Pattern Recognition 46 (2013) 2874–2889

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Automatic salient object extraction with contextual cue and its applications to recognition and alpha matting

Jianru Xue [a,*], Le Wang [a], Nanning Zheng [a], Gang Hua [b]

[a] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China
[b] Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ, USA

ARTICLE INFO

Article history:
Received 4 October 2012
Received in revised form
18 February 2013
Accepted 26 March 2013
Available online 11 April 2013

Keywords:
Salient object
Object extraction
Graph cut
Visual attention
Visual context

ABSTRACT

A method for automatically extracting salient object from a single image is presented in this paper. The proposed method is cast in an energy minimization framework. Unlike that only appearance cues are leveraged in most previous methods, an auto-context cue is used as a complementary data term. Benefitting from a generic saliency model for bootstrapping, the segmentation of the salient object and the learning of the auto-context model are iteratively performed without any user intervention. Upon convergence, the method outputs not only a clear separation of the salient object, but also an auto-context classifier which can be used to recognize the same type of object in other images. Our experiments on four benchmarks demonstrated the efficacy of the added contextual cue. It is shown that our method compares favorably with the state-of-the-art, some of which even embraced user interactions. Furthermore, we present some initial recognition results from the induced auto-context model and also show that the segmentation produced by our approach could serve as a good initialization for alpha matting.

1. Introduction

The problem of extracting a foreground object from a single image has wide applications in both computer vision and computer graphics. The vast majority of recent research on this topic had adopted an energy based binary segmentation method, which is defined in a standard form as a minimization of the energy formulation [1–4].

Typical energy formulation adopts either the Markov Random Fields (MRFs) [1] or Conditional Random Fields (CRFs) [5], which is subsequently optimized by energy minimization algorithm such as belief propagation [6] and its variants [7], or graph-cuts [2,4]. The energy formulation often incorporates a *data term* that models the appearances of the foreground and background, and a *spatial prior term* that is often intended to re-enforce the smoothness of the labels. Nevertheless, most of the previous energy formulations for image segmentation only model the appearance cues such as color or texture in their data term, which neglected valuable high level information such as visual context, and thus ambiguous segmentation is often observed.

The visual context of objects is the glue that binds objects in coherent scenes, and can be referred to as Gestalt laws in middle level knowledge regarding intra-object configurations and inter-object relationships [8,9]. One simplest form of context information is co-occurrence context, i.e., a co-occurrence frequency of a pair of objects [10,11]. In addition to co-occurrence context, many approaches take into account the spatial relationships between objects [12,13]. Although contextual information has been extensively studied for object category recognition [11], its efficacy is less explored in the context of foreground object extraction. Intuitively, the visual context should provide beneficial and complementary information for separating a foreground object from its background. This motivated us to explore the usage of context information for the task of automatic salient object extraction from a single image.

In this paper, we incorporate the auto-context model by Tu [14] into an energy minimization formulation to improve both the efficiency and accuracy for salient foreground object segmentation. The auto-context model builds a multi-layer Boosting classifier on image features and context features surrounding a pixel to predict if this pixel is associated with the target concept, where subsequent layer is working on the classification maps from the previous layer. Hence through the layered learning process, it automatically takes more spatial context into consideration when classifying one pixel.

Learning both the appearance model and the auto-context model of the foreground/background necessitates a set of labeled

* Corresponding author. Tel./fax: +86 29 82668672.
E-mail addresses: jrxue@mail.xjtu.edu.cn (J. Xue), wangleabc@gmail.com (L. Wang), nnzheng@mail.xjtu.edu.cn (N. Zheng), ganghua@gmail.com (G. Hua).
URL: http://gr.xjtu.edu.cn/web/jrxue (J. Xue).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.03.028

image pixels with both negative and positive examples. Some previous approaches have either resorted to user interactions to provide such labels, such as GrabCut [3] and Lazy Snapping [1], or made assumptions on the location of the object of interest [15]. Notwithstanding the efficacy of user interaction, it is still desirable to have a fully automated system to extract the salient object from a single image.

The notion of a "salient" object could have multiple implications. In this paper, we regard a visual object to be salient if it accounts for a significant portion of the image. To achieve a fully automated system, we resort to a graph-based computational attention model [16] to bootstrap our energy minimization process. This overcomes the original auto-context algorithm's restriction [14] of off-line training with a set of images with labeled ground-truth. We employ the saliency map to generate an initial segmentation, which is subsequently used to train the appearance model and the first-layer Boosting classifier of the auto-context model. Once these models are obtained, we use the implementation of the max-flow algorithm in [17] to produce a new segmentation.

The new segmentation then serves to re-estimate the appearance model and the subsequent layer of Boosting classifier of the auto-context model. This process iterates until convergence, which returns not only an automatic segmentation of the salient object, but also a fully trained auto-context model. This automatically learned context model can indeed be applied to recognize the same type of object in new images. The segmentation process can also produce a good trimap for alpha matting [18].

In our formulation, we also utilize the visual saliency cue to compute the weights for fusing a set of low-level features to form our appearance model, where the weight of each feature is in proportion to its contribution to the saliency map. This feature fusion process is shown to produce a more robust appearance model of the foreground object. In summary, the key contributions of this paper are on the following aspects:

- An automatic segmentation method that can perform segmentation of the object of interest from its background without any user intervention.
- A unified energy minimization formulation which leverages saliency cue, appearance cue, and contextual cue in the data term.
- An iterative algorithm to jointly estimate the segmentation of the foreground object, and learn the auto-context model which can be used to recognize the same type of object in new images.

We demonstrate our algorithm on challenging foreground object extraction tasks. Four well-known datasets including the Berkeley segmentation dataset [19], the GrabCut dataset [3], the Weizmann horse segmentation dataset [20], and the MSRC dataset [9], are used to test the algorithm. The results show the advantages of our method when compared with the state-of-the-art. In addition, we present some initial recognition results from the induced auto-context model and also show that the segmentation produced by our approach could serve as a good initialization for alpha matting.

This paper is an extended version of our previous work in [21]. We have extended it by including our recent theoretical and experimental improvements. We have expanded and structured the related works, and have analyzed and discussed the convergence of the proposed iterative algorithm for the employed energy minimization framework. Moreover, the efficiency of the method and alternatives to cut down complexity have been discussed. Finally, we have added experiments to assess the performances of the method in the context of both segmentation and image matting.

The rest of the paper is organized as follows. Immediately below, we discuss related work in Section 2. Our energy formulation incorporated with contextual cue is presented in Section 3. We present the iterative optimization process in Section 3.2, the context model in Section 4, the appearance model in Section 4.2, and convergence analysis in Section 5. Extensive experimental results and discussions are given in Section 6. Finally, we conclude in Section 7.

2. Related work

In this section, we summarize related work in three areas, including research on energy formulation, visual context modeling, and joint segmentation and recognition.

2.1. Energy formulation

Salient object extraction can be posed as an energy based binary labeling problem to assign a unique label to each pixel (belonging to the object or background).

There have been many energy formulations for figure/ground segmentation developed in the past. The models adopt either MRF [1,22] or CRF [5,23]. The optimal labeling can be obtained by inference algorithms range from belief propagation [6] to graph cut [4]. Although these formulations have produced encouraging results, there are drawbacks. For example, these energy functions only model simple appearance cues such as color or texture in their data term, which neglect valuable high level information such as visual context. To alleviate these problems, interactive graph cuts algorithms have been proposed in [2,3,24,25], and achieved encouraging results. Notwithstanding the efficacy of user interaction, it is still desirable to have a fully automated system to extract the foreground object from a single image.

This motivated us to make an effort to develop a fully automated foreground object extraction method. We use an auto-context cue as a complementary data term to introduce high level information in the energy formulation. We also adopt the notion of saliency-based proto-object [26] as the initial segmentation of the model to alleviate the problem that optimization process is sensitive to the initial segmentation. The multi-layer classifier learns the foreground and background distributions of a single image by beginning with the initial segmentation. As the algorithm iterates, the object is better segmented out, and the distributions are better learned. No additional images with ground-truth figure/ground segmentation are needed to train our model to conduct the segmentation. This makes our algorithm different from Tu's auto-context algorithm [14], which iteratively learns their model from a set of training images with pixel-wise ground-truth labels.

2.2. Visual context model for segmentation

The idea of adopting context information into segmentation can be traced back to Rosenfield, Zucker, and Humel's pioneering work on relaxation labeling [27–29]. It has become popular and more tractable due to the advancements in machine learning and statistics in recent years [21,30,31]. Borenstein et al. [20,32] combine top–down information (configuration learned on image patches) and bottom–up approaches (segmentation based on intensity) for figure/ground segmentation. Fergus et al. [33] build a top–down model based on features extracted by interest point operators. CRF models [34,35] have been used to enforce local consistency between neighboring structures. Combining both

top–down and bottom–up learning are advocated in [36]. OBJCUT [37] combines different levels of information and performs segmentation by graph cuts. Zhu et al. [38] present a stochastic grammar to incorporate semantic and context information at different levels. Auto-context model [14] iteratively uses classification result to improve the performance of figure/ground segmentation in an elegant way. Shotton et al. [39] extend the decision forest classifier to directly enforce local consistency and semantic context.

These aforementioned approaches have shown promising results by combining low-level, mid-level, and context information into segmentation. Our work adopted an online layered learning approach to combine local information and context information both implicitly and explicitly. Unlike the previous work that have employed contextual cue to improve the performance of classifiers [14], we have explored contextual and saliency cues in an energy minimization framework for automatic extraction of object of interest. Benefitting from a generic saliency model for bootstrapping, the segmentation of the salient object and the learning of the auto-context model are iteratively performed without any user intervention. Therefore, our approach is fully automatic.

2.3. Joint segmentation and recognition

The fields of object recognition and segmentation have been very active in recent years. However, many works have considered these two tasks separately. It is often observed that segmentation can be ambiguous, if not impossible, in the absence of the contextual information provided through recognition.

Joint segmentation and recognition of a single object class have been achieved by several approaches [9,23,40]. Typically, these approaches exploit a global layout model, and only address highly structured object classes. He et al. [40] first segment images by a bottom–up algorithm to produce 'superpixels', and then merge 'superpixels' together and semantically label them using a combination of several scene-specific CRF models. Their work used Gibbs sampling for both parameter estimation and label inference. Their more recent work [23] incorporates both regional and global features into a CRF model to model layout and context. Further state-of-the-art techniques along this line of thinking include TextonBoost [9] and semantic texton forests [41]. The first one learns a discriminative model of object classes by incorporating texture, layout, and context information efficiently. The learned model is used for automatic visual understanding and semantic segmentation of images. The second one uses ensembles of decision trees that act directly on image pixels, and thus are fast to both train and test.

The performance of most of the aforementioned techniques depends on a discriminative model learned off-line, and they almost always need a large labeled dataset. However, our method is completely automatic and easy to use. It can automatically extract an object of interest from a single image without any additional labeled dataset and off-line learning process.

3. Problem formulation

In this section, we propose an energy minimization formulation for salient object extraction. We first present the energy formulation with contextual cue and then briefly discuss the iterative optimization algorithm.

3.1. Energy formulation with contextual cue

For an image $\mathbf{I}$, each pixel $p \in \mathbf{I}$ will be assigned a binary label $L_p \in \{0,1\}$, where 0/1 corresponds to the background/foreground, respectively. Our objective is to identify a labeling L that minimizes $E(L)$, i.e.,

$$E(L) = \lambda \sum_{p \in \mathbf{I}} (C_p(L_p) + D_p(L_p)) + \sum_{(p,q) \in N} \omega_{pq} \delta(L_p, L_q), \quad (1)$$

where $C_p(L_p)$ measures the cost of labeling pixel p to be L_p from an auto-context model, and $D_p(L_p)$ for the cost from an appearance model. The sum of $C_p(L_p)$ and $D_p(L_p)$ composes the *data term* in our model, which encodes regional properties of the salient object. The function $\delta(L_p, L_q)$ is a Dirac delta function. In our implementation, ω_{pq} is computed based on the edge probability map from the Berkeley boundary detection system [19], which incorporates texture, luminance, and color cues. The difference is that they [19] measure the dissimilarity of adjacent pixels, while ours measure the similarity between adjacent pixels p and q. Naturally, $\omega_{pq}\delta(L_p, L_q)$ composes the *spatial prior term* to encourage piecewise smooth labeling L, which captures boundary properties of the salient object. The coefficient $\lambda \geq 0$ controls the relative weight of the data term and the spatial prior term.

The particular contextual model we adopted is the auto-context model proposed by Tu [14], which seeks for a multi-layer Boosting classifier, with subsequent Boosting classifier working on the classification maps of the previous layer. Hence, $C_p(L_p)$ relies on the discriminative probability (or classification) maps created by a learned auto-context classifier. If we know the auto-context model $C_p(L_p)$ and the appearance model $D_p(L_p)$, the energy function $E(L)$ can be efficiently solved by leveraging graph-cut [17] to obtain an optimal solution. However, these models often need to be pre-trained with a set of labeled training examples, which is not always available. This is why many previous methods even resort to user interactions to obtain the model.

Our target is a fully automatic system, so we need to jointly estimate the segmentation, the auto-context model, and the appearance model. Intuitively, an iterative optimization process is needed. We initialize such an iterative algorithm by obtaining the very initial segmentation by using a bottom–up visual saliency model [16].

We proceed to present our iterative algorithm in Section 3.2. Our auto-context model $C_p(L_p)$ and its iterative estimation will be presented in Section 4.1. The detailed representation and the iterative updating scheme of our appearance model $D_p(L_p)$ will be discussed in Section 4.2, which is fused from a number of low-level features [42].

3.2. Iterative optimization

Bootstrapped by a visual saliency model [16], in each iteration of our algorithm, we first update the auto-context model, the appearance model and the spatial prior term, and then minimize the energy via graph-cut [2,4]. This process iterates until it converges, as illustrated in Fig. 1.

3.2.1. Generate the initial region

We use a bottom–up visual saliency model, namely the GBVS [16] to locate and create the most salient region of the image. The saliency map is generated by combining multi-scale image features including color, intensity, and orientation into a single topographical saliency map [26]. Since the most salient region is often associated with the most salient object, we select it as the initial region of the salient object.

Obviously, a good initial segmentation is necessitated to obtain a good final segmentation result. Then the question becomes: *how can we measure the quality of an initial segmentation?* We identified three useful measures: (1) **Connectivity**, which requires the initial region to be a single region with closed contour. (2) **Convexity**, which requires the contour of the region to be convex. We adopt the algorithm in [22] to compute the "convex" measure, as

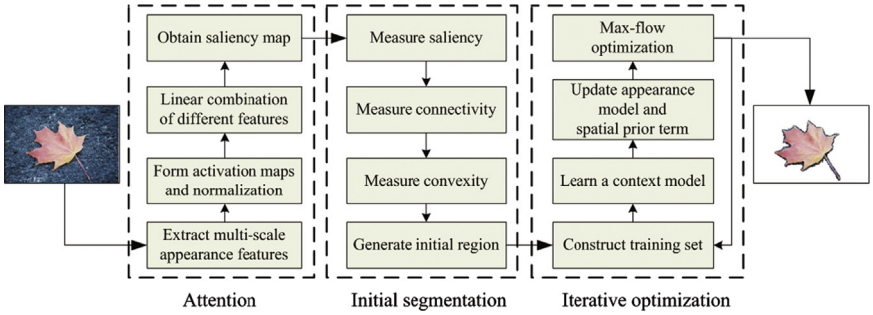

Fig. 1. The flow chart of our method.

Fig. 2. Generation of an initial segmentation. Left: the input image. Middle: saliency map generated by GVBS [16]. Right: illustration of how to calculate the convexity measure of an initial segmentation of the salient object. Specifically, (1) calculate the center of the initial region of the salient object, e.g., the dot c marked with red; (2) shot straight lines passing through c; (3) for any point p marked with green dot on the line inside the object, just assess whether any point q also marked with green dot on the straight line connecting c and q is also inside the object. If so, the convexity is satisfied. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

illustrated in Fig. 2. (3) **Saliency**, which indicates that the most salient region shall be more likely to be a good initial segmentation.

We design an adaptive selection mechanism to select the minimal connected region of the saliency map as the initial segmentation of the salient object to satisfy these three measures. Specifically, the region in the saliency map with equal or greater saliency than a threshold (e.g., 90% of the maximal saliency value of the saliency map) is firstly selected as a candidate region. This will first meet the "saliency" measure. Secondly, if the candidate region satisfies the "connectivity", and to some extent meet the "convexity" measure, we then use it as the initial region of the salient object. Otherwise the candidate region is discarded, and we reduce the threshold by 5% of the maximal saliency. The aforementioned procedure is repeated until a minimal region is found. In Fig. 2, we presented the procedure of generating an initial segmentation.

3.2.2. *Iterative process*

By treating the identified salient region as the foreground, and the rest of the image as the background, the iterative process is started and performed until convergence. In each iteration, the auto-context model and the appearance model of the energy in Eq. (1) are updated separately based on the segmentation and the discriminative probability maps of the previous iteration, where the discriminative probability maps are estimated by the auto-context model.

The spatial prior term in Eq. (1) can be calculated directly. The spatial prior term consists of two items: (1) ω_{pq} which is the edge probability map of the input image; and (2) $\delta(L_p, L_q)$ which depends on the segmentation from previous iteration. ω_{pq} can be computed before the iteration, and is fixed during the iterative process, while $\delta(L_p, L_q)$ changes along with the segmentation.

To have a clear understanding of the algorithm, we defer the details of how to update the auto-context and appearance models to Sections 4.1 and 4.2, respectively. The updated energy is then minimized by performing a max-flow algorithm [17] twice to take both the shrinkage and expansion of the salient object into account.

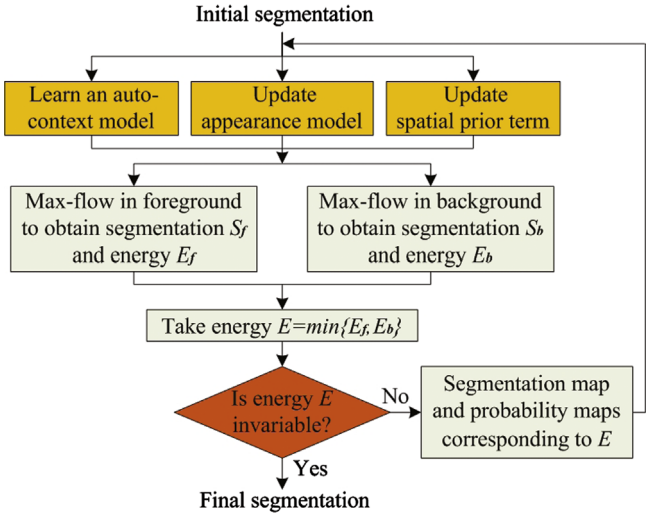

Fig. 3. The iterative process of the energy minimization.

Theoretically, shrinkage and expansion are equivalent to two $\alpha-$expansions [43] and empirically help to converge faster. The first max-flow algorithm is performed within the foreground region, which means that only pixels belonging to foreground are allowed to flip their labels. A new segmentation map S_f and the corresponding energy E_f are obtained. This $\alpha-$expansion actually grabs space from the foreground. The second max-flow algorithm is performed in the background, only background pixels are allowed to flip their labels. A new segmentation S_b and the corresponding energy E_b are also obtained. This $\alpha-$expansion actually grabs space from the background. We then choose the one with lower energy from these two segmentations to be the output of the current iteration. This process iterates until the energy becomes invariable. We then take the final segmentation as our ultimate result. Fig. 3 illustrates the iterative process of segmentation.

The two max-flow optimizations within each iteration are performed in parallel. We then choose the one with a smaller energy compared with the other one. One alternative is to perform these two max-flow optimizations in a sequential way, i.e. by alternating between a foreground shrinkage and expansion. Actually, these two ways have achieved similar segmentation results in our experiments. However, we observed that the parallel one has a faster speed than the sequential one in the vast majority of our experiments, or at least, it is as fast as the sequential update in quite few experiments, when it is implemented in a computer with a multi-core CPU. Thus, we prefer to the parallel one in our experiments.

4. Appearance and context model learning

In this section, we present the auto-context model, the appearance model, and their updating process.

4.1. Auto-context model

Contextual dependency is one major visual cue in segmentation, which has not been well explored by previous energy minimization formulation. To better determine how fit a pixel belongs to foreground or background by including a large amount of contextual information, we resort to learn an auto-context model iteratively (for details about the auto-context model itself, refer to Tu [14]), adapting each iteration of the energy minimization in Eq. (1). To achieve this in the iterative learning process, we firstly define an adaptive sampling strategy to collect samples for training the auto-context model, and subsequently use it to update the model.

4.1.1. Sampling structure

As illustrated in Fig. 4, with a given segmentation map and the discriminative probability maps from the previous iteration, a training set for the auto-context model is formed by including a large amount of patches centered at each pixel location of the discriminative probability maps along with its label. It consists of two sets of patches: a set of positive patches which are centered around foreground pixel locations, and a set of negative patches centered around background pixel locations (which are those locations far from the foreground region).

Instead of using all pixels around a pixel location of interest to extract the patches, we define a multi-scale sampling structure for each pixel location, and sample patches along the circles of the structure to form the training set. For each pixel location, the sampling structure of patches first includes the pixel locations within 3 pixels away from the current pixel location; and further, circles centered at the current pixel location with different radiuses are built, and patches are sparsely sampled on these circles along 8 rays in 45° intervals, as shown in Fig. 4.

Our method is fundamentally different from other methods [9,14,41] using contextual cues. Specifically, contextual cues of other methods [9,14,41] are learned from pre-labeled training data (usually multiple images) or derived under strong constraint prior, while our context information is directly derived from the single query image without pre-labeled training data or strong constraint prior.

4.1.2. Update the auto-context model

In the first round of the iterative learning process, the training patches set for the auto-context model is constructed as

$$S_1 = \{(L_p, P(N_p)), p = 1, \ldots, n\},$$

where n is the number of pixel samples, $P(N_p)$ denotes the local image patch centered at pixel p (we use local image patches of fixed size 11×11). The haar features are then extracted from this patch to form the appearance feature vector of the current pixel p.

After the first classifier is learned on the appearance feature vector extracted from the local image patch $P(N_p)$, the discriminative probabilities $\mathbf{p}_p^{(1)}$ for each pixel p on the discriminative probability maps $\mathbf{P}^{(1)} = \{\mathbf{p}_p^{(1)} | p \in \mathbf{I}\}$ output by the learned classifier are used as contextual cue (individual probabilities or the mean

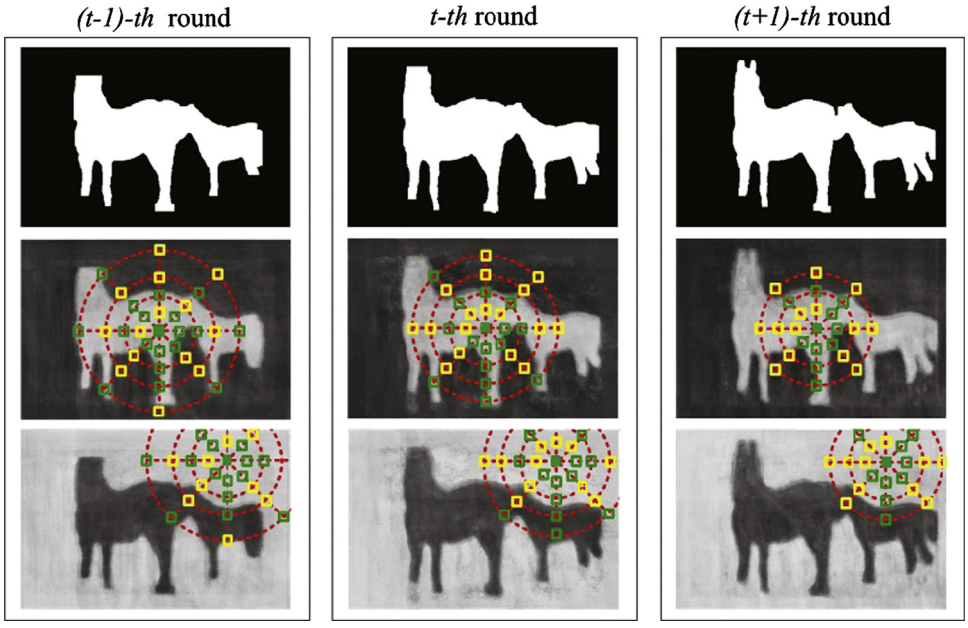

Fig. 4. Collecting training samples with the multi-scale sampling structure on three consecutive rounds of the iterative learning process of the auto-context model. The first row lists segmentation maps, the 2nd and 3rd rows list discriminative probability maps. The patches in yellow are uniformly sampled along circles. The patches in green are selected context features to construct the strong classifier. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

probability within a 3×3 patch). The auto-context model $C_p(L_p)$ in Eq. (1) for pixel p is then updated as

$$C_p^{(1)}(L_p) = \mathbf{p}_p^{(1)} = p(L_p|P(N_p)),$$

$$\sum_{L_p} C_p^{(1)}(L_p) = 1,$$

$$\forall p \in \mathbf{I}. \tag{2}$$

From the second round of the iterative learning process, we construct the training patch set as

$$S_2 = \{(L_p, (P(N_p), O^{(1)}(p))),\ p = 1, \ldots, n\},$$

where $O^{(1)}(p)$ are patches on the sampling structure centered at pixel p, which is sampled from the discriminative probability maps $\mathbf{P}^{(1)} = \{\mathbf{p}_p^{(1)} | p \in \mathbf{I}\}$ of the previous round. $P(N_p)$ is the local image patch centered at pixel p as before. As discussed above, $O^{(1)}(p)$ is a collection of patches of the sampling structure sampled on the discriminative probability maps which are obtained from previous round, and hence can be expressed as

$$O^{(1)}(p) = \{O(p, r_i, \theta_j),\ i = 1 \ldots N_r,\ j = 1 \ldots N_\theta\},$$

where r_i and θ_j denote the radius and angle, i and j are the indices of radiuses and angles, N_r and N_θ are the total numbers of radiuses and angles, respectively. $O(p, r_i, \theta_j)$ is the patch with radius r_i and angle θ_j away from the pixel p. More specifically, $r_i \in \{1, 3, 5, 7, 10, 12, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 125, 150, 175, 200\}$. At the beginning of the iteration, N_r is initialized to be 23 while r_i varies from 1 to 200 for all values of r_i. With the progress, N_r is gradually reducing and remains unchanged until it reaches a minimum (4 or 5), and the N_θ is fixed to be 8 in 45° intervals.

Then, a new classifier is trained on the probabilities of patches of the sampling structure $O^{(1)}(p)$ sampled on the discriminative probability maps $\mathbf{P}^{(1)} = \{\mathbf{p}_p^{(1)} | p \in \mathbf{I}\}$, and on the appearance features extracted from the local image patch $P(N_p)$. The auto-context model is then updated as follows:

$$C_p^{(2)}(L_p) = \mathbf{p}_p^{(2)} = p(L_p|P(N_p), O^{(1)}(p)),$$

$$\sum_{L_p} C_p^{(2)}(L_p) = 1,$$

$$\forall p \in \mathbf{I}, \tag{3}$$

where $\mathbf{p}_p^{(2)}$ denotes the discriminative probabilities on the new discriminative probability maps $\mathbf{P}^{(2)} = \{\mathbf{p}_p^{(2)} | p \in \mathbf{I}\}$ created by the new learned classifier.

This process will iterate until it converges where the discriminative probability maps are not changing anymore. Table 1 presented a summarization on one round of the iterative process of learning the auto-context model. Indeed, in our formulation, the auto-context model is iteratively updated seamlessly with the iterative minimization of the energy in Eq. (1).

It should be noted that the sampling structure of the auto-context model on the discriminative probability maps is in a contractive fashion, i.e., at the beginning of the iteration, the sampling structure is large to cover the boundary of the object and then it reduces as the segmentation map is better and better. At the beginning, large sampling steps are taken to generate the patches due to less accuracy of the training labels (i.e. segmentation map), where r_i may take the maximum radius 200, but with the progress, sampling step is reduced to find fine-grained boundary of the foreground object, which largely avoided the impact of background clutter. Fig. 4 clearly illustrated this point on consecutive three rounds of the iterative learning process.

4.2. Appearance model

Our appearance model $D_p(L_p)$ in Eq. (1) fuses color and intensity as

$$D_p(L_p) = \omega_i D_p^i(L_p) + \omega_c D_p^c(L_p), \tag{4}$$

where $D_p^i(L_p)$ and $D_p^c(L_p)$ are simply the intensity distribution and color distribution, while ω_i and ω_c specify the importance of intensity cue and color cue in composing the appearance model $D_p(L_p)$, respectively.

The appearance model is also updated in each iteration of the energy minimization. As a segmentation obtained from the previous iteration, intensity distribution $Pr(I_p|''Fg'')$ for the foreground and $Pr(I_p|''Bg'')$ for the background can be approximated by histograms of intensities of pixels belonging to the foreground and background, respectively. These histograms are then used to calculate $D_p^i(L_p)$, i.e.,

$$D_p^i(L_p = 1) = -\ln(Pr(I_p|''Fg'')),$$
$$D_p^i(L_p = 0) = -\ln(Pr(I_p|''Bg'')), \tag{5}$$

where I_p is the intensity of pixel p. Similarly, the term $D_p^c(L_p)$ is obtained as

$$D_p^c(L_p = 1) = -\ln(Pr(C_p|''Fg'')),$$
$$D_p^c(L_p = 0) = -\ln(Pr(C_p|''Bg'')), \tag{6}$$

where C_p is the RGB color vector of pixel p.

It is worth noting that, the weights ω_i and ω_c are adaptively selected based on the visual saliency model. First, color and intensity features are normalized to maintain the same dimension. Then the values of the weights ω_i and ω_c are assigned in proportion to their respective contributions to the saliency map generated by the bottom–up visual saliency model [16].

5. Optimality properties of the iterative algorithm

In this section, we first discuss the convergence of our iterative optimization algorithm, and then present an analysis of the efficiency of the proposed algorithm.

5.1. Convergence analysis

The energy $E(L)$ in Eq. (1) theoretically ensures that the better labeling L, the lower the energy $E(L)$. The algorithm can automatically terminate when $E(L)$ ceases to decrease significantly. It is

Table 1
Example of one round of the iterative process of learning the auto-context model.

Input: the input image, the segmentation map and the discriminative probability maps from previous iteration.
Output: the learned auto-context model at current *t-th* iteration.

- Construct a training set: $p = 1, \ldots, n$.

$S_t = \{(L_p, (P(N_p), O^{(t-1)}(p)))\}$

- Train a classifier on both image appearance and context features extracted from $P(N_p)$ and $O^{(t-1)}(p)$, respectively.
- Use the trained classifier to compute new discriminative probability maps $\mathbf{P}^{(t)} = \{\mathbf{p}_p^{(t)} | p \in \mathbf{I}\}$.

The output is exactly the auto-context model presented in Eq. (1) $C_p(L_p) = \mathbf{p}_p^{(t)} = p(L_p|P(N_p), O^{(t-1)}(p))$

known in [43] that an energy function which can be minimized via graph cuts algorithm depends on whether it is graph-representable or not. Our spatial prior term of the energy function $E(L)$ satisfies triangular inequality, and thus $E(L)$ is graph-representable. This property of $E(L)$ guarantees that it can be minimized using graph cuts algorithm.

The iterative process of the algorithm shown in Fig. 3 actually implements an alternative iteration optimization. Within each round of iteration, the auto-context model $C_p(L_p)$, the appearance model $D_p(L_p)$, and the spatial prior term are firstly updated with the segmentation result output from previous iteration. The updating of the data term and the spatial prior term in Eq. (1) re-estimate regional parameters of the salient object. Then the twice operations of the max-flow algorithms start the minimization process from the energy of the labeling L that produced from the previous iteration, and update boundaries of the salient object.

It should be noted that we currently cannot achieve theoretic guarantee of global convergence of our algorithm. However, each step within a round of iteration converges in theory. The updating processes of appearance model $D_p(L_p)$, and the spatial prior can be computed directly, and the convergence of the auto-context model learning $C_p(L_p)$ has been proved in [14]. The max-flow algorithm can achieve minimum of the energy given by labeling L in polynomial time. Moreover, we empirically observe that the iterative optimization always converges. Fig. 5 illustrates the trend of energy function tested on the 3 images in Fig. 8. According to the experimental results, each step of our energy minimization ensures that the energy in Eq. (1) is non-increasing. The curve in Fig. 5 shows that the energy function always converges within 30 iterations.

To provide a more intuitive discussion regarding the convergence of our algorithm, we may consider the energy function as it consists of two set of variables: (1) the labeling L, and (2) both variables of the auto-context model and the appearance model that are changed during the algorithm's iteration. In this way, the update of the auto-context model and the appearance model, and the twice operations of the max-flow algorithm, can be shown to be a minimization of the energy with respect to the two sets of variables alternatively. The total energy then deceases in an iterative way, and this trend is illustrated in Fig. 5. Furthermore, it is straightforward to detect when the energy ceases to decrease significantly, and to terminate iteration automatically. Thus, we often observe that the algorithm converges, or at least converges to a local minimum of the energy in experiments.

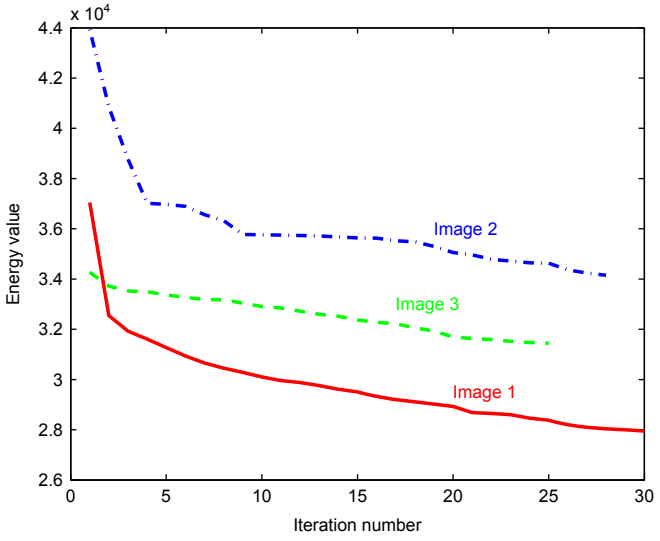

Fig. 5. Energy values in the iterative process of energy minimization. The energy values are obtained by testing the algorithm on the 3 images in Fig. 8.

Table 2
Processing time (in seconds) of our algorithm implemented with Matlab. The algorithm is tested with the horse image in Fig. 7, and which took 13 rounds of iteration to converge. The first column presents the round of iteration. The 2nd and the 3rd columns show the time used for the updating processes of models (including both appearance model and the auto-context model) and the max-flow algorithm, respectively. The 4th and the 5th columns show the time used for updating processes of context model and appearance model, respectively. The final row lists percentages of time used by each step.

Horse image	Model updating	Max-flow computing	Updating auto-context	Updating appearance
Iteation 1	16.073	0.0207	15.6115	0.463
Iteation 2	27.5418	0.0192	26.207	0.4046
Iteation 3	27.3271	0.0201	26.326	0.39
Iteation 4	27.484	0.0196	26.4487	0.3903
Iteation 5	26.2459	0.02	25.238	0.3923
Iteation 6	26.0696	0.0198	25.0992	0.3923
Iteation 7	26.4645	0.0199	25.4721	0.3891
Iteation 8	25.4455	0.0202	25.454	0.3916
Iteation 9	26.1831	0.0203	25.1621	0.3938
Iteation 10	26.7891	0.0205	25.7899	0.3926
Iteation 11	26.2214	0.0198	25.2024	0.3907
Iteation 12	26.4964	0.0199	25.5058	0.3896
Iteation 13	26.2044	0.0195	25.2006	0.3942
Percentage	99.92%	0.08%	96.08%	1.45%

5.2. *Efficiency discussion*

The convergence rate depends on the amount of energy reduced in each round of iteration of the algorithm. Each round of iteration consists of two major steps: (1) the update of the auto-context model, and the appearance model and the spatial prior term, (2) the twice operations of the max-flow algorithm. In the first step, the update of the appearance model and the spatial prior term in the first step can be computed in linear time, while the update of auto-context model has a similar computational complexity as that of the Adaboost algorithm [14], which is a little time consuming. In the second step, the max-flow algorithm can achieve minimum of the energy given by the labeling from the previous round of iteration in polynomial time.

Table 2 lists the processing time of our algorithm implemented in Matlab. The algorithm is tested with the horse image in Fig. 7, and it converges after 13 rounds of iterations. In Table 2, the 2nd and the 3rd columns list the time cost by the updating processes of models and max-flow computing, respectively. The 4th and the 5th columns, respectively, show the time cost of updating processes of auto-context and appearance. The final row presents each step's percentages of the total processing time. It clearly shows that auto-context learning is the bottleneck of the efficiency of the algorithm, even it brings significant performance increase in segmenting salient object. How to speed up auto-context model learning is out of scope of this paper, and it will be discussed in our future work.

We also studied to what extend different initial salient region generations, as well as the (weak) convexity requirement, affect our system's convergence. Empirically, we found that our fully automatic system is robust to initial salient region, as long as it is not totally off the target, which we almost never observe in our experiments.

6. Performance evaluation and extended applications

In this section, we first evaluate the performance of our method on four challenging image segmentation datasets, and compare our results with existing work. Then we present the performance of the learned auto-context classifier, as well as results when using our method as a frontend of an alpha image matting system.

Fig. 6. Segmentation results of our method. The rows 1, 3, 5, 7, and 9 are some test images from 4 segmentation benchmarks including the Berkeley [19], GrabCut [3], Weizmann [20], and MSRC database [9]. The rows 2, 4, 6, 8, and 10 are the corresponding segmentation results, respectively.

6.1. Performance evaluation

6.1.1. Segmentation accuracy

To evaluate the segmentation accuracy of our method, we test it on four image segmentation datasets including the Berkeley segmentation dataset [19], the GrabCut dataset [3], the Weizmann single object segmentation dataset [20], and the MSRC dataset [9]. All the experiments are done with a computer of 2.99 GHz CPU and 2.0 GB RAM. The running time ranges from a dozen of seconds to 3 min per image with our Matlab source code, depends on the resolution of the images used in the experiments.

Table 3
F-measures of our method by evaluating it on the MSRC dataset [9]. We choose 14 categories from the MSRC dataset.

Image category	F-measure score
aeroplane	0.8943 ± 0.0117
bird	0.9169 ± 0.0158
body	0.8389 ± 0.0227
book	0.9058 ± 0.0076
building	0.9165 ± 0.0103
car	0.8673 ± 0.0126
cat	0.8907 ± 0.0127
chair	0.8449 ± 0.0156
cow	0.9457 ± 0.0058
dog	0.9051 ± 0.0114
face	0.8876 ± 0.0087
flower	0.9263 ± 0.0078
sheep	0.9358 ± 0.0143
sign	0.9217 ± 0.0094
tree	0.8905 ± 0.0211

Table 4
Performance comparison of our method with other segmentation methods: F-measures of our method and 5 state-of-the-art segmentation algorithms by evaluating them on the Weizmann single object database [20].

Algorithms	F-measure score	Remarks
Our method	0.91 ± 0.013	Automatic
Our method without using auto-context cue	0.88 ± 0.011	Automatic
Unified approach [44].	0.87 ± 0.01	Interactive
Cues integration [20]	0.86 ± 0.012	Automatic
Texture segmentation [45].	0.83 ± 0.016	Automatic
Normalized cut [46].	0.72 ± 0.018	Automatic
Meanshift [47].	0.57 ± 0.023	Automatic

In Fig. 6, we illustrate some sample segmentation results. Empirical results show that our method is able to extract important part of some difficult salient objects, and is able to deal with weak boundaries and complex background without any user intervention. For more results, we highly recommend to check our supplementary video {pan.baidu.com/share/link?shareid=211535&uk=4097062518}.

To perform an objective evaluation of our algorithm, we also calculate F-measure score of our algorithm by evaluating it on both the MSRC dataset and the Weizmann single object segmentation dataset. The F-measure score is the harmonic mean of precision and recall measures calculated on the foreground pixels, i.e.,

$$F = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}. \quad (7)$$

Table 3 summaries the F-measure scores of our algorithm on MSRC dataset. The F-measure scores of our algorithm on the Weizmann single object segmentation dataset can be found in Table 4. Results in Table 3 clearly shows that the F-measure scores of our algorithm varies according to different categories of images. Thus, we do not show F-measure scores of our algorithm on the Berkeley segmentation dataset and the GrabCut dataset, since images in these two datasets do not belong to a same category, and an average F-measure score of several different categories of images cannot always provide a clear evaluation of the performance of these image segmentation algorithms.

Fig. 7 illustrates the minimized energy values obtained by our algorithm by setting λ to different values. The parameter λ in Eq. (1) controls the relative weight of the data term versus the spatial prior term. For a specific image, over large λ always leads to over-segmentation, while too small λ results in under-segmentation. This has been illustrated in Fig. 7. Ideally, λ should be estimated for each image separately. However, it is too hard to do so in practice for an automatical image segmentation system. According to our empirical observation, we found that setting λ within an interval instead of an exact value can produce good segmentation results. This makes it possible to set λ to an exact value that suitable for nearly all test images. Throughout our experiments, we empirically fixed λ to 5.

6.1.2. Performance comparison between our method with and without using auto-context model

We are interested in the relative performance change in segmentation accuracy, i.e., with and without incorporating auto-context model in the energy function. For this purpose, we implemented two versions of our method. Their only difference is

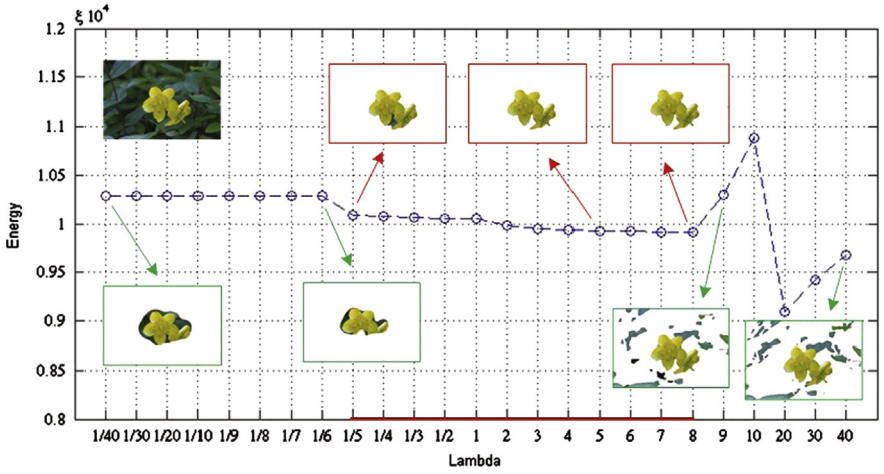

Fig. 7. The minimized energy values obtained by our algorithm by setting λ to different value. Some segmentation results corresponding to the obtained minimized energies are shown.

that one includes the auto-context model in the energy function, and the other one does not include the auto-context model.

We compare these two by using their F-measure scores on Weizmann single object segmentation evaluation dataset [20]. This dataset contains 100 sample images along with ground truth segmentations. The segmentation accuracy is evaluated by assessing its consistency with the ground truth segmentation, and the average F-measure score on the entire database is calculated to serve as the final score.

In Table 4, we summarize the performance of average segmentation accuracy of these two implementations in the first two rows. Clearly, the average segmentation accuracy is improved considerably with the inclusion of auto-context model. Specifically, a 0.03 increase of F-measure score is completely contributed by the auto-context model.

In Fig. 8, we present three representative complex examples, which also subjectively demonstrated the usefulness of the context cue. For example, for the horse image in Fig. 8, the success of cutting out the legs of the horses is obviously due to the auto-context model. Similarly, the context model plays an essential role in completely cutting out of the cap of the framer for the farmer image, and successful removal of line connected to the female body for the female image.

6.1.3. *Performance comparison with 5 state-of-the-art image segmentation algorithms*

We compare our method with 5 state-of-the-art image segmentation algorithms listed in Table 4 by evaluating their F-measure scores on the Weizmann single object segmentation evaluation dataset [20].

Table 4 summarized the F-measure scores of these methods on the database. The F-measure scores of these previous methods are directly quoted from www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/scores.html. Our method's F-measure score is 0.91 ± 0.013, which significantly outperforms all other 5 methods, including the unified approach [44], one interactive segmentation method. This strongly manifested the efficacy of the contextual cue and saliency cue we leveraged.

In addition to the unified approach [44], we also compare our method with another interactive segmentation method [3]. Fig. 9 shows several results of ours and other two interactive segmentation methods. In order to guarantee the fairness of the comparison in terms of initialization, for GrabCut [3], we marked a bounding box exactly containing the object, and used the GrabCut implementation provided in www.cs.cmu.edu/~mohitg/segmentation.htm; for the unified approach [44], we quoted the best results reported by the authors from www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/scores.html. The empirical results show that our automatic method compares favorably with these two interactive segmentation methods. The results are encouraging since no supervision is used in our automatic method.

We only reported quantitative results on the Weizmann single object segmentation dataset, since it contains 100 images along with ground truth segmentations, and it shall be sufficient to manifest the efficacy of our method. Furthermore, among the 4 datasets, only the Weizmann single object segmentation dataset provided a segmentation evaluation code and reported F-measure scores of other five state-of-the-art image segmentation systems, hence these can better guarantee the fairness of comparison. Additionally, we choose these five methods for comparison

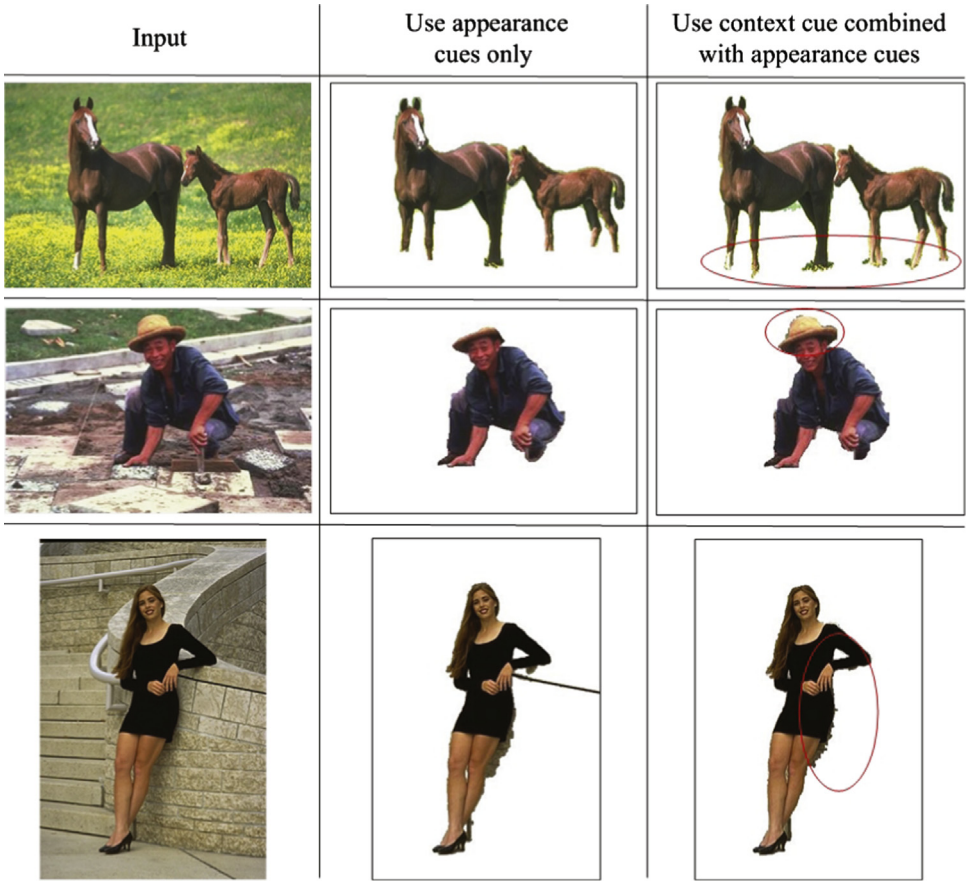

Fig. 8. Performance comparison with and with out auto-context model. Left column: input images. Middle column: the results obtained by our method that uses appearance cue only. Right column: the results obtained by our method that integrates both appearance and contextual cues. The red circle overlayed contains area different from that of the corresponding image in the middle column. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

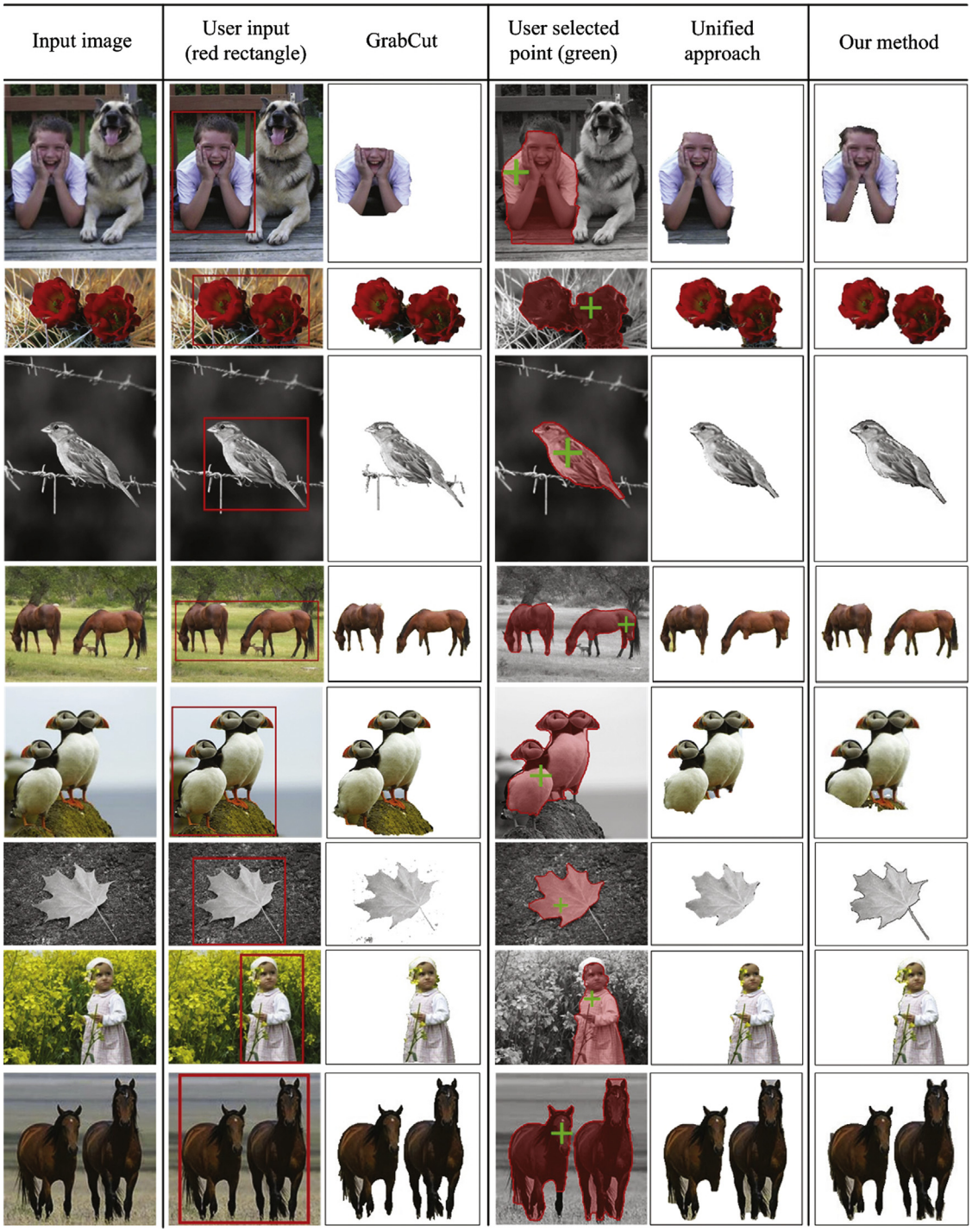

Fig. 9. Comparison results of our method with interactive segmentation methods Grabcut [3], and the unified approach [44].

because they just use information from a single image, and this is a common point with our method.

6.2. Application I: joint segmentation and recognition

Upon convergence of the iterative process of the energy minimization, besides the extracted salient object, we also obtained a fully trained auto-context classifier which can be readily used to recognize the same type of object in new images. Fig. 10 shows some recognition results obtained by applying the auto-context classifier learned from one image to new images. As it is clearly shown, the learned auto-context model generalized well to new images.

To further evaluate the recognition performance of our auto-context model learned from one single image, we tested the performance of our learned classifier on the Weizmann horse database [32], which consists of 328 horse images along with manually annotated label maps. Specifically, we first learn the auto-context model during the segmentation of a single image from the database, which was shown in the top left of Fig. 11. Then the learned auto-context classifier is used to test all other images in the Weizmann horse database. For comparison, we quoted from

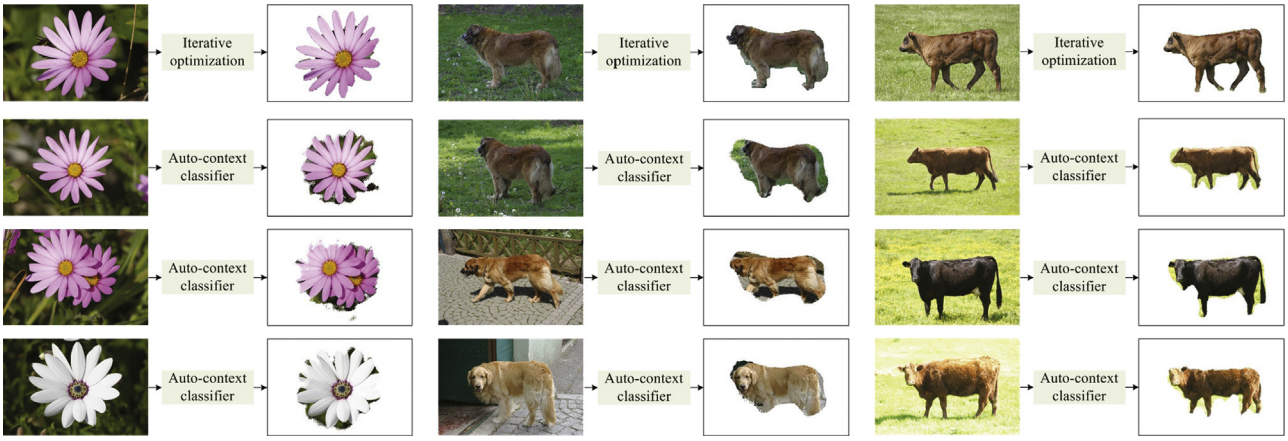

Fig. 10. Recognition results. The 1st row presents input images and the corresponding segmentation results. The 2nd, the 3rd, and the 4th rows list new images including the same type of objects and the corresponding recognition results.

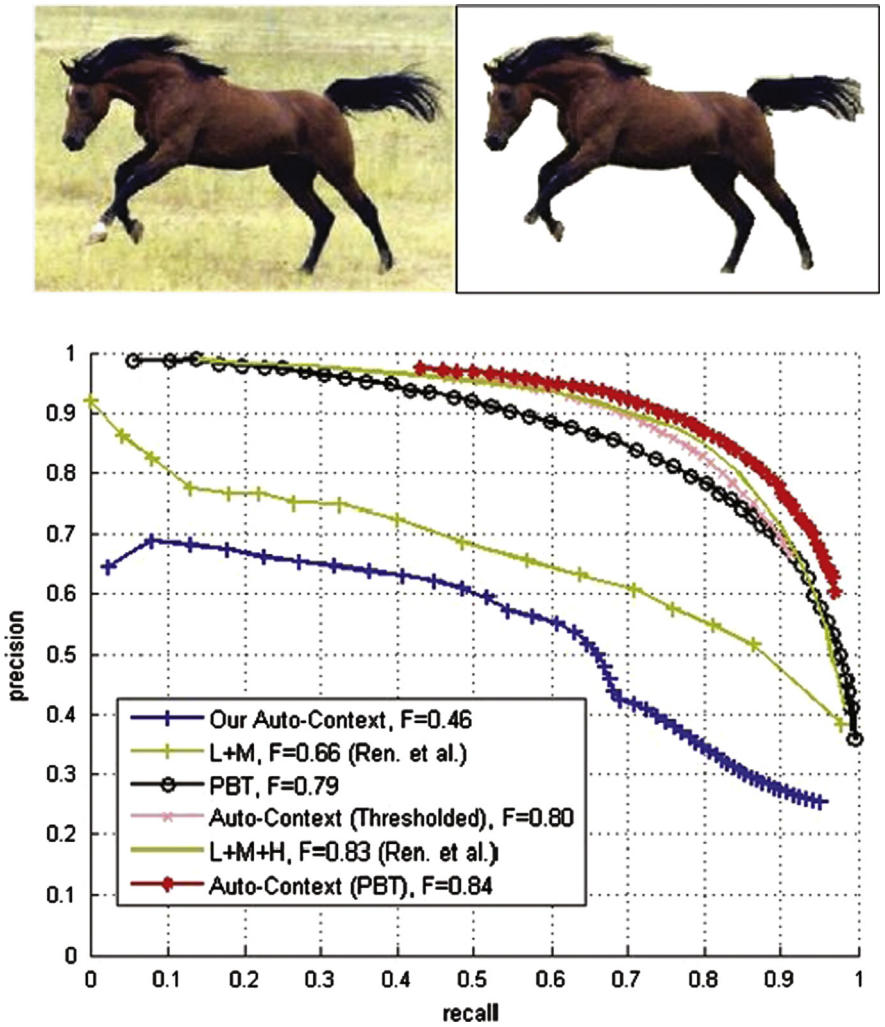

Fig. 11. Precision–recall curves of our method, Tu's method and Ren et al.'s method, obtained by testing on the Weizmann horse database [32].

Tu [14] the figure of the full precision–recall curves for various algorithms including Tu [14] and Ren et al. [48], and added our precision–recall curves. As shown in bottom of Fig. 11, our auto-context model learned from one single image is lower than all other methods. This is a natural result, since all other methods were learned from a large number of images, and needed about one day or several days for training; while ours was learned from one single image and needed several minutes for training.

Furthermore, instead of dividing the training and classification into two steps, we consider our automatic segmentation method with contextual cue as an integral whole, and compute the average F-measure score directly on the entire Weizmann horse database

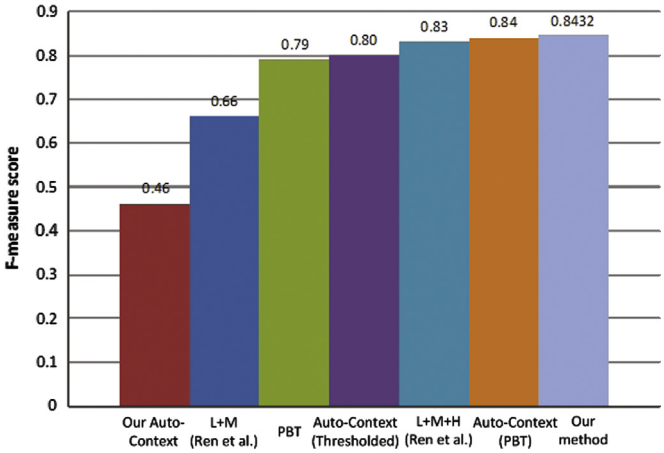

Fig. 12. F-measure score comparison of our method, Tu's methods and Ren et al.'s methods tested on the Weizmann horse database [32].

Table 5
The segmentation accuracy comparison results between our method and the textonboost method [9] by evaluating them on the MSRC dataset. We list segmentation accuracies of 14 categories of the MSRC dataset. The accuracy values in this table are computed as percentage of image pixels assigned to the correct class label, ignoring pixels labeled as void in the ground-truth. The segmentation accuracies of Textonboost are quoted from the experimental results reported in [9].

Image category	Ours	Textonboost
Aeroplane	0.8397	0.596
Bird	0.8646	0.194
Body	0.7966	0.621
Book	0.8485	0.919
Building	0.8609	0.616
Car	0.8038	0.625
Cat	0.8457	0.536
Chair	0.7804	0.154
Cow	0.9121	0.583
Dog	0.8608	0.192
Face	0.8518	0.735
Flower	0.8578	0.628
Sheep	0.8906	0.504
Sign	0.8453	0.351
Tree	0.8491	0.863

[32]. We then compare the average F-measure scores of various methods including Tu [14] and Ren et al. [48] with ours. As shown in Fig. 12, our obtained average F-measure score is nearly the same with Tu's PBT based auto-context method (shown as Auto-Context (PBT)), and it significantly outperforms both Tu's other methods and Ren et al.'s methods.

Through this experiment, we observed that it is difficult to obtain good segmentation results on the entire database when simply relying on the recognition performance of the auto-context model learned from one single image. However, comprehensively benefitting from both the auto-context model learned from one single image and our automatic segmentation, it is easy to achieve good segmentation performance on the entire database while without training on large amount of images.

We also report the comparison results between our method and the textonboost method [9] in Table 5. We choose the textonboost method since it is one representative of methods [9,10,41] which perform jointly segmentation and recognition, and exploit context. Table 5 summaries the segmentation accuracy comparison results. The test set contains 14 categories of images, and these test images are chosen from MSRC dataset, according to the standard that each of these images contains a visual salient object that accounts for a significant portion of the image.

One concern about the fairness of the comparison is that the segmentation accuracies reported in Table 5 for TextonBoost are relative to a multi-class segmentation. The TextonBoost algorithm performs a multi-class semantic labeling, our algorithm, however, performs a figure/ground segmentation. To address this concern, we choose to compare their segmentation performance in terms of the segmentation accuracy of the salient object. The segmentation accuracy is computed as percentage of image pixels assigned to the object, ignoring pixels labeled as void in the ground-truth. The pixels labeled as 'void' are pixels that do not belong to a database class. The results in Table 5, in addition to the *tree* and the *book* categories, clearly show that our algorithm outperforms the textonboost method in term of segmentation accuracy.

Additionally, it should be noted that there is fundamental difference between our method and the other methods using contextual cues [9,10,41]. Specifically, contextual cues of the three methods [9,10,41] are learned from pre-labeled training data (usually multiple images) or derived under strong constraint prior, while our context information is directly derived from the single query image without pre-labeled training data or strong constraint prior. It is not clear what would be a fair comparison between our method with them in term of classification performance.

6.3. Application II: automatic image matting

Although the results shown in Figs. 6, 8–10 are visually compared good, the details of their boundaries are not as good as that of image matting techniques. This motivated us to employ alpha matting techniques to further improve the boundary of the extracted salient object.

Alpha matting aims at softly and accurately extracting the foreground from an image, and user-specified trimap or scribbles which indicate the known foreground/background and the unknown pixels are often required. With the extracted salient object by our method, the trimap or scribbles can be automatically created with a uniform bandwidth (set by the user) through eroding and dilating the binary mask of the extracted object. Once the trimap or scribbles are obtained, any standard matting methods can be adopted to estimate the matte, and a finer boundary of the salient object can then be obtained. Here we use the closed form solution proposed in [18] as the alpha matting system.

Fig. 13 shows several matting results on images from the 4 datasets aforementioned. The results show that our method can be seamlessly fitted into the automatic image matting system [18] as an intelligent frontend.

To further evaluate effectiveness of the trimaps created by our method, we compare final boundaries when the image matting system [18] works with different trimaps or scribbles from our method and from [49]. Fig. 14 presents some comparison results tested on images from [49]. The trimaps from [49] are the finest trimaps among 10 trimaps for each image. The results clearly show that the trimaps created by our method may embrace the trimaps labeled by human labor.

Fig. 15 shows comparison results tested with images from [18] by using the trimaps created by our method and the scribbles from [18]. It can be observed that the matting results by using the trimaps created by our method compare favorably with the results by using the scribbles, as the segmentation is close enough to the ground truth segmentation. We also observed several failure examples of our algorithm in providing trimaps, due to the attention model of our method cannot provide a good initial segmentation in term of three measures described in Section 3.

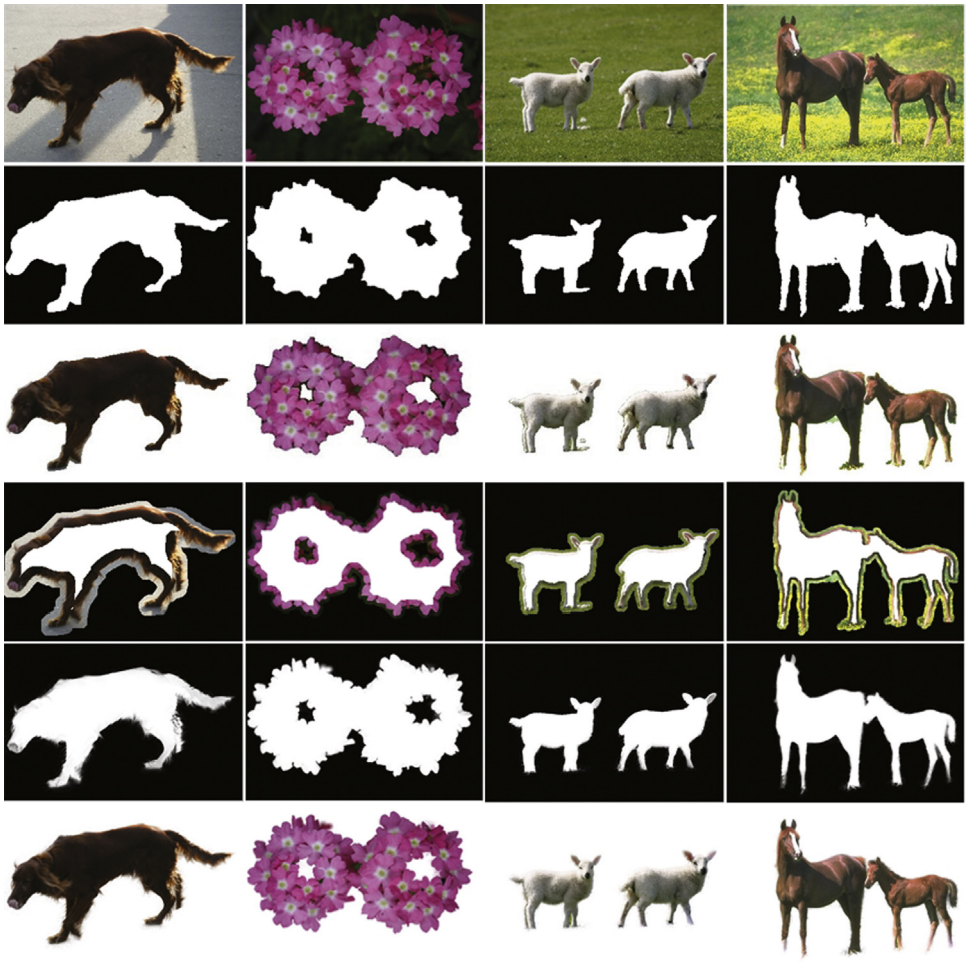

Fig. 13. Matting results. The 1st row presents original images. The 2nd and the 3rd rows are segmentation masks and extracted foreground objects using our method, respectively. The 4th row presents the created trimaps used for matting. The 5th and the 6th are estimated mattes and extracted foreground objects with constant background using close form method [18].

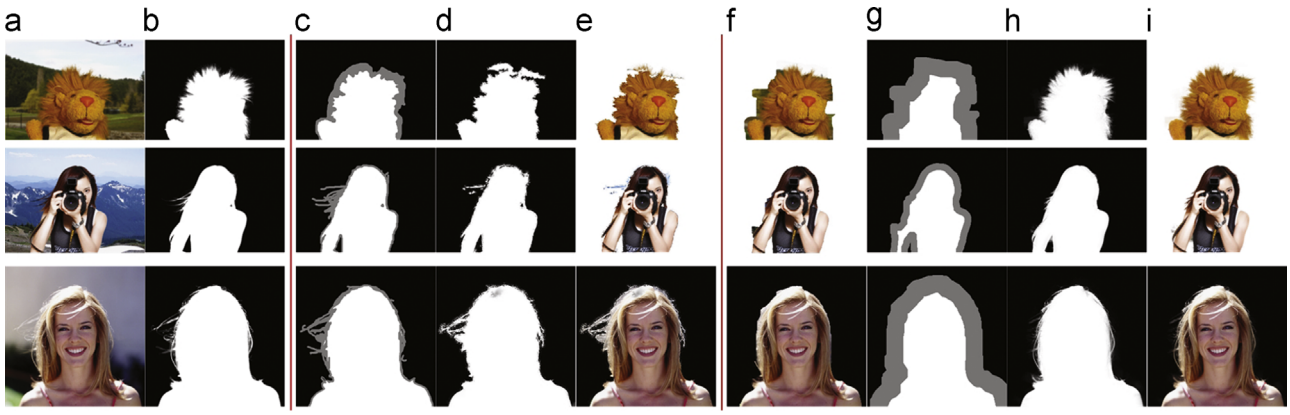

Fig. 14. Comparison with different trimaps. (a) Input image from [49]. (b) Ground truth matte from [49]. (c) Finest trimap among 10 trimaps from [49]. (d) Matting result tested on (c) by the closed form solution [18]. (e) Compositing result with a constant background using (d). (f) Our segmentation result. (g) Trimap created from our segmentation. (h) Matting result tested on (g) by the closed form solution [18]. (i) Compositing result with a constant background using (h).

7. Conclusion

We presented an automatic salient object extraction method, as well as its applications to recognition and alpha matting. Our method is able to automatically extract the object of interest from its background without any user intervention. This is enabled by casting saliency cue, contextual cue and appearance cue into a unified energy minimization framework. Empirical results on four popular segmentation benchmarks demonstrated the superb performance of our method. It compares favorably with even foreground extraction algorithms which leveraged user interaction. We also showed performance of our method in recognition, as

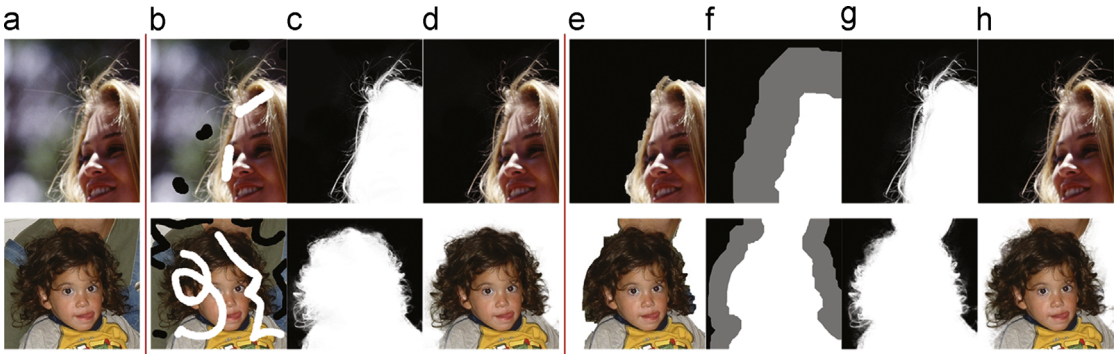

Fig. 15. Comparison results. (a) Input image from [18]. (b) Scribbles from [18]. (c) Matting result tested on (b) by the closed form solution [18]. (d) Compositing result with a constant background using (c). (e) Our segmentation result. (f) Trimap created from our segmentation. (g) Matting result tested on (f) by the closed form solution [18]. (h) Compositing result with a constant background using (g).

well as acting as an intelligent front end for an alpha image matting system. It shall be noted that the accuracy of the initial segmentation obtained from the bottom–up visual saliency model sometimes affects the result of our method, which is the focus of our future research.

Conflict of interest statement

None declared.

Acknowledgement

This work is supported in partial by The National Basic Research Program (973 project) under grant no 2012CB316400, and NSFC projects 90920301,61273252, and 61228303.

References

[1] Y. Li, J. Sun, C. Tang, H. Shum, Lazy snapping, ACM Transactions on Graphics 23 (3) (2004) 303–308.
[2] Y. Boykov, M. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in ND images, in: International Conference on Computer Vision, vol. 1, 2001, pp. 105–112.
[3] C. Rother, V. Kolmogorov, A. Blake, Grabcut: interactive foreground extraction using iterated graph cuts, ACM Transactions on Graphics 23 (3) (2004) 309–314.
[4] Y. Boykov, G. Funka-Lea, Graph cuts and efficient nd image segmentation, International Journal of Computer Vision 70 (2) (2006) 109–131.
[5] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, H. Shum, Learning to detect a salient object, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2) (2011) 353–367.
[6] J. Yedidia, W. Freeman, Y. Weiss, Generalized belief propagation, in: Neural Information Processing Systems, vol. 13, 2000, pp. 689–695.
[7] M.J. Wainwright, T.S. Jaakkola, A.S. Willsky, Tree-reweighted belief propagation algorithms and approximate ml estimation by pseudo-moment matching, in: Workshop on Artificial Intelligence and Statistics, vol. 21, 2003.
[8] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence (2002) 509–522.
[9] J. Shotton, J. Winn, C. Rother, A. Criminisi, Textonboost: joint appearance, shape and context modeling for multi-class object recognition and segmentation, The European Conference on Computer Vision (2006) 1–15.
[10] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, S. Belongie, Objects in context, in: The 11th International Conference on Computer Vision, IEEE Computer Society, 2007, pp. 1–8.
[11] A. Oliva, A. Torralba, The role of context in object recognition, Trends in Cognitive Sciences 11 (12) (2007) 520–527.
[12] L. Wolf, S. Bileschi, A critical view of context, International Journal of Computer Vision 69 (2) (2006) 251–261.
[13] K. Murphy, A. Torralba, W. Freeman, Using the forest to see the trees: a graphical model relating features, objects and scenes, Advances in Neural Information Processing Systems 16 (2003).
[14] Z. Tu, Auto-context and its application to high-level vision tasks, in: International Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[15] G. Hua, Z. Liu, Z. Zhang, Y. Wu, Iterative local-global energy minimization for automatic extraction of objects of interest, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) (2006) 1701–1706.
[16] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, Advances in Neural Information Processing Systems 19 (2007) 545.
[17] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9) (2004) 1124–1137.
[18] A. Levin, D. Lischinski, Y. Weiss, A closed-form solution to natural image matting, IEEE Transactions on Pattern Analysis and Machine Intelligence (2007) 228–242.
[19] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (5) (2004) 530–549.
[20] S. Alpert, M. Galun, R. Basri, A. Brandt, Image segmentation by probabilistic bottom–up aggregation and cue integration, in: International Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[21] L. Wang, J. Xue, N. Zheng, G. Hua, Automatic salient object extraction with contextual cue, in: The 13th International Conference on Computer Vision, 2011, pp. 1–8.
[22] O. Veksler, Star shape prior for graph-cut image segmentation, in: The European Conference on Computer Vision, 2008, pp. 454–467.
[23] X. He, R. Zemel, D. Ray, Learning and incorporating top–down cues in image segmentation, in: The 9th European Conference on Computer Vision, 2006, pp. 338–351.
[24] A. Blake, C. Rother, M. Brown, P. Perez, P. Torr, Interactive image segmentation using an adaptive GMMRF model, The European Conference on Computer Vision (2004) 428–441.
[25] D. Freedman, T. Zhang, Interactive graph cut based segmentation with shape priors, in: The International Conference on Computer Vision and Pattern Recognition, 2005.
[26] D. Walther, C. Koch, Modeling attention to salient proto-objects, Neural Networks 19 (9) (2006) 1395–1407.
[27] A. Rosenfeld, R. Hummel, S. Zucker, Scene labeling by relaxation operations, IEEE Transactions on Systems, Man and Cybernetics 6 (1976) 420–433.
[28] R. Hummel, S. Zucker, On the foundations of relaxation labeling processes, IEEE Transactions on Pattern Analysis and Machine Intelligence 3 (1983) 267–287.
[29] M. Pelillo, The dynamics of nonlinear relaxation labeling processes, Journal of Mathematical Imaging and Vision 7 (4) (1997) 309–323.
[30] W. Zhou, Q. Tian, Y. Lu, L. Yang, H. Li, Latent visual context learning for web image applications, Pattern Recognition 44 (10–11) (2011) 2263–2273.
[31] Z. Liang, Z. Chi, H. Fu, D. Feng, Salient object detection using content-sensitive hypergraph representation and partitioning, Pattern Recognition 45 (11) (2012) 3886–3901.
[32] E. Borenstein, E. Sharon, S. Ullman, Combining top–down and bottom–up segmentation, in: International Conference on Computer Vision and Pattern Recognition, 2004.
[33] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, International Conference on Computer Vision 2 (2003) 264–271.
[34] S. Kumar, M. Hebert, Discriminative random fields: a discriminative framework for contextual interaction in classification, in: International Conference on Computer Vision, 2003, pp. 1150–1159.
[35] A. Torralba, K. Murphy, W. Freeman, Contextual Models for Object Detection Using Boosted Random Fields, IEEE Computer Society, 2004.
[36] A. Levin, Y. Weiss, Learning to combine bottom–up and top–down segmentation, in: The European Conference on Computer Vision, 2006, pp. 581–594.
[37] M. Kumar, P. Torr, A. Zisserman, OBJ CUT, in: International Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 18–25.
[38] Y. Chen, L. Zhu, C. Lin, A. Yuille, H. Zhang, Rapid inference on a novel and/or graph for object detection, segmentation and parsing, Advances in Neural Information Processing Systems (2008) 289–296.

[39] A. Montillo, J. Shotton, J. Winn, J. Iglesias, D. Metaxas, A. Criminisi, Entangled decision forests and their application for semantic segmentation of ct images, in: Information Processing in Medical Imaging, Springer, 2011, pp. 184–196.
[40] X. He, R. Zemel, M. Carreira-Perpinán, Multiscale conditional random fields for image labeling, in: The International Conference on Computer Vision and Pattern Recognition, vol. 2, 2004, pp. 695–702.
[41] J. Shotton, M. Johnson, R. Cipolla, Semantic texton forests for image categorization and segmentation, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2008 (CVPR 2008), IEEE, 2008, pp. 1–8.
[42] S. Wu, F. Crestani, Data fusion with estimated weights, in: International Conference on Information and Knowledge Management, 2002, pp. 648–651.
[43] V. Kolmogorov, R. Zabin, What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2) (2004) 147–159.
[44] S. Bagon, O. Boiman, M. Irani, What is a good image segment? A unified approach to segment extraction, in: The European Conference on Computer Vision, 2008, pp. 30–44.
[45] M. Galun, E. Sharon, R. Basri, A. Brandt, Texture segmentation by multiscale aggregation of filter responses and shape elements, in: International Journal of Computer Vision, 2008, pp. 716–723.
[46] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2002) 888–905.
[47] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.
[48] X. Ren, C. Fowlkes, J. Malik, Cue integration for figure/ground labeling, Advances in Neural Information Processing Systems 18 (2005) 1121–1128.
[49] J. Wang, M. Cohen, Optimized color sampling for robust matting, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.

Jianru Xue got his Master and Ph.d. degree from Xi'an Jiaotong University in 1999 and 2003, respectively. He had worked in FujiXerox from 2002 to 2003, visited University of California, Los Angeles from 2008 to 2009. His research field includes computer vision, visual navigation, and video coding based on analysis. He is currently a professor of Institute of Artificial Intelligence and Robotics of Xi'an Jiaotong University. He served as co-organization chair Asian conference on computer vision 2009 and Virtual System and Multimedia 2006. He also served as PC member of Pattern recognition 2012, Asian conference on Computer vision 2010 and 2012.

Le Wang got his B.S. and M.S. from Xi'an Jiaotong University in 2008 and 2010, respectively. He is now pursuing his Ph.D. degree in the Institute of Artificial Intelligence and Robotics of Xi'an Jiaotong University. His research field includes image segmentation, web-image processing and analysis.

Nanning Zheng graduated from the Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an, China, in 1975, and received the M.S. degree in information and control engineering from Xi'an Jiaotong University in 1981 and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1985. He joined Xi'an Jiaotong University in 1975, and is currently a Professor and the Director of the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His current research interests include computer vision, pattern recognition, machine vision and image processing, neural networks, and hardware implementation of intelligent systems. Dr. Zheng became a member of the Chinese Academy of Engineering in 1999, and has been the Chief Scientist and the Director of the Information Technology Committee of the China National High Technology Research and Development Program since 2001. He was the General Chair of the International Symposium on Information Theory and Its Applications and the General Co-Chair of the International Symposium on Nonlinear Theory and Its Applications, both in 2002. He is a member of the Board of Governors of the IEEE ITS Society and the Chinese Representative on the Governing Board of the International Association for Pattern Recognition. He also serves as an Executive Deputy Editor of the Chinese Science Bulletin.

Gang Hua received the B.S. degree in automatic control engineering and the M.S. degree in control science and engineering from Xi'an Jiaotong University (XJTU), Xi'an, China, in 1999 and 2002, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering at Northwestern University, Evanston, IL, in 2006. He was enrolled in the Special Class for the Gifted Young of XJTU in 1994. He is currently an Associate Professor of Computer Science at Stevens Institute of Technology, Hoboken, NJ. Before that, he was a research staff member at the IBM Research T. J.Watson Center, Hawthorne, NY, from 2010 to 2011, a senior researcher at Nokia Research Center, Hollywood, CA, from 2009 to 2010, and a scientist at Microsoft Live Labs Research, Redmond, WA, from 2006 to 2009. He is the author of more than 50 peer reviewed publications in prestigious international journals and conferences. As of September 2011, he holds 3 U.S. patents and has 17 more patents pending. Dr. Hua is an associate editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and IAPR Journal of Machine Vision and Applications, and a guest editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the International Journal on Computer Vision. He is an area chair of the IEEE International Conference on Computer Vision, 2011, an area chair of ACM Multimedia 2011, and a Workshops and Proceedings Chair of the IEEE Conference on Face and Gesture Recognition 2011. He is a member of the ACM.