Computer Vision and Image Understanding 114 (2010) 1055–1067

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Detecting object boundaries using low-, mid-, and high-level information

Songfeng Zheng [a], Alan Yuille [b], Zhuowen Tu [c,*]

[a] Department of Mathematics, Missouri State University 901 S. National Ave., Springfield, MO 65897, USA
[b] Department of Statistics, Department of Psychology, and Department of Computer Science, UCLA 8967 Math Sciences Bldg, Los Angeles, CA 90095, USA
[c] Lab of Neuro Imaging (LONI), Department of Neurology, and Department of Computer Science, UCLA 635 Charles E. Young Drive South, Los Angeles, CA 90095, USA

ARTICLE INFO

Article history:
Received 13 October 2008
Accepted 15 July 2010
Available online 13 August 2010

Keywords:
Boundary detection
Low-level information
High-level information
Shape matching
Cue integration

ABSTRACT

Object boundary detection is an important task in computer vision. Recent work suggests that this task can be achieved by combining low-, mid-, and high-level cues. But it is unclear how to combine them efficiently. In this paper, we present a learning-based approach which learns cues at different levels and combines them. This learning occurs in three stages. At the first stage, we learn low-level cues for object boundaries and regions. At the second stage, we learn mid-level cues by using the short and long range context of the low-level cues. Both these stages contain object-specific information – about the texture and local geometry of the object – but this information is implicit. In the third stage we use explicit high-level information about the object shape in order to further improve the quality of the object boundaries. The use of the high-level information also enables us to parse the object into different parts. We train and test our approach on two popular datasets – Weizmann horses [3] and ETHZ cows [24] – and obtain encouraging results. Although we have illustrated our approach on horses and cows, we emphasize that it can be directly applied to detect, segment, and parse other types of objects.

1. Introduction

Object boundary detection and foreground/background segmentation are important problems in computer vision, and they are often tightly coupled.

Local cues like gradients used in classical edge detectors (e.g. [4]) are often insufficient to characterize object boundaries [20,27]. For example, Fig. 1 shows the results of the Canny edge detector [4] applied to some natural images with cluttered backgrounds [3]. The edge map alone does not provide enough cues for segmenting the object. Marr [26] proposed a strategy for addressing this problem by combining low-, mid-, and high-level cues. However, despite some progress made in this direction [9,10,32,37], the problem remains unsolved.

Recent advances in machine learning had made it more practical to combine low-, mid-, and high-level cues for object detection. For example, Borenstein et al. [3] combined top-down information (learned configurations of image patches) with bottom-up approaches (intensity-based segmentation) in order to achieve foreground/background segmentation. In the image parsing framework [35], data-driven proposals (using low-level cues) were used to guide high-level generative models. Fergus et al. [14] built a top-down model based on features extracted by interest point operators. Conditional Markov random fields models [22,33] were used to enforce local consistency for labeling and object detection. Other approaches combine bottom-up and top-down learning in a loop [25]. OBJCUT [21] combined cues at different levels in order to perform object segmentation. He et al. [17] proposed a context-dependent conditional random field model to take context into account. In related work, Wang et al. [39,40] proposed a dynamic conditional random field model to incorporate context information for segmenting image sequences. More recently, Zhu et al. [41,42] built hierarchical models to incorporate semantic and context information at different levels.

These approaches have shown the effectiveness of of combining cues at different levels. But, when, where and how to combine cues from different levels is still unclear. For example, it is very difficult to build a generative appearance model, to capture the complex appearance patterns of the horses in Fig. 1; the patches used in [3,7,25] cannot deal with large scale deformations and they also have difficulties in capturing complex variations in appearance. Other approaches, like [16,17,21,23,39,40], lead to complex models which require solving time-consuming inference problems.

In this paper, we use a learning-based approach to learn and combine cues at different levels. This gives a straightforward method with a simple and efficient inference algorithm. More precisely, we use probabilistic boosting trees (PBTs) [34] (a variation of boosting [13]) for learning and combining low- and mid-level cues. Then we use a shape matching algorithm [36] to engage high-level shape information, and to parse the object into different components, e.g. head, back, legs, and other parts of horses or cows. Our strategy relates to Wolpert's work on stacking, which builds classifiers on top of other classifiers, but is very different in detail.

* Corresponding author.
E-mail address: ztu@loni.ucla.edu (Z. Tu).

1077-3142/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2010.07.004

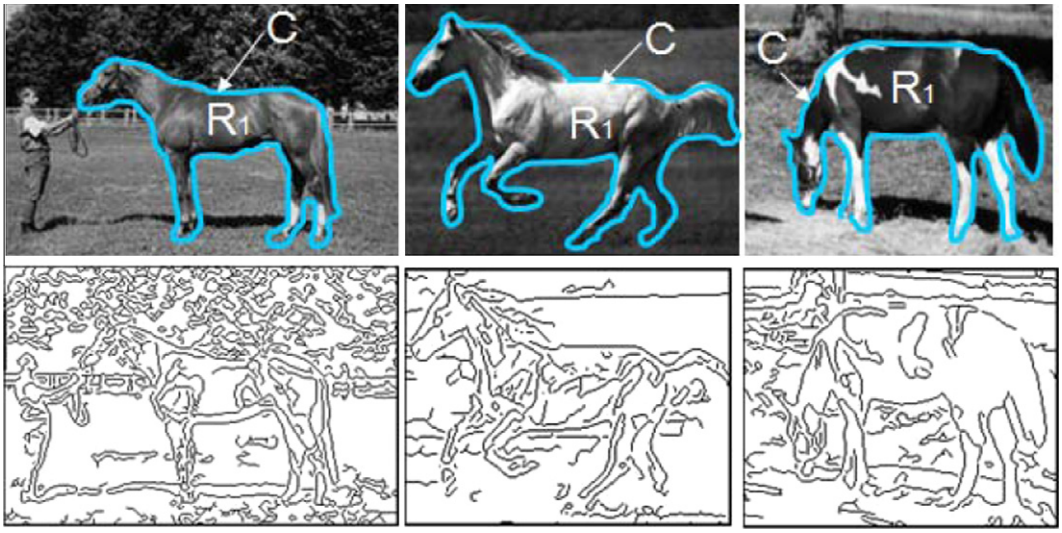


Fig. 1. Examples of the Weizmann horse dataset. The first row shows three typical images, each containing a horse, where C is the boundary we want to detect and R_1 denotes the foreground region. The second row displays edges detected by Canny edge detector at scale $\sigma = 1.0$.

We note that Ross and Kaelbling [29] also addresses segmentation using learning, but their approach is very different and involves motion cues and learning Markov random field models.

We compare our system with other approaches for this problem. The most directly comparable one is the work by Ren et al. [28] which gives detailed performance evaluations for combining low-, mid-, and high-level information. Our results show large improvement over their approach in many respects, particularly at the low- and mid-levels. It is less easy to make direct comparison with other works [3,7,21,25] because some of them [21] were not evaluated on large testing datasets, and the details of performance evaluation were not all given. Also some approaches [21,25] used color images. In [7], the authors first get a shortlist containing 10 candidates, and then pick the best one by hand, while our approach outputs only one result for each image. Hierarchical methods [41,42] obtain very good results but use more complex object models and require heavy inference.

2. Problem formulation

Given an image $\mathbf{I}$, we assume there is an object of interest in the foreground. The goal is to automatically detect the boundary of this object, and thus, perform foreground/background segmentation. In addition, it is desired to parse the object and identify its parts (e.g. head, leg, back, etc. of a horse or cow).

More precisely, we seek to decompose an image defined on a 2D image lattice Λ into two disjoint connected regions R_0, R_1 so that $R_0 \cup R_1 = \Lambda$ and $R_0 \cap R_1 = \emptyset$. R_0 is the background region and R_1 is the foreground (i.e. corresponding to the object). We denote a solution by:

$$W = (R_0, R_1), \quad R_0 \text{ background}, \quad R_1 \text{ foreground}. \tag{1}$$

We can also represent this by the object boundary curve $C = \partial R_1$ with the convention that the object is in the interior of the boundary, i.e. $R_1 = \text{interior}(C)$. In this paper, the object boundaries are closed curves and are represented by point sets.

2.1. The bayesian formulation

The optimal solution W^* for for this boundary detection task can be obtained by solving the Bayesian inference problem:

$$W^* = \arg\max_W p(W|\mathbf{I}) = \arg\max_W p(\mathbf{I}|R_0, R_1)p(R_0, R_1), \tag{2}$$

where $p(\mathbf{I}|R_0, R_1)$ models the image generating process in the foreground and background regions, and $p(R_0, R_1)$ defines the prior for the boundary contour. For example, we can use a probability model for the shape of the object.

However, it is difficult to use Eq. (2) directly because the image generating process is very complicated. Objects, such as horses and cows, have complex image appearance due to their varied texture patterns and the lighting conditions. Moreover, the background is even more varied and complex to model. Hence it is hard to model the image appearance $p(\mathbf{I}|R_0, R_1)$ directly although might be easier to model the boundary shape $p(R_0, R_1)$.

2.2. An alternative perspective

We avoid the difficulties above by defining the conditional distribution $P(W|\mathbf{I})$ directly:

$$P(W|\mathbf{I}) \propto \exp\{-E(W;\mathbf{I})\}.$$

Then we seek to estimate:

$$W^* = \arg\max P(W|\mathbf{I}) = \arg\min E(W;\mathbf{I}). \tag{3}$$

From the definition of C and W, finding the optimal W is equivalent to finding the optimal C. As such, we can rewrite Eq. (3) as

$$C^* = \arg\min E(C;\mathbf{I}),$$

where the energy function $E(C;\mathbf{I})$ is defined by:

$$E(C;\mathbf{I}) = E_{dis}(C;\mathbf{I}) + \tau E_{shape}(C), \tag{4}$$

where $E_{dis}(C;\mathbf{I})$ models the image appearance cues discriminatively, and $E_{shape}(C)$ models the boundary shape.

In our approach, the low- and mid-level cues are captured *implicity* by $E_{dis}(C;\mathbf{I})$. The high-level cues are represented *explicitly* by $E_{shape}(C)$, which is analogous to $-\log P(R_0, R_1)$ in the Bayesian formulation given by Eq. (2). The parameter τ balances the importance of $E_{dis}(C;\mathbf{I})$ and $E_{shape}(C)$ and is determined by cross-validation.

We define $E_{dis}(C;\mathbf{I})$ to be:

$$E_{dis}(C;\mathbf{I}) = -\sum_{\mathbf{r}\in\Lambda/C} \log p(\mathbf{I}(\mathbf{r}), y(\mathbf{r}) = 0|\mathbf{I}(\mathcal{N}(\mathbf{r})/\mathbf{r})) - \sum_{\mathbf{r}\in C} \log p(\mathbf{I}(\mathbf{r}), y(\mathbf{r}) = 1|\mathbf{I}(\mathcal{N}(\mathbf{r})/\mathbf{r}))), \tag{5}$$

where $\mathcal{N}(\mathbf{r})$ is a neighborhood of pixel $\mathbf{r}$; $p(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$ is a discriminative (classification) probability; $\mathbf{I}(\cdot)$ is the intensity value(s) at the given pixels(s); $y(\mathbf{r})$ is a binary variable indicating whether a point $\mathbf{r}$ is on the boundary or not, which is defined as

$$y(\mathbf{r}) = \begin{cases} 1, & \text{if } \mathbf{r} \in C \\ 0, & \text{otherwise} \end{cases}.$$

If we add $-\sum_{\mathbf{r}\in C} \log p(y(\mathbf{r}) = 0|\mathbf{I}(\mathcal{N}(\mathbf{r})/\mathbf{r}))$ to the first term on the right side of Eq. (5) and subtract it from the second term on the right side of Eq. (5), we would have

$$E_{dis}(C;\mathbf{I}) = -\sum_{\mathbf{r}\in C} \log \frac{p(y(\mathbf{r}) = 1|\mathbf{I}(\mathcal{N}(\mathbf{r})))}{p(y(\mathbf{r}) = 0|\mathbf{I}(\mathcal{N}(\mathbf{r})))} - \sum_{\mathbf{r}\in\Lambda} \log p(y(\mathbf{r}) = 0|\mathbf{I}(\mathcal{N}(\mathbf{r})/\mathbf{r})). \quad (6)$$

The second term in the right hand side of Eq. (6) does not depend on C and hence can be ignored. Therefore $E_{dis}(C;\mathbf{I})$ can be formulated as a sum of log-likelihood ratio tests:

$$E_{dis}(C;\mathbf{I}) = -\sum_{\mathbf{r}\in C} \log \frac{p(y(\mathbf{r}) = 1|\mathbf{I}(\mathcal{N}(\mathbf{r})))}{p(y(\mathbf{r}) = 0|\mathbf{I}(\mathcal{N}(\mathbf{r})))}. \quad (7)$$

where $p(y(\mathbf{r}) = 1|\mathbf{I}(\mathcal{N}(\mathbf{r})))$ is the discriminative probability of a pixel $\mathbf{r}$ belonging to the object boundary given an image patch centered at $\mathbf{r}$. The next section discusses how we learn $E_{dis}(C;\mathbf{I})$ to integrate the low-level and mid-level information.

We also need to learn the shape prior term $E_{shape}(C)$ corresponding to the high-level information. We build the shape model on exemplar based approach and use a mixture model to define a shape prior by

$$E_{shape}(C) = -\log\left[\frac{1}{|DB|}\sum_{C_i\in DB} p_{C_i}(C)\right], \quad (8)$$

where DB includes all the shape templates manually labeled for the training images, and $p(C_i)$ [36] allows global affine and local non-rigid transformations for a template C_i in the training images.

To summarize, in our model as Eq. (4), the first term $E_{dis}(C;\mathbf{I})$ integrates low-level and mid-level cues, and the second term $E_{shape}(C)$ explicitly models the high-level shape information. One thing worth to mention is that Eq. (7) depends on both C and the discriminative model $p(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$. Eq. (7) is just a classification ratio for a fixed C. To achieve the minimal energy in Eq. (7) w.r.t. C, reducing the number of pixels on C will not necessarily reduce the energy because $\log\frac{p(y(\mathbf{r})=1|\mathbf{I}(\mathcal{N}(\mathbf{r})))}{p(y(\mathbf{r})=0|\mathbf{I}(\mathcal{N}(\mathbf{r})))}$ can be either positive or negative. Therefore, Eq. (7) will not lead to trivial solution where there is either no point in C or C being the entire image lattice.

3. Learning $E_{dis}(C;\mathbf{I})$

We now describe how to learn and compute $p(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$ in Eq. (7). This will be performed by training classifiers which classify a pixel $\mathbf{r}$ based on the image intensities $\mathbf{I}(\mathcal{N}(\mathbf{r}))$ within a neighborhood $\mathcal{N}(\mathbf{r})$. The information within the neighborhood is required because images are locally ambiguous.

The size of the neighborhood $\mathcal{N}(\mathbf{r})$ determines the range of *context* used to classify the pixel. There are two extreme situations. Firstly, the neighborhood is very small and may, in the extreme case, only contain the pixel $\mathbf{r}$ itself. It is easy to learn and compute a classifier defined over a small neighborhood, but the classification performance will be poor since the neighborhood contains too little information to classify the pixel. Secondly, the neighborhood is very large – e.g., the entire image. In this case there is sufficient information in the neighborhood to classify the pixel. But the problem is how to learn and compute a classifier that takes advantage of this information. Moreover, training a classifier over such a large neighborhood requires a large amount of data to avoid over-fitting.

In this paper, the low-level cues will be defined using small sized neighborhoods. The mid-level cues will take the low-level cues as input and combine them using larger neighborhoods and hence introduce short and long range context.

3.1. Low-level cues

Classic edge detectors [4,18] only depend on the intensity gradients which correspond to using a very small neighborhood $\mathcal{N}(\mathbf{r})$. The relative ineffectiveness of Canny edge detectors, as shown in Fig. 1, demonstrate that these local cues are not rich enough. Learning-based approaches using larger neighborhoods have shown to outperform the Canny edge detector [11,20,27], and our approach follows and extends this line of work.

To model low-level cues, we learn classifiers based on image properties computed in local neighborhoods. We learn two types of low-level cues. Firstly boundary-cues $p(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$ which classify whether pixels are on (i.e., $y(\mathbf{r}) = 1$), or off (i.e., $y(\mathbf{r}) = 0$), the object boundary. Secondly, body-cues $p(z(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$ which classify whether a pixel is foreground (i.e. inside the object, $z(\mathbf{r}) = 1$) or background (i.e. outside the object, $z(\mathbf{r}) = 0$).

3.1.1. Learning boundary cues

We model boundary-cues by $p(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$ using 31×31 image patch $\mathbf{I}(\mathcal{N}(\mathbf{r}))$ centered at pixel $\mathbf{r}$.

The number of samples from a single training image is the number of pixels in that image, of which most (over 90 percent) are negative samples.

We use Boosted Edge Learning (BEL) [11] which is designed for learning edge detection. We restrict it to learning boundary cues – i.e., we distinguish between boundary and non-boundary, instead of between edge and non-edge. BEL is trained by using the probabilistic boosting tree (PBT) algorithm [34], which is a variant of boosting. We briefly describe below how to learn and compute $p(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$, and we refer to [11] for more details.

We use a dictionary of 30,000 candidate features. These include image intensity, image gradients, Haar filters, Gabor filters, differences of Gaussians (DOGs), and differences of offset Gaussians (DOOGs)). All are evaluated at different scales and locations.

For training:

(1) Collect a set of training images in which the object boundaries are manually labeled.
(2) Sample a number of positive examples (image patches with a boundary pixel at the center) and negative examples (image patches with a non-boundary pixel at the center) to form a training set.
(3) Train a boosting classifier [13] using a dictionary of roughly 30,000 features computed in each image patch – including Canny edges at different scales, the magnitude and orientation of gradients, Gabor filter responses at different scales and orientations, and Haar filter responses [38].
(4) Divide the training set (or bootstrap more samples from the training images) into left and right branches and recursively train sub-trees.
(5) Return to step 3 until a stopping criterion is met (either it reaches the specified level or there are too few training samples).

For testing:

(1) Scan through the input image pixel by pixel.
(2) Compute the discriminative probability $p(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$, based on the features selected by the overall classifier from a 31×31 image patch.
(3) Output the edge probability map.

In the testing procedure, the overall discriminative probability is given by:

$$p(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r}))) \approx \sum_{l_1} \hat{q}(y(\mathbf{r})|l_1, \mathbf{I}(\mathcal{N}(\mathbf{r})))q(l_1|\mathbf{I}(\mathcal{N}(\mathbf{r})))$$
$$\approx \sum_{l_1,\cdots,l_n} \hat{q}(y(\mathbf{r})|l_n,\cdots,l_1,\mathbf{I}(\mathcal{N}(\mathbf{r})))\cdots q(l_2|l_1,\mathbf{I}(\mathcal{N}(\mathbf{r})))q(l_1|\mathbf{I}(\mathcal{N}(\mathbf{r})))$$

where the l_i's are augmented variables denoting the tree levels in PBT. $l_i \in \{+1,-1\}$ indicates which branch node i points to, i.e., $l_i = +1$ and $l_i = -1$ point to the right branch and the left branch respectively. $q(l_i|l_{i-1},\cdots,l_1,\mathbf{I}(\mathcal{N}(\mathbf{r})))$ is the discriminative probability computed by the AdaBoost strong classifier at the node specified by $(l_1,\cdots,l_{n-1})$, and $\hat{q}(y(\mathbf{r})|l_n,\cdots,l_1,\mathbf{I}(\mathcal{N}(\mathbf{r})))$ is the fraction of example having class label y at the leaf node, which is estimated in the training process. At the top of the tree, information is accumulated from its descendants and an overall posterior is calculated.

Fig. 2a illustrates the boundary classifier learnt using BEL. Fig. 2b shows an example of the output probability boundary map representing the probability of each point being on the boundary (the darker the pixel the higher the probability is). The result shows significant improvement over the Canny results in Fig. 1.

3.1.2. Learning body cues

Our second low-level cue exploits knowledge about the regional properties of the object and the background. This provides complementary information to the edge-based information described above.

We learn a model $p(z(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$, where $z(\mathbf{r}) = 1$ if pixel $\mathbf{r}$ is on the object, and $z(\mathbf{r}) = 0$ otherwise. This gives an implicit way to model the texture and other appearance properties of objects and the background. We use PBT learning, as described above.

We use a dictionary of 35,000 features. These include the 30,000 features used for the boundary classifier with an addition 5000 features which are histograms of Gabors filters (designed to capture texture properties of regions).

Fig. 2c shows an example of a probability map of the foreground object (the brighter the pixel, the bigger the probability is).

3.2. Mid-level cues: exploit context information

We now proceed to build a mid-level classifier which combines low-level cues such as the boundary map and the body map. This enables us to add context information – for example, a boundary edge is more likely if the body map provides evidence for background on one side of the edge and for foreground on the other side. This gives a *refined boundary map*.

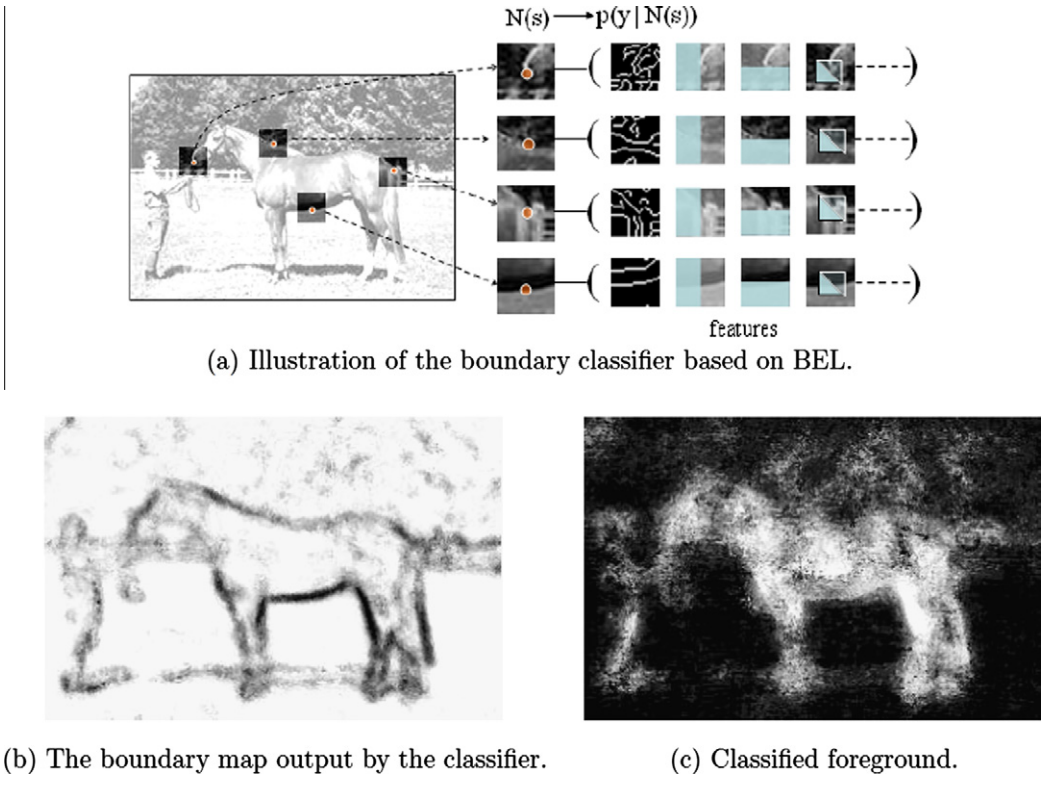
More precisely, we learn a probability distribution for the refined boundary map $p_R(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$. This will be based on the probability boundary map $p(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$ and the probability body map $p(z(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$.

Conditional random fields [22] or hierarchical random field models [1,16,23] are able to exploit some context information, but are limited by the number of neighborhood connections and require time-consuming inference algorithms. By contrast, we learn a direct classifier to combine the context information.

We design two schemes to learn the refined boundary map $p_R(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$.

3.2.1. Short-range context

We first use a simple approach to learn another classifier using inputs from the edge and body maps. To improve the precision of the edges, we prune the edge map by removing all points for which there is no Canny edge at small scale (e.g., $\sigma = 1.0$). This assumes that the majority of the boundary pixels will appear in the Canny edge map at small scale.



(a) Illustration of the boundary classifier based on BEL.

(b) The boundary map output by the classifier.

(c) Classified foreground.

Fig. 2. The boundary and body maps based on BEL and learnt by PBT. Panel (a) shows some positive examples and features, (b) shows a boundary map, which is clearly better than the Canny edges, and (c) shows the body map.

From Fig. 1, we can see that the Canny edge map has good localization, but it has many false positives. The probability boundary map has fewer false positives, as shown in Fig. 2b, but poor localization. Therefore, the Canny edge map provides complimentary information to probability boundary map. For a pixel **r** on the Canny edge map, we consider image patches of size 31×31 centered at pixel **r**. We then train a classifier using a dictionary of 5000 features, which includes Haar feature responses from the edge map, the body map, and the Canny edge map. The training/testing procedure is identical to that described in learning/computing processes of the probability boundary or body map. Fig. 3a shows the result of an example image.

We call this "short-range" context information and it is illustrated in Fig. 4a.

3.2.2. Long-range context

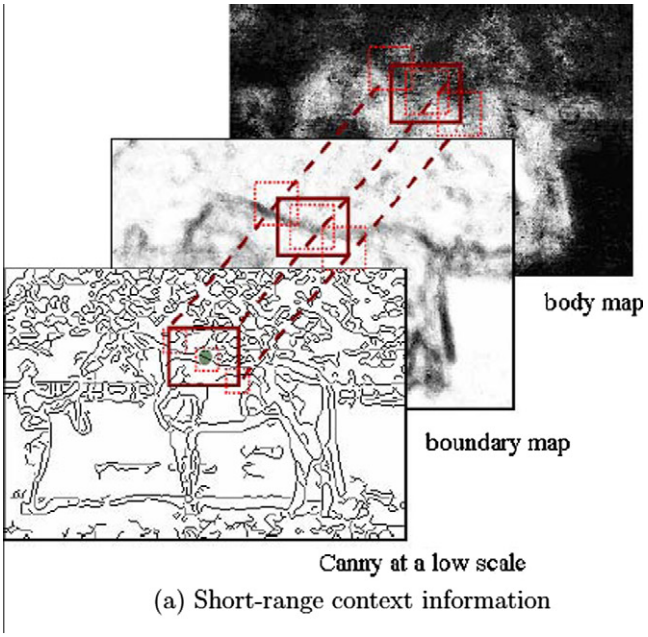
We design an alternative strategy to exploit long-range context. Intuitively, any point on the boundary of the foreground object has similar appearance properties to a point on "the other side" of the object.

Given a boundary point, we shoot a ray along its normal direction until it hits another boundary point. Often: (1) the intensity patterns in between the two boundary points obey some regularities and (2) the local geometric properties of the two boundary points also have some consistency (e.g., the gradient directions are parallel). This relates to previous work [15] which uses Gestalt rules [19] to exploit this type of information.

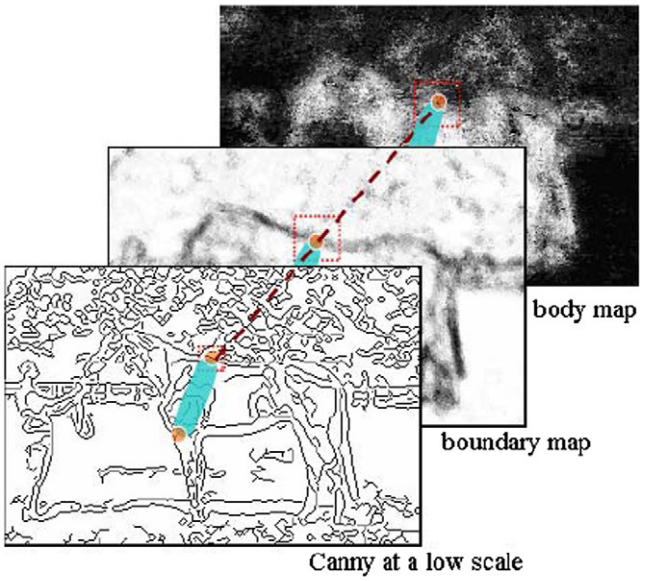
This motivates us to study image patches centered at two points, and measure their similarity. As before we use patches of size 31×31. If both points are on the object boundary, we consider them as a positive pair example; otherwise, it is a negative pair example. Then we build a classifier to classify the positive and negative examples.

For each example, we extract around 20,000 features, which include differences of the texture patterns of the two image patches, differences of geometric properties of the two ending points, differences of filter responses of the two ending image patches, and differences between the boundary/body map of the two ending points. Fig. 4b shows an illustration. The training process is similar to that described earlier, except now each example is a pair of points. The classifier tries to select and combine a set of features, based on these difference measures, to estimate how likely two points lie on the boundary of an object. In the testing stage, for every point in the Canny edge map, we shoot a ray along its normal direction. For any edge point on the ray, we apply the learned classifier to compute how likely the two form a pair of object boundary points. The probability of each point being on the object boundary is given by the maximal probability among all the pairs for this point. Our results also demonstrate an improvement over the original boundary map and Fig. 3b shows some results.

In Fig. 3a and b, we observe that both short-range and long-range context information improve the quality of the boundary map, with the classifier using short-range context being more effective. We will give quantitative measures and a detailed analysis in Section 6.1.

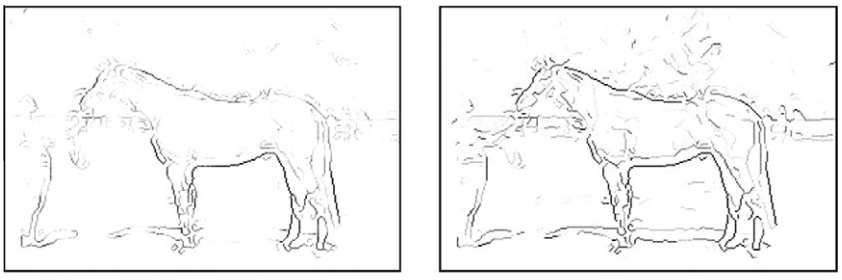


(a) Short-range context information



(b) Long-range context information

Fig. 4. Illustration of the refined probability map using short-range and long-range context.



(a) Short Range Context (b) Long Range Context

Fig. 3. The refined probability map on an example image.

4. Incorporating high-level information by shape matching

Once the refined boundary map $p_R(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$ has been learned by integrating the low- and mid-level cues, we proceed to infer a solution C from Eq. (4). This requires taking into account the shape prior $E_{shape}(C)$, which is regarded as high-level information in this paper. We perform this by shape matching using a method reported in [36]. This method can be viewed as a probabilistic combination of the approaches in [5] and shape context [2]. Recall that $E_{shape}(C)$ is represented by a mixture model, as shown in Eq. (8). We formulate the term $p_{C_i}(C) \propto \exp\{-E_{shape}(C, C_i)\}$ by

$$E_{shape}(C, C_i) = E_{matching}(C, \mathscr{T}(C_i)) + E_{prior}(\mathscr{T}) \tag{9}$$

where C_i denotes one of the templates in the training set, and $\mathscr{T} = (\mathbf{A}, \mathbf{f})$ includes a global affine transformation ($\mathbf{A}$) and a local non-rigid deformation ($\mathbf{f}$) on C_i. The first term is the similarity between C and a transformation of C_i by $\mathscr{T}$, and the second regularizes the transformation $\mathscr{T}$. Intuitively, we prefer the template which best matches to C without undergoing a large deformation.

We represent the shape as a point set sampled from the boundary. Suppose the point set for the target shape is $\{\mathbf{x}_i : i = 1, \ldots, M\}$, which is sampled from the refined boundary map according to probability $p_R(y(\mathbf{r})|\mathbf{I}(\mathcal{N}(\mathbf{r})))$; we denote the template point set as $\{\mathbf{y}_a : a = 1, \ldots, N\}$ which is sampled from the boundary of a training object. We want to morph the template to the target with a small energy on the transformation. Then the energy function for the shape matching can be defined as in Eq. (10):

$$\begin{aligned} E_{shape}(\{m_{ai}\}, \mathbf{A}, \mathbf{f}) = & \sum_{i=1}^{M} \sum_{a=1}^{N} m_{ai} \{\|\mathbf{x}_i - \mathbf{A}\mathbf{y}_a - \mathbf{f}(\mathbf{y}_a)\|^2\} \\ & + \lambda \|L\mathbf{f}\|^2 + T \sum_{i=1}^{M} \sum_{a=1}^{N} m_{ai} \log m_{ai} - \zeta \sum_{i=1}^{M} \sum_{a=1}^{N} m_{ai} \end{aligned} \tag{10}$$

where $(\mathbf{A}, \mathbf{f})$ is the geometric transformation and $\mathbf{A}$ corresponds to the affine part and $\mathbf{f}$ is the non-rigid deformation part. $m_{ai} \in (0, 1)$ measures the goodness of match between point $\mathbf{x}_i$ and the transformed point $\mathbf{A}\mathbf{y}_a + \mathbf{f}(\mathbf{y}_a)$; $L\mathbf{f}$ measures the smoothness of the non-rigid part of the transformation; λ, T, and ζ are positive parameters which balance the importance of each term; The detailed explanation of these parameters and the optimization process can be found in [36]. Fig. 5 gives an illustration of the basic idea.

The overall shape matching algorithm combines low-, mid-, and high-level information, and it give an estimate of the object boundary. Using this estimate, we can perform foreground/background segmentation, object detection, and object parsing.

An alternative way to incorporate high-level information is to use GrabCut [30] algorithm, initialized with the probability body map. However, this approach was not used in this paper.

5. Outline of the algorithm

We are now equipped with all components of our algorithm so we give a complete outline.*Training*:

(1) Collect a set of training images with the object boundaries manually annotated. Obtain the corresponding shape templates using these labels.
(2) Train a classifier on the object boundaries to obtain the boundary maps.
(3) Train a classifier on the foreground label maps to obtain body maps.
(4) Train an overall classifier based on the low-level maps (using either short-range or long-range context, see Section 3.2).

Testing, given an input image:

(1) Run the boundary classifier to obtain the boundary map.
(2) Run the body map classifier to obtain the body map.
(3) Run the overall classifier using context to obtain a refined boundary map (short-range context gives a better result).
(4) Sample points based on the probability boundary map obtained in step 3.
(5) Use the shape matching algorithm to match the obtained point set from step 3 against the shape templates in the training set, select the one with the smallest energy as the best match.
(6) Based on the best matching result, refine the boundary map, and perform foreground/background segmentation and object parsing (see Section 6.1).

Fig. 6 illustrates our approach showing how it uses low-, mid-, and high-level cues. The Canny edge uses low-level cues since it uses the intensity gradient only. The low-level boundary and body maps also depend on local properties of the image. The mid-level cues use either short-range context or long-range context, which is more like traditional Gestalt grouping laws. Finally, the high-level stage uses object models to clarify ambiguities which cannot be resolved without using explicit shape information.

6. Experimental results

We tested our approach on two publicly available datasets: the Weizmann horse dataset [3] and the ETHZ cow dataset [24]. Both datasets contain manually segmented foreground objects for training and evaluation. In both the experiments, we perform the task of object boundary detection, foreground/background segmentation, and object parsing (based on further annotated object parts). In

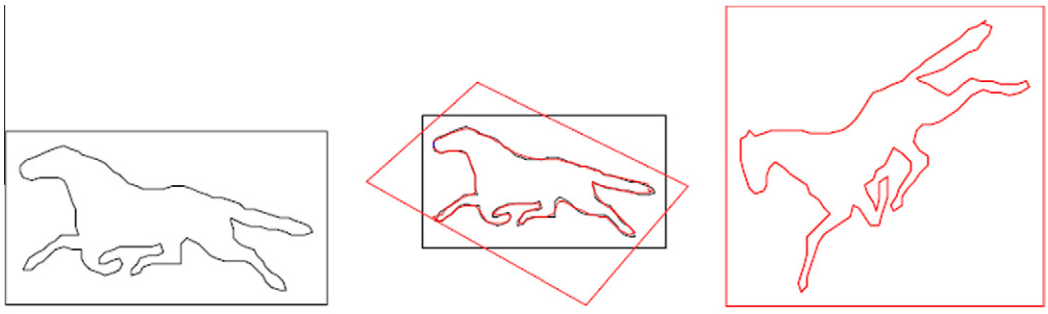


Fig. 5. Illustration of a shape matching case in which a source shape C_i is matched with a target shape C through a transformation $\mathscr{T}$.

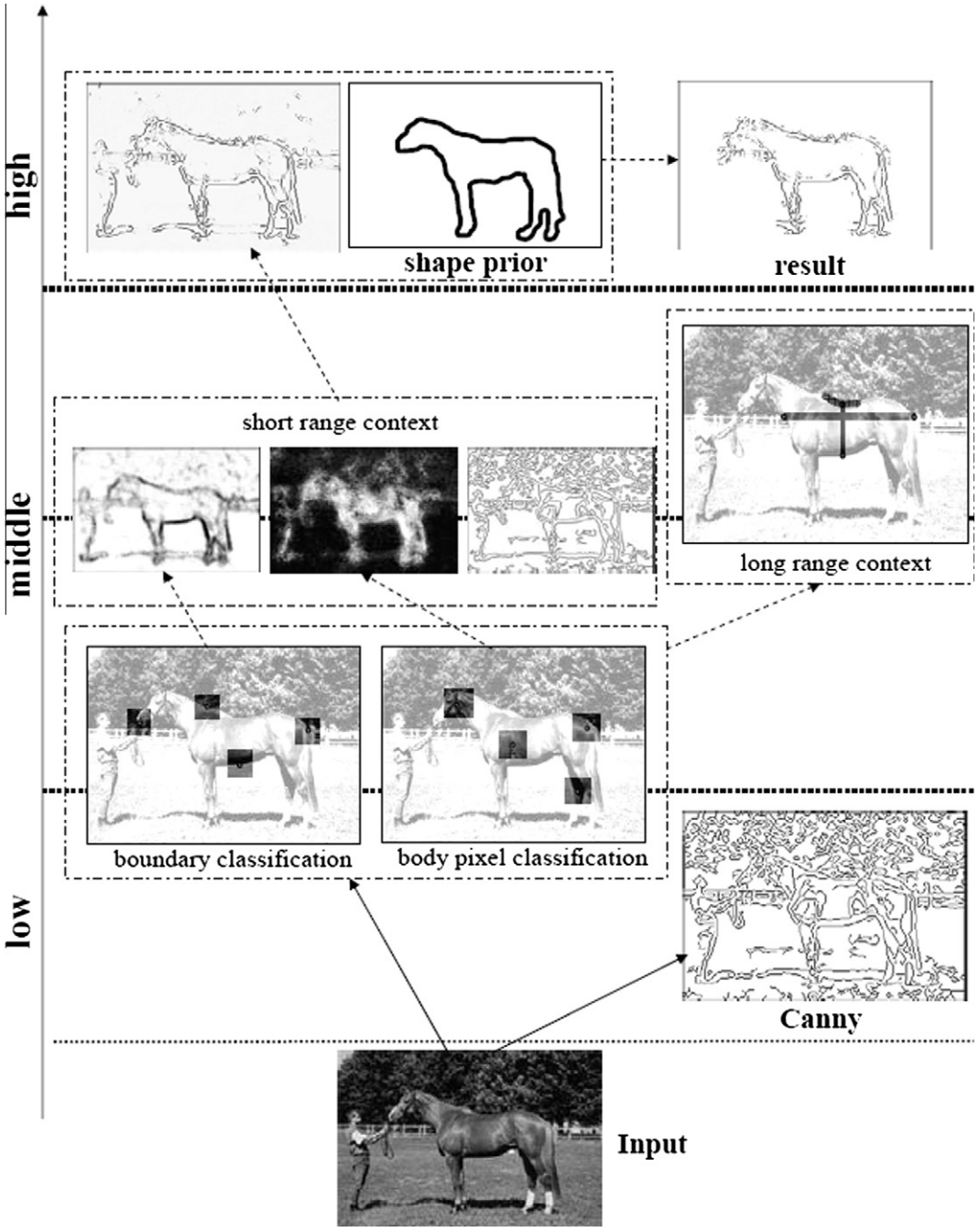


Fig. 6. Illustration of our methods engaging the low-, mid-, and high-level cues.

this section, the probability maps are all normalized to [0,255] for the purpose of visualization.

6.1. Results on the Weizmann horse dataset

In this experiment, the system was trained on 150 gray-scale images randomly selected out of 328 images from the Weizmann horse dataset [3], and we used the rest 178 images for testing. For each image, there is only one foreground object and it mostly appears in the center (but this knowledge is not exploited by the algorithm). But these images have large variations in appearance, scale, and pose.

We used the boundary and body probability maps which are learnt as described in Section 3.1.1. Our analysis shows that the boundary map mostly selects Haar features while the body map prefers histograms of Gabors. This is not surprising since histograms of Gabors are effective at capturing the appearance of texture patterns [43]. Fig. 7a shows some tests images, with the detected boundary and body maps shown in Fig. 7b and c, respectively.

The training of these low-level cues needs 10 h (it depends on the size of the training set and the parameter settings of the probabilistic boosting tree algorithm). The testing stage requires about 15 s for a typical 300×200 gray scale image. Standard code optimization techniques can reduce these times significantly. The computer used in this experiment was an ordinary PC with 2.4 GHz CPU and 1.0 GB memory.

After the low-level cues were learnt, we trained another classifier to use the mid-level cues. We trained both short-range context and long-range context, as described in Section 3.2. Our results

show that short-range context gives better results, see Figs. 10 and 3a and b. The output is the refined boundary map, as shown in Fig. 7d. This stage – using mid-level context – gives the biggest performance improvement in our system, as illustrated in Fig. 10.

Finally, we incorporate the high-level shape models to improve the results of the refined boundary map. We use the exemplar based approach described in Section 4. We sample 300 points from the refined boundary map and match them to the 150 exemplars by minimizing the energy given by Eq. (4). We define the *final match* to be the match with lowest energy. The time spent on matching is about 1 min despite having 150 templates.

We use the final match to improve the refined boundary map by removing false alarms and inferring missing parts. In particular, we make the following changes to obtain the *final boundary*: (I) If part of the refined boundary map is far from the shape matching result, then we decrease its magnitude by 10%. (II) If part of the Canny edge map is close to the final match, then we enhance the corresponding part of the refined boundary map by 10%. Given the final boundary as a probability map, we apply thresholding to get a binary map.

In addition, the final match enables us to parse the object and detect parts, such as the head, back, and legs (by annotating the object exemplars with these labels). Fig. 7e shows the final results for some test images including the labels of object parts.

We refer to Fig. 8 for more results on the horse dataset. Fig. 9 shows some "failure" examples by our algorithm, which are mostly due to confusing and cluttered backgrounds. For these examples, even human beings have difficulties in telling where the horse boundary is.

We use precision–recall to evaluate the performance of our system, which have been widely used in information retrieval [8]. The task of boundary detection has a large skew of class distributions,

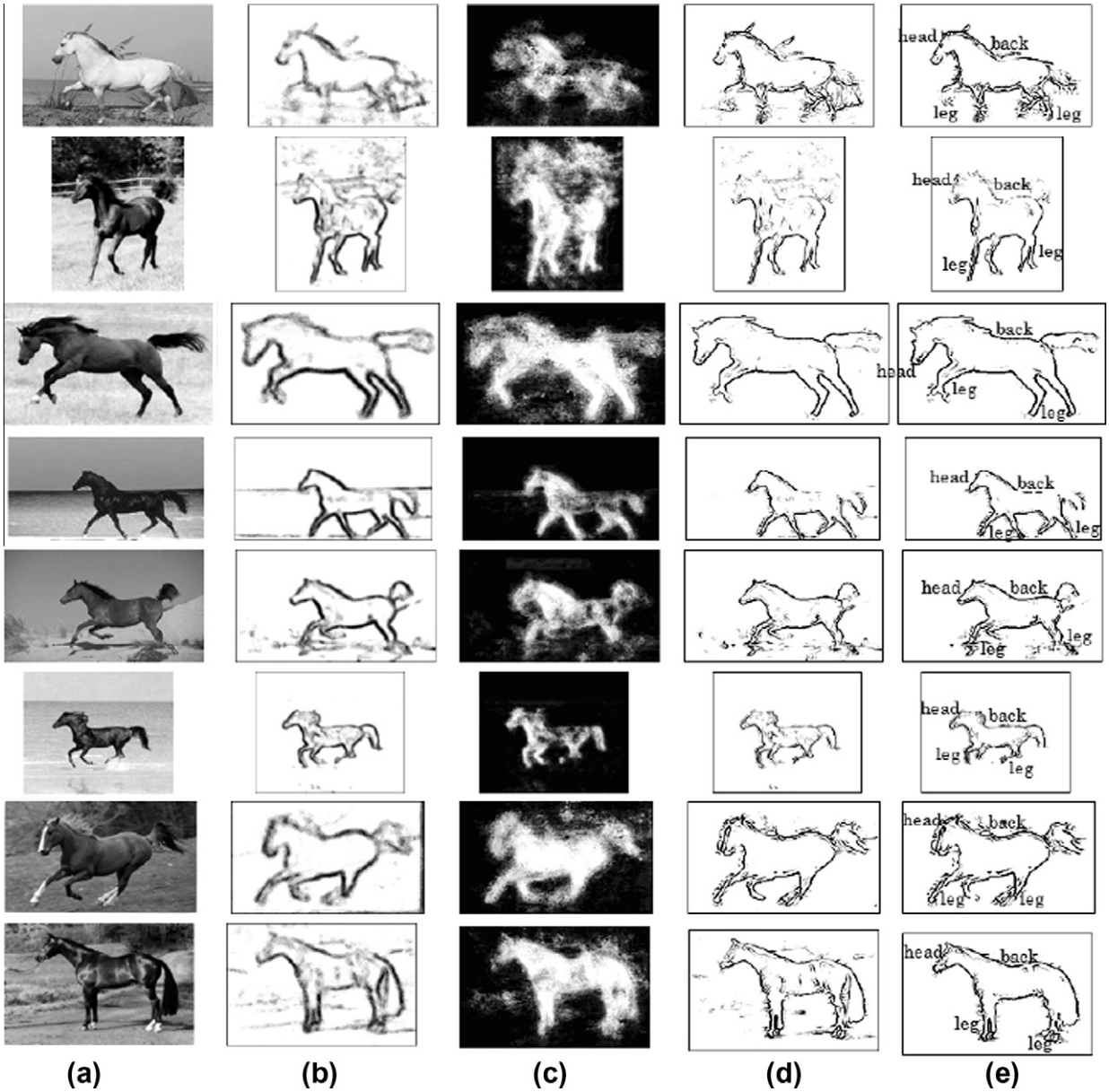


Fig. 7. Results on some testing images from Weizmann horse dataset: (a) shows input images in gray scale; (b) are the probability oundary maps; (c) shows the probability body maps; (d) demonstrates refined boundaries based on the short-range context information; and (e) gives final boundary maps after shape matching, and also labels different parts of the horses according to the shape matching results. These images are representatives of the dataset, which have different appearances, poses, scales, and lighting conditions. We see how cues at each level help to detect the boundaries.

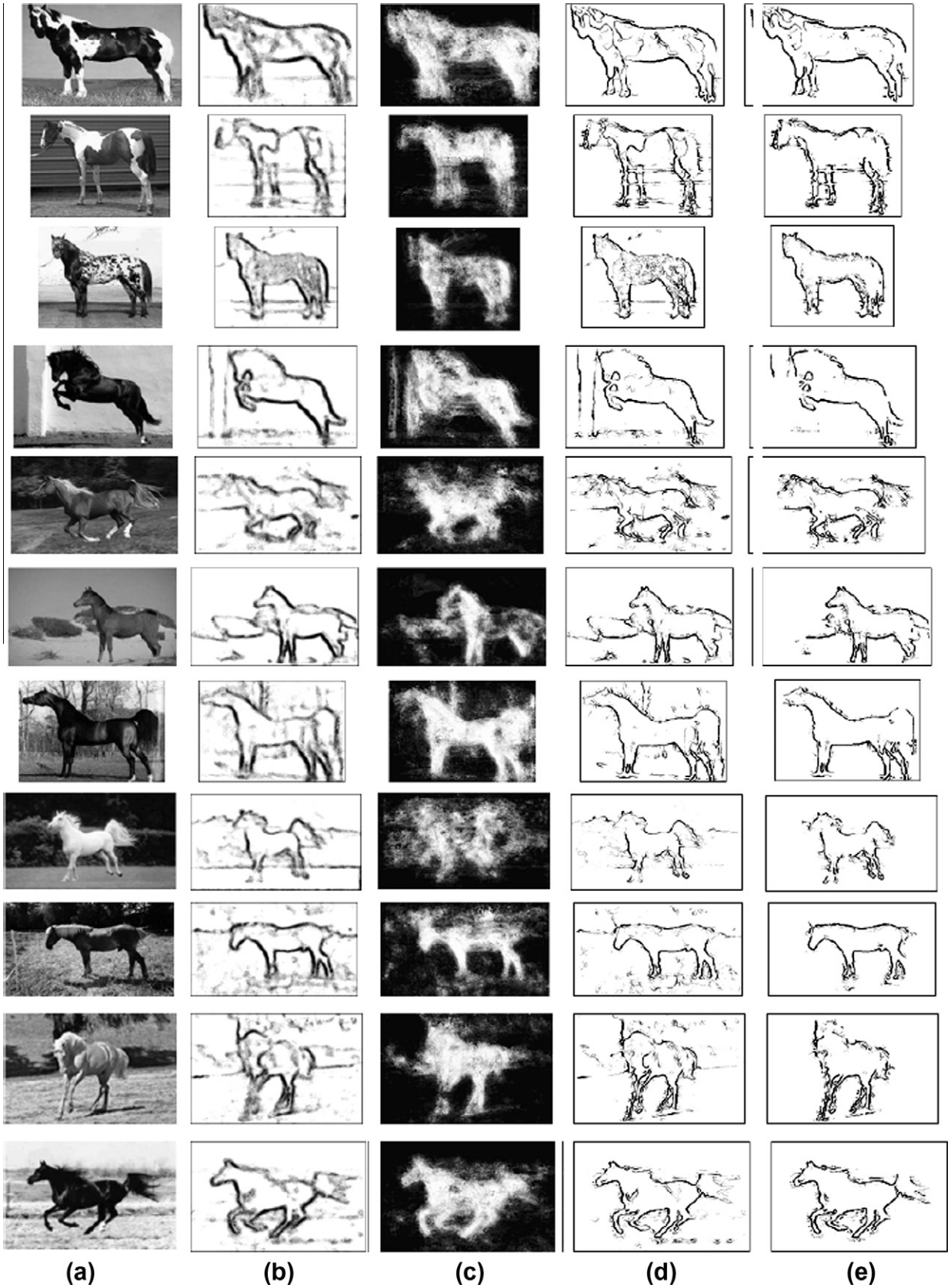


Fig. 8. More results on some testing images from Weizmann horse database. See Fig. 7 for the meaning of each column. For the clarity, the labels of the parsing result are not presented.

so precision–recall rate would be a better measure than ROC [8]. Precision is the fraction of the object detected that belongs to the ground truth, and recall is the fraction of the object belonging to the ground truth that is successfully retrieved. Formally, we set

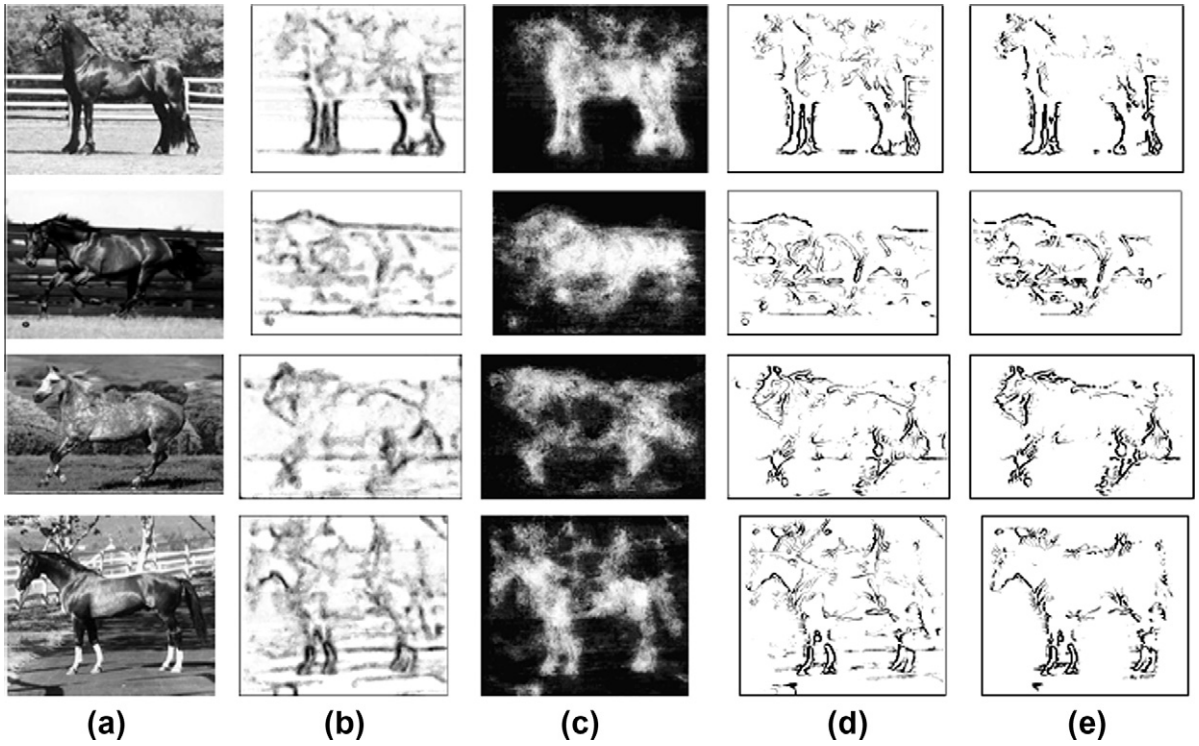


Fig. 9. Some failure results on test images from Weizmann horse database: (a) shows input images; (b) are the probability boundary maps (c) shows the probability body maps; (d) demonstrates refined boundaries based on the short-range context information; and (e) gives final boundary maps after shape matching. These images are among the most difficult images in the data set, even humans find it difficult to detect the horse boundary precisely.

$$\text{precision} = \frac{|D \cap L|}{|D|} \quad \text{and} \quad \text{recall} = \frac{|D \cap L|}{|L|},$$

where L is the manually labeled target, and D is the detected target which is obtained by applying a threshold value to the final boundary probability map. The harmonic mean of precision and recall is called the F-measure or balanced F-score, which is defined as:

$$F = 2(\text{precision} \times \text{recall})/(\text{precision} + \text{recall}),$$

and is often used as a measurement of the overall performance of the system [8].

We allow for a tolerance of 3 pixel-width and a boundary point within 3 pixel range of the ground truth is considered as a success (this is a typical procedure in the evaluation of the edge/boundary detection algorithm [27]). Changing the threshold for obtaining detection result would result in a set of different precision and recall values. Fig. 10 shows the average precision and recall values, and the four black curves are produced by our system. As we can see, the performances improved when we use cues from more levels.

We observe that short-range context information is more effective than long-range context information, as shown in Figs. 10 and 3. This may be because the long-range context relies on pairs of image patches which are more difficult to classify than the single patches used by the short-range cues.

Another observation is that we get a big performance improvement by adding mid-level context cues to low-level cues than when we add the high-level shape cues. This is surprising, but may occur because the shape models used are not rich enough to capture the large variations of articulated objects like horses and cows.

Fig. 10 also shows the performance curves from [28] for comparison. Our system achieves better performance when their system was trained on 174 images.

We obtain a similar set of precision–recall measures by comparing the results with the ground truth. The detection rate curve is shown in Fig. 11. For example, Fig. 11 shows that at a certain threshold value, we have 95% of the foreground pixels and 83% of the background pixels correctly labeled. This is better than the performance reported in [3,25] (the two percentages they reported are 95% and 66%, respectively).

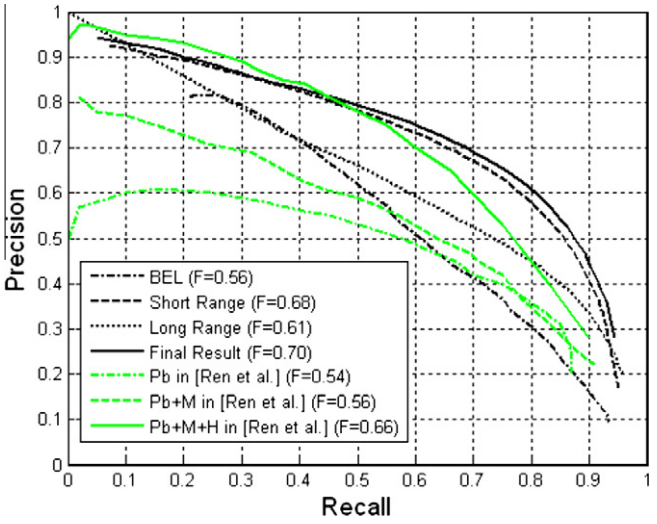


Fig. 10. Precision–recall curves for boundary detection of the horse testing images. The four black curves show the results of the proposed approach. Results on the same dataset from [28] are also displayed for a comparison (three grey/green curves). The F value shown in the legend is the maximal harmonic mean of precision and recall and provides an overall ranking. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

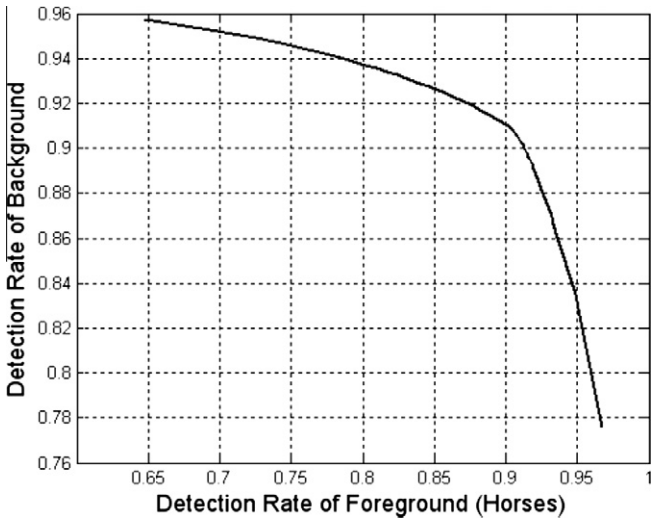


Fig. 11. Performance for the task of foreground/background segmentation on the Weizmann horse database.

There are other works [3,7,12,21,25,28,31] in the literature tackling similar problem. The most comparable approach is the work of Ren et al. [28] which gave detailed performance evaluations for combining low-, mid-, and high-level information. Our results show improvements at all levels, as shown in Fig. 10. It is not straightforward though to make direct comparison with other works [3,7,21,25], because some of them [21] were not evaluated on large testing datasets, and the evaluation details are missing. Also, some approaches [21,25] used color images, which are less challenging than gray-scale images used in our work. Levin et al. [25] also assumed that the position of the object is roughly given while there is no such assumption in our work; Kumar et al. [21] only evaluated their algorithm on 5 images which is not as convincing as our evaluation results because we evaluated on 178 testing images. Moreover, the details of the performance evaluation in [3,21,25] are not so clear. In [7], the authors first get a shortlist of 10 candidates and manually pick the best one from the shortlist. Corso [6] used Boosting on Multilevel Aggregates (BMA) to add features into PBT classifier, and tested on the Weizmann horse dataset. Our approach is a simple and clear one, and the speed of our algorithm is about 1.5 min per image, while speed is not reported in most of the above works.

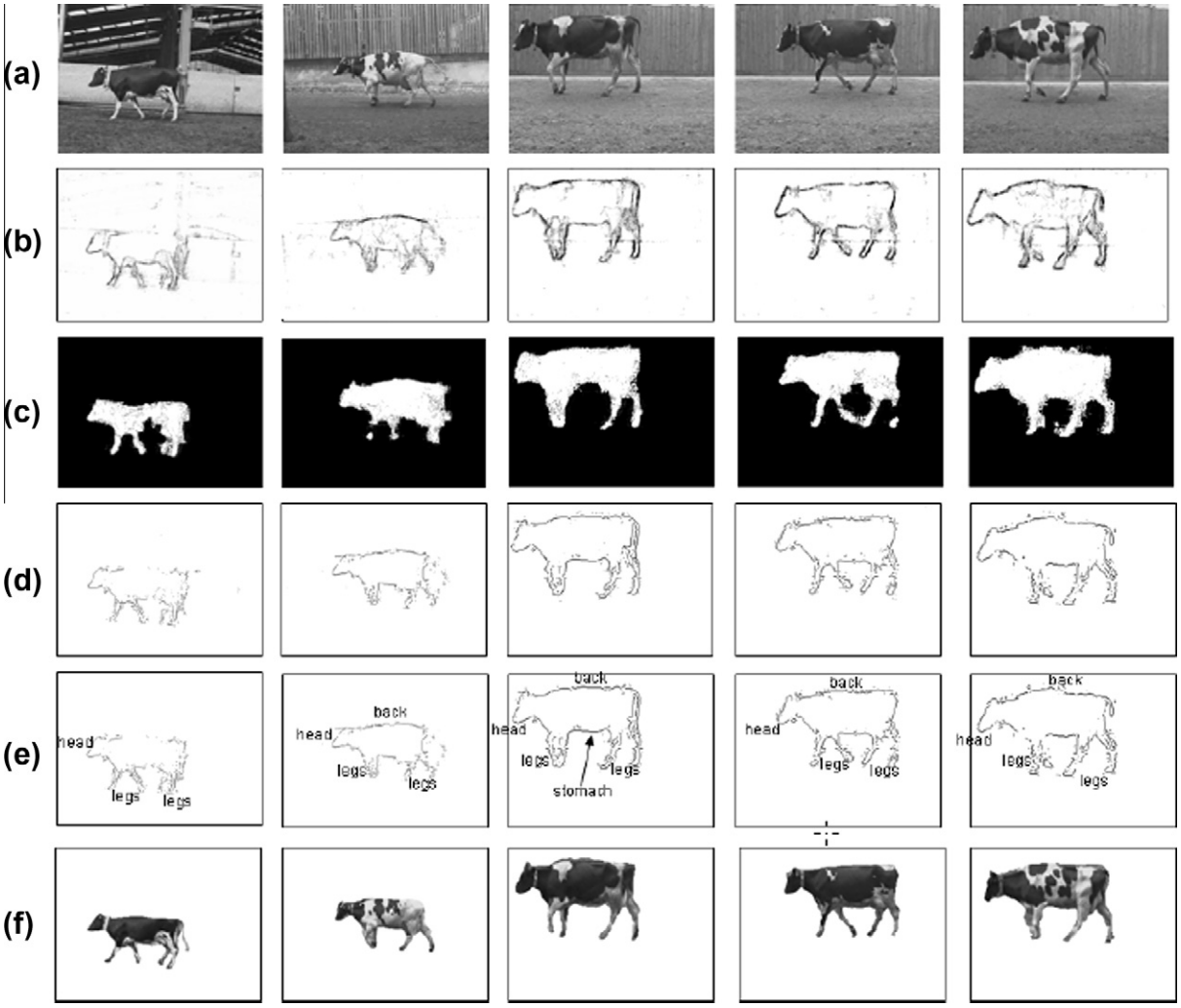


Fig. 12. Results on some test images from the cow dataset: (a) shows the input images; (b) shows the probability boundary maps; (c) shows the final body maps after refined by the high-level shape model; (d) demonstrates refined boundaries based on the short-range context information; (e) gives final results after shape matching, and also the different parts of the horses are labeled according to the shape matching results; and (f) gives the segmentation results. These images are representatives of the dataset, which have different appearances, poses, and background conditions.

Our approach can also be applied to color images by designing color-dependent features, and we expect the system performs better on color images than that on gray scale images since color images provide richer information.

6.2. Experiment on the ETHZ Cow Dataset

In this experiment, the system was trained on 40 randomly selected images from a dataset consisting 112 gray scale cow images [24], and we used the remaining images for testing. Compared with the Weizmann horse dataset, the ETHZ cow dataset has relatively less pose change, smaller pose variations, simpler texture properties, and less cluttered backgrounds.

We use the same process as in the horse segmentation case, with identical parameter settings. It took less training time due to less number of images used with relatively simpler foreground and background. Fig. 12 shows several typical images in this dataset, with the corresponding results computed.

Fig. 13 shows the precision–recall curve for detected boundaries. The performance on this dataset is better than that of the horse dataset. We observe that the short-range context information improves the curve significantly, and the high-level shape information further improves the performance.

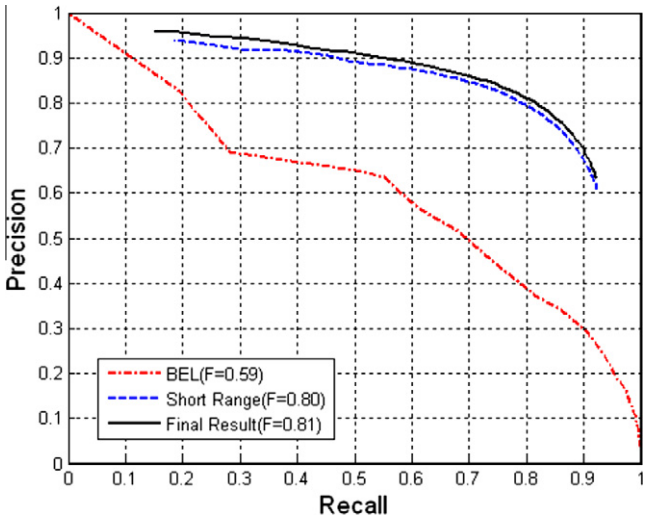


Fig. 13. Precision–recall curves for boundary detection of the cow testing images. The *F* value shown in the legend is the maximal harmonic mean of precision and recall and provides an overall ranking.

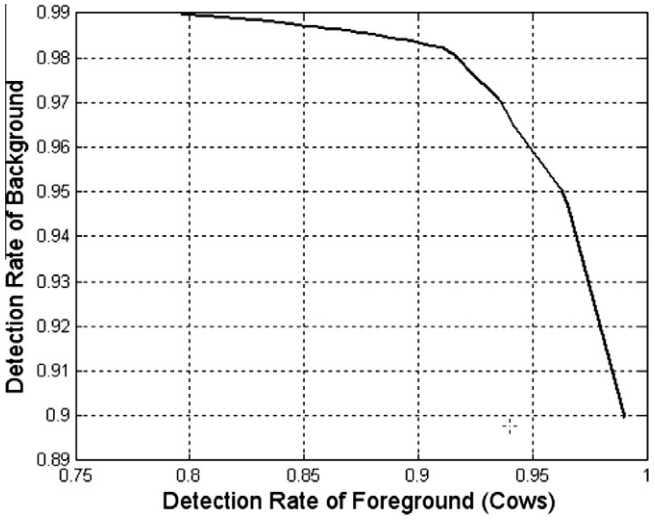


Fig. 14. Performance for the task of foreground/background segmentation on the cow dataset.

Fig. 14 shows the performance for the task of foreground/background segmentation on the cow dataset. As we can read from the curve, the performance is encouraging: we can achieve 95% accuracy of the foreground and 96% accuracy of the background .

The performance of OBJ CUT [21] on the ETHZ Cow Dataset was reported as 95.8% accuracy of classification rate for the foreground pixels and 98.7% accuracy for the background pixels. However, in [21], color information was used and the performance was evaluated only on 6 images in the testing stage. On the other hand, the proposed system was evaluated on 72 gray scale images in the testing stage. Levin et al. [25] reported the performance on cow data set about 92% of pixel classification accuracy. However, there is no accuracy for foreground pixel and background pixels. Furthermore, the size of the testing dataset in [25] is unclear.

7. Conclusions and discussions

In this paper, we have proposed a general learning based approach for object boundary detection and foreground/background segmentation. The algorithm described in this paper uses low-, mid-, and high-level cues. The proposed approach incorporates the low- and mid-level cues by a sequence of classifiers which requires only limited computation. This can be contrasted with alternative ways to introduce context which require sophisticated inference algorithms, – e.g., see [21,25,35], The learning processes relies on standard existing methods and the same approach is used for training all the low- and mid-level cues. Our experiments, on the Weizmann horse dataset and ETHZ cow dataset, show big improvement over many existing approaches. We also evaluate the effectiveness of each stage of our approach, which facilitates future research by identifying the importance of different cues.

The current limitations of our proposed approach are: it only works for single objects, the high-level model is not adequate to capture the bigger variations of the objects, and the link between low-, mid-, and high-level information is not yet fully clear.

Acknowledgment

This work is funded by Office of Naval Research Award, No. N000140910099 and NSF CAREER award IIS-0844566. We also thank the support from NIH U54 RR021813 and China Grant 863 2008AA01Z126. The second author would like to acknowledge funding support from NSF with grants IIS-0917141 and 0613563 and from AFOSR FA9550-08-1-0489. We thank X. Ren and C. Fowlkes for giving many constructive suggestions in carrying out the experimental evaluations.

References

[1] P. Awasthi, A. Gagrani, B. Ravindran, Image modeling using tree structured conditional random fields, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007.
[2] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (24) (2002) 509–522.
[3] E. Borenstein, E. Sharon, S. Ullman, Combining top-down and bottom-up segmentation, in: Proceedings of IEEE Workshop on Perceptual Organization in Computer Vision, June 2004.
[4] J.F. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.
[5] H. Chui, A. Rangarajan, A new point matching algorithm for non-rigid registration, Comp. Vis. Image Und. 89 (2003) 114–141.
[6] J.J. Corso, Discriminative modeling by boosting on multilevel aggregates, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[7] T. Cour, J. Shi, Recognizing objects by piecing together the segmentation puzzle, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, June 2007.

[8] J. Davis, M. Goadrich, The relationship between precision–recall and ROC curves, in: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, 2006.
[9] S. Dickinson, A. Pentland, A. Rosenfeld, From volumes to views: an approach to 3-D object recognition, Comp. Vis. Image Und. 55 (2) (1992).
[10] B. Dubuc, S.W. Zucker, Complexity, confusion, and perceptual grouping, Int. J. Comp. Vis. 42 (2001) 55–82.
[11] P. Dollár, Z. Tu, S. Belongie, Supervised learning of edges and object boundaries, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2006.
[12] V. Ferrari, L. Fevrier, F. Jurie, C. Schmid, Groups of adjacent contour segments for object detection, IEEE Trans. Pattern Anal. Mach. Intell. 30 (1) (2008) 36–51.
[13] Y. Freund, R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: Proceedings of International Conference on Machine Learning, 1996.
[14] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2003.
[15] T. Tversky, W.S. Geisler, J.S. Perry, Contour grouping: closure effects are explained by good continuation and proximity, Vis. Res. 44 (2004) 2769–2777.
[16] X. He, R.S. Zemel, M.A. Carreira-Perpinan, Multiscale conditional random fields for image labelling, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2004.
[17] X. He, R.S. Zemel, D. Ray, Learning and incorporating top-down cues in image segmentation, in: Proceedings of the European Conference on Computer Vision, May, 2006.
[18] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, Int. J. Comp. Vis. 1 (4) (1988) 321–332.
[19] K. Koffka, Principles of Gestalt Psychology, Lund Humphries, London, 1935.
[20] S. Konishi, A.L. Yuille, J.M. Coughlan, S. Zhu, Statistical edge detection: learning and evaluating edge cues, IEEE Trans. Pattern Anal. Mach. Intell. 25 (1) (2003) 57–74.
[21] M.P. Kumar, P.H.S. Torr, A. Zisserman, OBJCUT, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2005.
[22] S. Kumar, M. Hebert, Discriminative random fields: a discriminative framework for contextual interaction in classification, in: Proceedings of International Conference on Computer Vision, 2003.
[23] S. Kumar, M. Hebert, A hierarchical field framework for unified context-based classification, in: Proceedings of International Conference on Computer Vision, 2005.
[24] B. Leibe, A. Leonardis, B. Schiele, Combined object categorization and segmentation with an implicit shape model, in: Proceedings of Workshop on Statistical Learning in Computer Vision, Prague, Czech Republic, May 2004.
[25] A. Levin, Y. Weiss, Learning to combine bottom-up and top-down segmentation, in: Proceedings of the European Conference on Computer Vision, May 2006.
[26] D. Marr, Vision, W.H. Freeman and Co., San Francisco, 1982.
[27] D. Martin, C. Fowlkes, J. Malik, learning to detect natural image boundaries using local brightness, color and texture cues, IEEE Trans. Pattern Anal. Mach. Intell. 26 (5) (2004) 530–549.
[28] X. Ren, C. Fowlkes, J. Malik, Cue integration in figure/ground labeling, in: Proceedings of Advance in Neural Information Processing Systems, 2005.
[29] M.G. Ross, L.P. Kaelbling, Learning static object segmentation from motion segmentation, in: 20th National Conference on Artificial Intelligence, July 2005.
[30] C. Rother, V. Kolmogorov, A. Blake, GrabCut: interactive foreground extraction using iterated graph cuts, ACM Trans. Graph. (SIGGRAPH'04), 2004.
[31] J. Shotton, A. Blake, R. Cipolla, Multi-scale categorical object recognition using contour fragments, IEEE Trans. Pattern Anal. Mach. Intell. 30 (7) (2008) 1270–1281.
[32] C. Taylor, D. Kriegman, Structure and motion from line segments in multiple images, IEEE Trans. Pattern Anal. Mach. Intell. 17 (11) (1995) 1021–1033.
[33] A. Torralba, K.P. Murphy, W.T. Freeman, Contextual models for object detection using boosted random fields, in: Proceedings of Advance in Neural Information Processing Systems, 2005.
[34] Z. Tu, Probabilistic boosting-tree: learning discriminative models for classification, recognition, and clustering, in: Proceedings of International Conference on Computer Vision, 2005.
[35] Z. Tu, X. Chen, A. Yuille, S.C. Zhu, Image parsing: unifying segmentation, detection, and object recognition, Int. J. Comp. Vis. 63 (2) (2005) 113–140.
[36] Z. Tu, A. Yuille, Shape matching and recognition–using generative models and informative features, in: Proceedings of European Conference on Computer Vision, 2004.
[37] S. Ullman, R. Basri, Recognition by linear combinations of models, IEEE Trans. Pattern Anal. Mach. Intell. 13 (10) (1991) 992–1006.
[38] P.A. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comp. Vis. 57 (2) (2004) 137–154.
[39] Y. Wang, Q. Ji, A dynamic conditional random field model for object segmentation in image sequences, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, June 2005.
[40] Y. Wang, K.F. Loe, J.K. Wu, A dynamic conditional random field model for foreground and shadow segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2) (2006) 279–289.
[41] L. Zhu, Y. Chen, A.L. Yuille, Unsupervised learning of a probabilistic grammar for object detection and parsing, Adv. Neural Inform. Process. Syst. (2007).
[42] L. Zhu, Y. Chen, X. Ye, A. Yuille, Structure-perceptron learning of a hierarchical log-linear model, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[43] S. Zhu, Y. Wu, D. Mumford, Minimax entropy principle and its application to texture modeling, Neural Comput. 9 (8) (1997) 1627–1660.