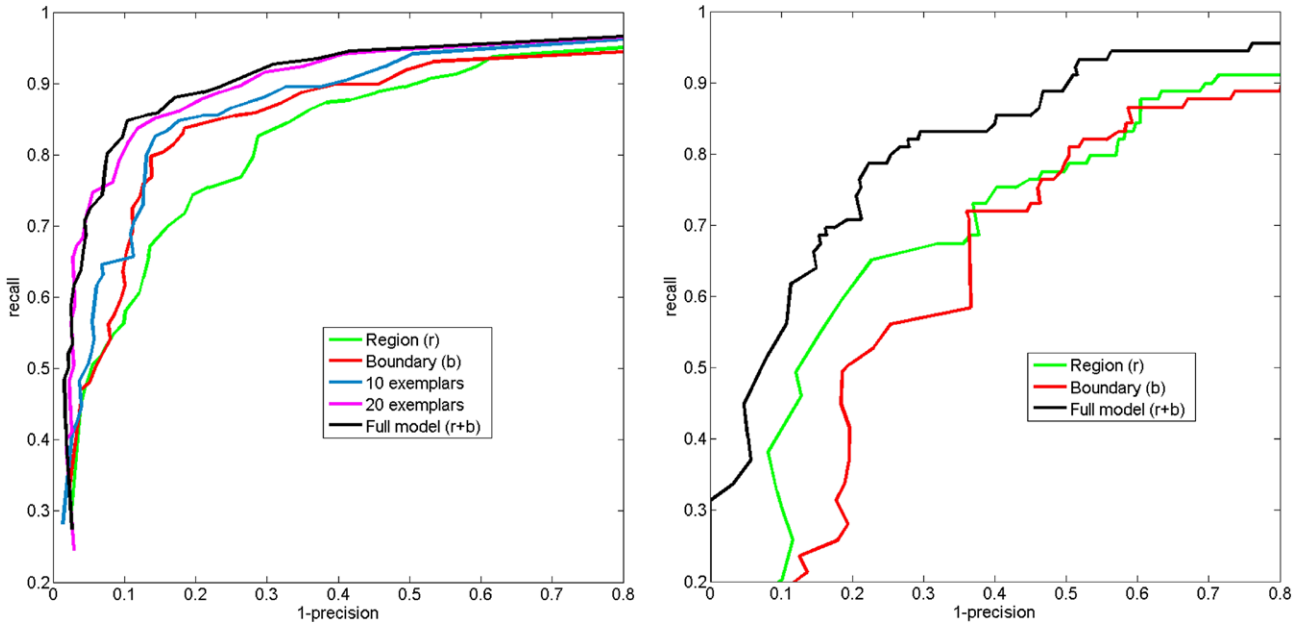

Fig. 12 Recall precision curves: the contribution of different aspects of our model and the effect of the training size on the performance, measured for the horse dataset (*left*) and skate-runner dataset (*right*)

sulting in 250 images). The second set (FD) consisted of images with undetected horses, resulting in 27 images. For each image and scale we considered all the segments that at least 90% of their area overlaps with the true silhouette or that covers at least 10% of the true silhouette. We then measured the recall (rc.) and precision (pr.) rates of the true boundary covered by the boundaries of these segments (we allowed up to 5 pixel deviation for these measures), and the portion of the segments' boundaries that fell inside (ib.) and outside (ob.) of the true silhouette. Finally, to evaluate the amount of leaking, for the segments that cover at least 10% of the true silhouette and at least the same area outside the true silhouette we measured the area portion that fell outside the true silhouette (oa.). These measures are summarized in Table 2. All these measures indeed indicate consistently that on average the quality of the segmentation is superior for the images with correctly detected horses than for the images with undetected horses. This suggests that detection performance as well as the top-down segmentations can further be improved if better segmentations are available.

4.1.2 Skate-Runner Dataset

The training consisted of the first 20 images of the database along with their manually segmented silhouettes, and the test set consisted of the remaining 89 images. 75 out of 89 (84%) images had the highest peak at the location of the correct centroid. We show recall-precision curves for this dataset in Fig. 12 (right), again comparing different setting of our system: using region representation (Region, green curve), boundary representation (Boundary, red) and both (r + b, black). The full model gives a recall-precision equal error rate of 78.1%. Partial models, i.e., based on region representation or boundary representation achieve 67.4% and 63.5% respectively.

Table 2 Quality of segmentation measured separately for the two sets of test images: (TD) consisting of 250 test images with correct detections, and (FD) consisting of 27 images with undetected horses

	Scale 8	Scale 9	Scale 10
TD avg. rc.	0.76 ± 0.009	0.76 ± 0.009	0.74 ± 0.009
FD avg. rc.	0.70 ± 0.029	0.69 ± 0.023	0.63 ± 0.027
TD avg. pr.	0.60 ± 0.007	0.64 ± 0.010	0.63 ± 0.013
FD avg. pr.	0.51 ± 0.015	0.52 ± 0.023	0.44 ± 0.031
TD avg. ib.	0.29 ± 0.005	0.20 ± 0.006	0.12 ± 0.005
FD avg. ib.	0.34 ± 0.015	0.26 ± 0.018	0.19 ± 0.019
TD avg. ob.	0.04 ± 0.004	0.11 ± 0.008	0.22 ± 0.011
FD avg. ob.	0.09 ± 0.015	0.17 ± 0.027	0.32 ± 0.035
TD avg. oa.	0.01 ± 0.002	0.09 ± 0.010	0.3 ± 0.024
FD avg. oa.	0.08 ± 0.008	0.13 ± 0.024	0.51 ± 0.076

4.2 Top-Down Segmentation

Next we evaluated our top-down segmentation algorithm. For this task we only used the test images with correct highest peak, resulting in 250 images for the horse dataset and 75 images for the skate-runner dataset. For all the images, ground truth segmentations were available, which allowed

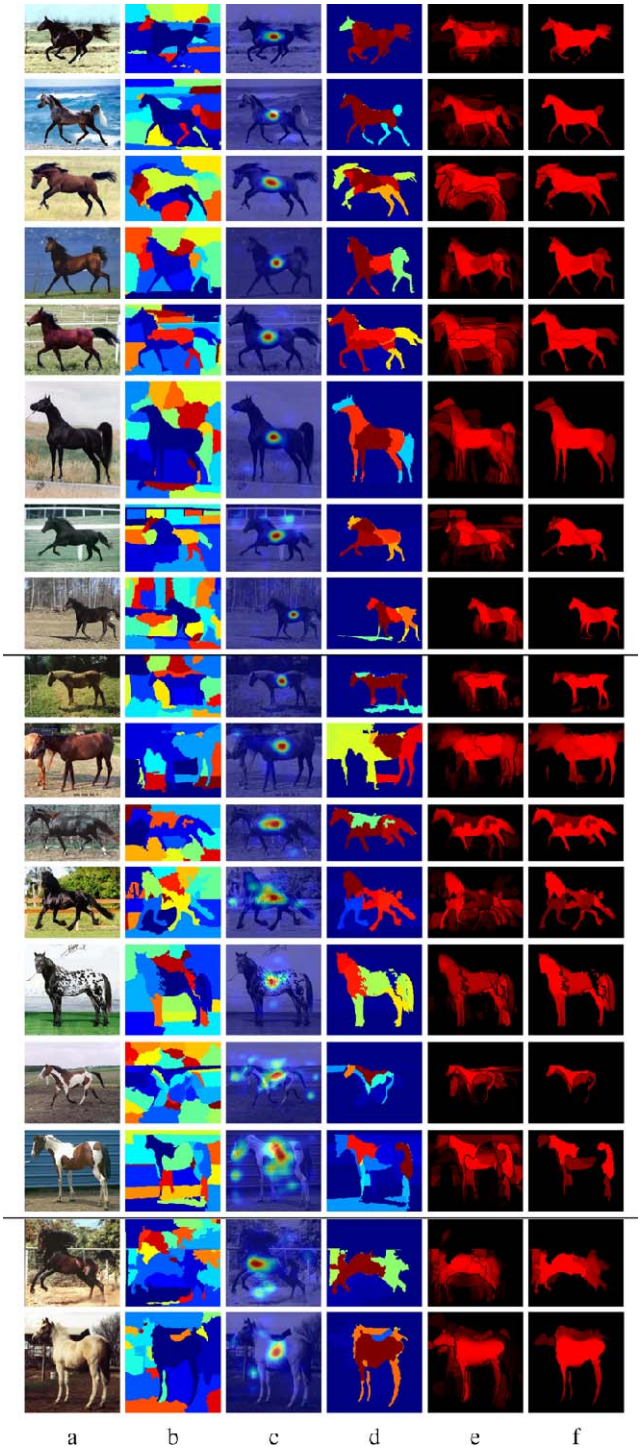

Fig. 13 (**a**) Examples of test images from horse dataset, (**b**) their bottom-up segmentations, (**c**) voting maps, (**d**) expected volume of model pixels covered by each segment (26) (jet colormap in Matlab), (**e**) back-projection map (14), (**f**) top-down segmentation map (27). In the first group of examples the detected hypothesis is significant, and the top-down segmentation is accurate. In the second group the bottom-up segmentation failed to extract some of the parts of an object due to the presence of highlights, texture, shadows or clutter, yet our algorithm is still able to detect the object and accurately delineate it. In the third group we show difficult examples for which the object is detected, but the quality of the top-down segmentation is poor

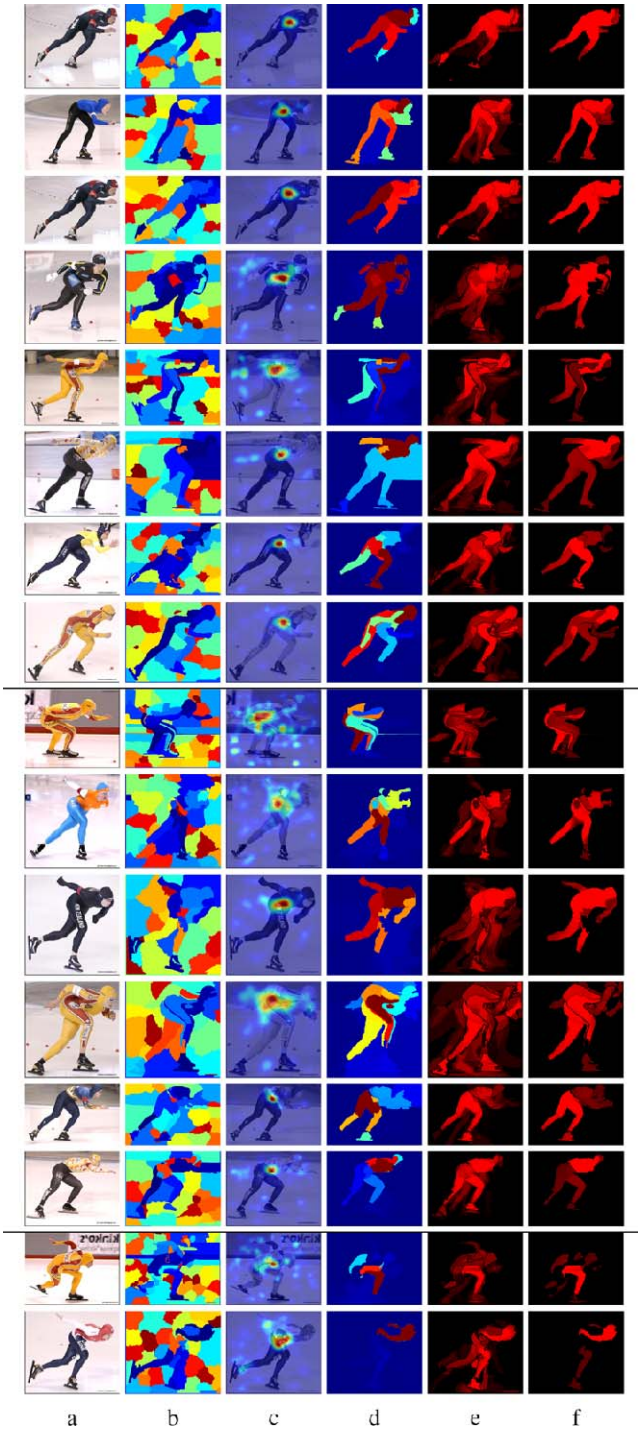

Fig. 14 (**a**) Examples of test images from skate-runner dataset, (**b**) their bottom-up segmentations, (**c**) voting maps, (**d**) expected volume of model pixels covered by each segment (26), (**e**) back-projection map (14), (**f**) top-down segmentation map (27). In the first group of examples the detected hypothesis is significant, and the top-down segmentation is accurate. In the second group the bottom-up segmentation failed to extract some of the parts of an object due to the presence of highlights, texture, shadows or clutter, yet our algorithm is still able to detect the object and accurately delineate it. In the third group we show difficult examples for which the object is detected, but the quality of the top-down segmentation is poor

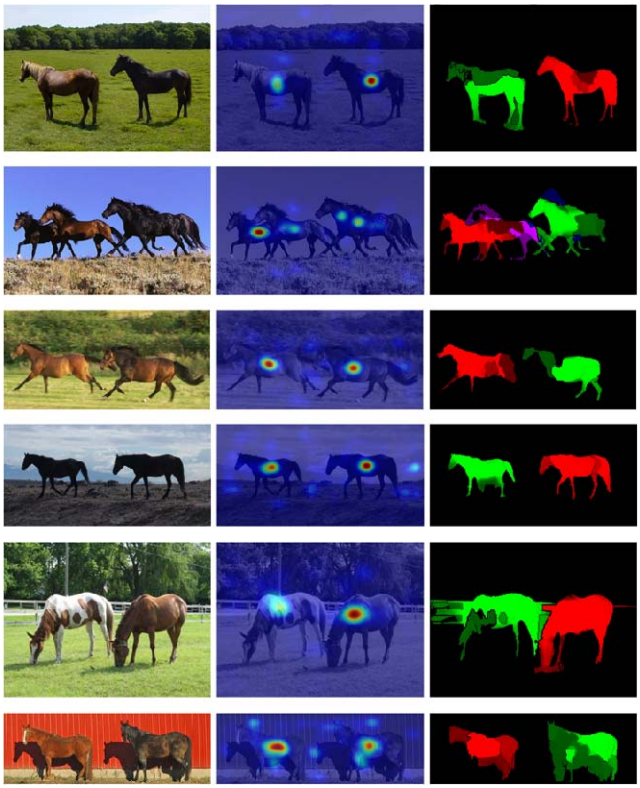

Fig. 15 Out-of-dataset difficult images of horses containing multiple object instances, voting maps, top-down segmentation computed as in (27)

Fig. 16 Out-of-dataset difficult images, voting maps, top-down segmentation computed as in (27)

us to assess the accuracy of our segmentation. We then used the hypothesis $h = (o, x)$ at the location of the highest peak to compute a top-down segmentation map as in (27) and counted the rate of mislabeled pixels when cut with a global threshold. We obtain 91.47% and 91.12% segmentation accuracy respectively for the horse and skate-runner datasets.

Figures 13 and 14 show representative voting and top-down segmentation results of various quality. In the first group of examples the detected hypothesis is significant, and the top-down segmentation is accurate. In the second group the bottom-up segmentation failed to extract some of the parts of an object due to the presence of highlights, texture, shadows or clutter, yet our algorithm is still able to detect the object and accurately delineate it. In the third group we show difficult examples for which the object is detected, but the quality of the top-down segmentation is poor.

Our results on the Weizmann horse dataset (conditions that are idealized for other existing methods, i.e., single non-occluded object in each image) are comparable to the state of the art results reported in the literature. For example, the algorithm of Borenstein (2006) achieved 93% accuracy. LOCUS algorithm (Winn and Jojic 2005) obtained 93% on 200 horses images, and Levin and Weiss (2006) performed over 95%. The OBJ-CUT algorithm (Kumar et al. 2005) also performs at around 96% but only for a subset of this dataset. While having the advantage of being "unsupervised" the algorithm of Cao and Fei-Fei (2007) obtains segmentation accuracy of 81.8% on the horse dataset. Moreover, our method allows us to handle images with multiple instances of objects and substantial occlusions, that are difficult to handle with other methods (see Sect. 4.3).

4.3 Out-of-Dataset Difficult Images

We further performed experiments to demonstrate the ability of our method to cope with multiple instances of class objects and with partial occlusion. In Figs. 15 and 16 we show detection maps and top-own segmentations computed for a number of difficult out-of-dataset images containing multiple instances of horses and skate-runners. The top-down segmentations were computed for the two/three highest peaks of the detection map.

Occlusion is a significant challenge to many existing top-down segmentation methods: For example, Winn and Jojic

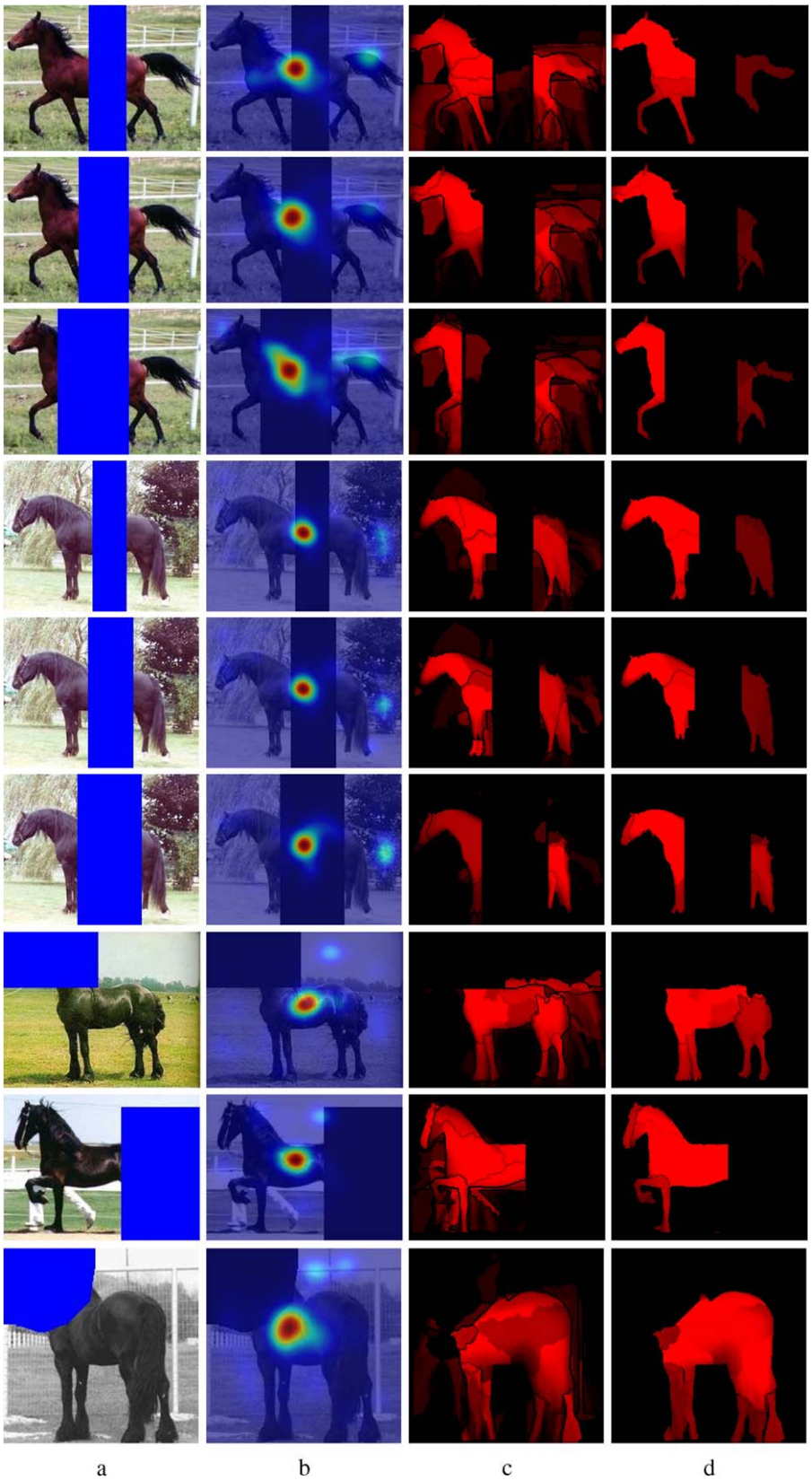

Fig. 17 (**a**) Examples of images with severe occlusions of various types, (**b**) their voting maps, (**c**) back-projection maps (style (Leibe et al. 2008)) as in (14), (**d**) top-down segmentation maps as in (27). Note the difference between the results in (**c**) and in (**d**). Using the competition allows us to eliminate false nearby regions present in the segmentations in (**c**) and accurately segment the objects from their background

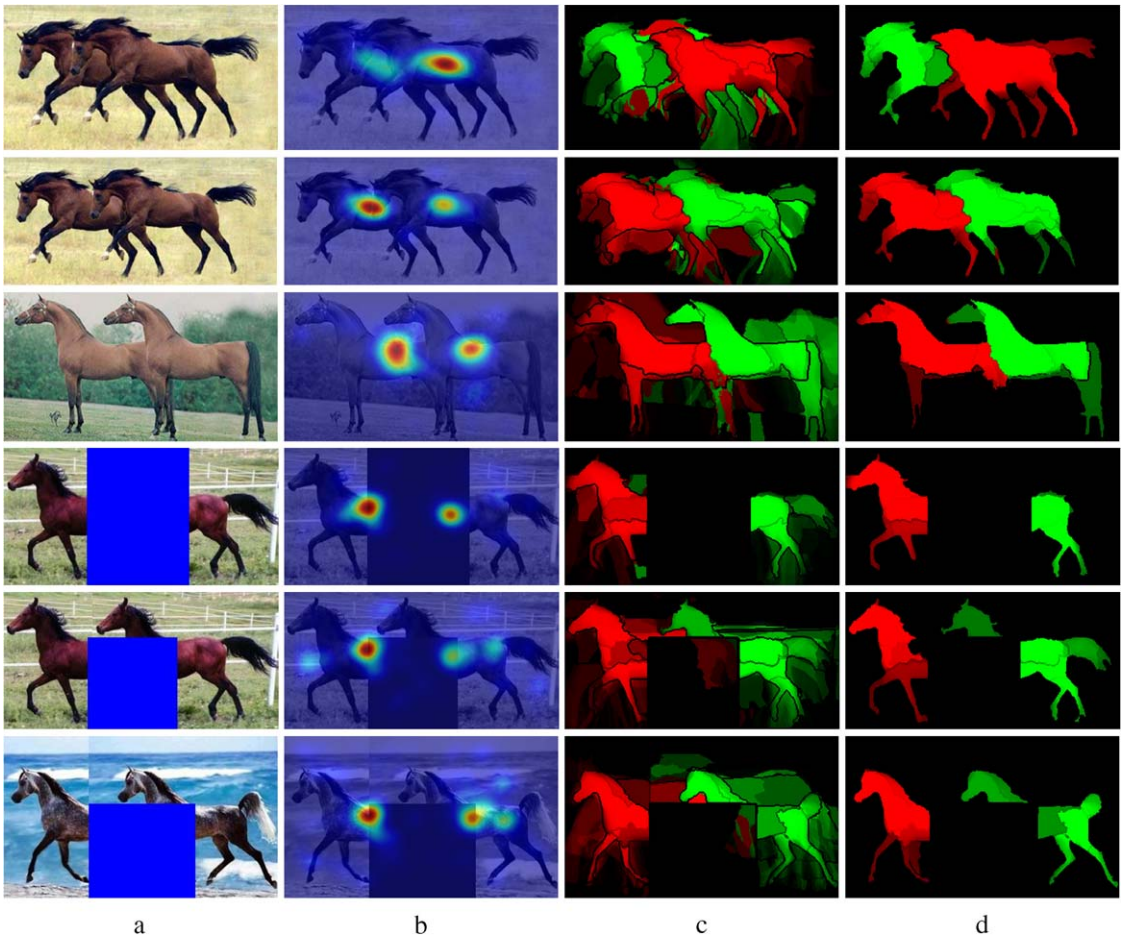

Fig. 18 (**a**) Examples of images containing two instances of horses and showing severe occlusions of various type, (**b**) their voting maps, (**c**) back-projection maps (style (Leibe et al. 2008)) as in (14), (**d**) top-down segmentation maps as in (27). As can be seen in column (**d**) our competitive coverage process allows us to correctly assign image pixels to either *red* or *green* object instances as well as sharply delineate their boundaries

(2005) assumes smooth deformation field of the object class mask. Levin and Weiss (2006) assumes detection resulting in a bounding box as a preprocessing stage to their algorithm and therefore do not handle multiple instances of objects. Moreover, Levin and Weiss (2006) as well as Borenstein et al. (2004) and Borenstein (2006) might have difficulties when the discriminative (most informative) parts of the object are occluded, (e.g., the front half of a horse) or when one horse is occluded by another horse of the same color. Others, e.g., Todorovic and Ahuja (2007), can identify each of the visible parts of an object, but may fail to assemble these into a coherent object detection, a process which is necessary in case of occlusion by e.g., a bar. The method of Leibe et al. (2008) has been shown to handle the types of occlusions mentioned above, but often produces a "ghosting effect" of the type discussed in Sect. 3.3, where correct votes cannot be distinguished from erroneous votes that were matched to the same semantic part of the model.

Figure 17 shows examples of images with severe occlusions of various types in (a), their voting maps in (b), back-projection maps (style (Leibe et al. 2008)) (14) in (c), and our final top-down segmentation maps (27) in (d) computed respectively for the highest peak of the voting map. Note the difference between the results in (c) and in (d). Using the competition allows us to eliminate false nearby regions present in the segmentations in (c) and accurately segment the objects from their background. Figure 18 shows similar results for images containing two instances of horses, with one occluding the other or where the overlap area is occluded by a bar. In this set of images each pixel is assigned the object hypothesis with the highest normalized value of the top-down segmentation. As can be seen in column (d) our competitive coverage process allows us to correctly assign image pixels to either red or green object instances as well as sharply delineate their boundaries.

5 Conclusions

We presented a new segmentation-based method for object detection and figure-ground delineation, which incorporates shape information of coherent image regions emerging from a bottom-up segmentation, along with their bounding contours. Shape information provides a powerful cue for object recognition that can potentially survive differences in appearance due to lighting conditions, color and texture. Using segments and their shape properties allows us to evaluate a smaller set of object hypotheses in the detection stage of our algorithm and provides a natural way for the top-down delineation.

We performed quantitative experimental tests on challenging horse and skate-runner datasets showing that our method can accurately detect and segment objects with complex shapes. Moreover, we showed further qualitative results on a number of difficult out-of-dataset images containing multiple object instances and severe occlusions.

Our work demonstrates that despite errors introduced by the bottom-up segmentation process we can achieve a performance comparable to that obtained with local appearance features, suggesting that those two types of information can potentially be combined to further improve recognition.

Acknowledgements The authors thank Achi Brandt for his help particularly in formalizing a multilevel solution to the Poisson equation. Research was supported in part by the US-Israel Binational Science Foundation grant number 2002/254 and by the European Commission Project IST-2000-26001 VIBES. The vision group at the Weizmann Institute is supported in part by the Moross Foundation.

References

Agarwal, S., & Roth, D. (2002). Learning a sparse representation for object detection. *European Conference on Computer Vision*, *2*, 113–130.

Borenstein, E. (2006). Shape guided object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2006.

Borenstein, E. (2009). http://www.msri.org/people/members/eranb.

Borenstein, E., Sharon, E., & Ullman, S. (2004). Combining top-down and bottom-up segmentation. In *Workshop on perceptual organization in computer vision, IEEE conference on computer vision and pattern recognition*, Washington, 2004.

Burl, M. C., Weber, M., & Perona, P. (1998). A probabilistic approach to object recognition using local photometry and global geometry. In *Lecture notes in computer science* (Vol. 1407).

Cao, L., & Fei-Fei, L. (2007). Spatially coherent latent topic model for concurrent object segmentation and classification. In *International conference on computer vision*, 2007.

Felzenszwalb, P., & Huttenlocher, D. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *61*(1), 55–79.

Ferrari, V., Jurie, F., & Schmid, C. (2007). Accurate object detection with deformable shape models learnt from images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2007.

Gorelick, L., Galun, M., Sharon, E., Brandt, A., & Basri, R. (2006). Shape representation and classification using the Poisson equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12), 2006.

Gustafson, K. (1998). Domain decomposition, operator trigonometry, robin condition. *Contemporary Mathematics*, *218*, 432–437.

Kumar, M. P., Torr, P., & Zisserman, A. (2004). Extending pictorial structures for object recognition. In *British machine vision conference*, 2004.

Kumar, M. P., Torr, P., & Zisserman, A. (2005). Obj cut. In *IEEE conference on computer vision and pattern recognition* (1) (pp. 18–25), 2005.

Leibe, B., Leonardis, A., & Schiele, B. (2008). Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, *77*(1–3), 259–289.

Levin, A., & Weiss, Y. (2006). Learning to combine bottom-up and top-down segmentation. In *European conference on computer vision*, 2006.

Mori, G., Ren, X., Efros, A., & Malik, J. (2004). Recovering human body configurations: Combining segmentation and recognition. In *IEEE conference on computer vision and pattern recognition*, 2004.

Opelt, A., Pinz, A., & Zisserman, A. (2006). A boundary-fragment-model for object detection. In *European conference on computer vision*, May 2006.

Pantofaru, C., Dorko, G., Schmid, C., & Hebert, M. (2008). Combining regions and patches for object class localization (pp. 23–30), June 2006.

Ren, X., Berg, A., & Malik, J. (2005a). Recovering human body configurations using pairwise constraints between parts. In *International conference on computer vision* (Vol. 1, pp. 824–831).

Ren, X., Fowlkes, C., & Malik, J. (2005b). Cue integration for figure ground labeling. In *Advances in neural information processing systems* (Vol. 18), 2005.

Russell, B. C., Efros, A. A., Sivic, J., Freeman, W. T., & Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2006.

Sharon, E., Galun, M., Sharon, D., Basri, R., & Brandt, A. (2006). Hierarchy and adaptivity in segmenting visual scenes. *Nature*, *442*(7104), 810–813.

Shotton, J., Blake, A., & Cipolla, R. (2005). Contour-based learning for object detection. In *International conference on computer vision*, (Vol. 1, pp. 503–510), October 2005.

Todorovic, S., & Ahuja, N. (2007). Learning the taxonomy and models of categories present in arbitrary images. In *International conference on computer vision*, 2007.

Trottenberg, U., Oosterlee, C., & Schuller, A. (2001). *Multigrid*. San Diego: Academic Press.

Ullman, S., Sali, E., & Vidal-Naquet, M. (2001). A fragment-based approach to object representation and classification. In *International workshop on visual form 4*, 2001.

Vidal-Naquet, M., & Ullman, S. (2003). Object recognition with informative features and linear classification. In *International conference on computer vision* (p. 281), 2003.

Wang, L., Shi, J., Song, G., & Shen, I.-F. (2007). Object detection combining recognition and segmentation. In *Asian conference on computer vision*, 2007.

Weber, M., Welling, M., & Perona, P. (2000). Towards automatic discovery of object categories. *IEEE Conference on Computer Vision and Pattern Recognition*, *2*, 101–108.

Winn, J., & Jojic, N. (2005). Locus: Learning object classes with unsupervised segmentation. In *International conference on computer vision*, Beijing, 2005.