Learning to Extract Focused Objects From Low DOF Images

Hongliang Li, *Member, IEEE,* and King N. Ngan, *Fellow, IEEE*

***Abstract*—This paper proposes an approach to extract focused objects (i.e., attention objects) from low depth-of-field images. To recognize the focused object, we decompose the image into multiple regions, which are described by using three types of visual descriptors. Each descriptor is extracted from a representation of some aspects of local appearance, e.g., a spatially localized texture, color, or geometrical property. Therefore, the focus detection of a region can be achieved by the classification of extracted visual descriptors based on a binary classifier. We employ a boosting algorithm to learn the classifier with a cascade of decision structure. Given a test image, initial segmentation can be achieved using obtained classification results. Finally, we apply a post-processing technique to improve the results by incorporating region grouping and pixel-level segmentation. Experimental evaluation on a number of images demonstrates the performance advantages of the proposed method, when compared with state-of-the-art methods.**

***Index Terms*—Attention, boosting, image segmentation, low depth-of-field, object segmentation, visual descriptor.**

I. Introduction

SEMANTIC object segmentation is one of the most important and challenging problems in computer vision and multimedia applications, which aims to assign an object mask to each pixel of a given image [1], [2]. It can be seen as a combination of the object detection and localization tasks. The first task is to find the possible object locations in a given image using a pattern classification method. It can be skipped by the user's definition of the object location in a supervised manner. The second task aims to extract the object by grouping together similar pixels, which provides the closed boundary or the mask of the semantic object [3].

In this paper, we are especially interested in performing segmentation on focused objects, which usually correspond to the visual attention objects in many photos, TV programs, or film productions. The recognition of focused object provides an important and powerful cue for visual information processing, including content-based coding, retrieval, browsing, and surveillance. In optics, the range of distance in front of and behind the object which appears to be in focus is called the depth-of-field (DOF). The lower DOF usually produces a visual effect like object-in-focus, which means that only the object of interest is in sharp focus, whereas background objects are typically blurred, being out-of-focus [4]. In order to highlight the attention object, such as an important person, it is usually focused on the image plane by the lens. An example of focused object is illustrated in Fig. 1, which includes a white flower to be extracted. Fig. 1(b) shows the ground truth mask of the original image in Fig. 1(a). The goal of our work is to extract the focused flower from the original low-DOF image, which is described in Fig. 1(c).

The segmentation of the focused object can be traced back to the work of Tsai [5], which relies on the measurement of defocus for object edges using the Sobel edge operator. For every edge point of interest in the gradient image, the amount of defocus at a pixel is measured by the proportion of the edge region in a small neighborhood window using the moment preserving method. Since distinct boundary edges are required for the following edge-linking and region-filling processes, this method usually fails to deal with those disconnected boundaries.

Some works utilize more robust statistical features, e.g., the local variance [6] and the fourth order moment [7], to identify low-DOF regions. In order to describe high-frequency components in an image, a local variance image field is first calculated, and a thresholding method is then applied for the segmentation [6]. Considering the limitations of details description, the method in [7] turns to calculate the higher order statistics (i.e., fourth-order moment) for all pixels in the low-DOF image. Then, a typical region-based segmentation technique is employed, which includes *simplification*, *seed growing*, and *region merging*. In that work, morphological filters are employed to generate the simplification. The regions with the highest value of fourth-order moment are then selected as seed regions. Finally, the region merging approach based on the maximum *a posteriori* is performed to extract the object. For these methods, the corresponding classification or region-merging approaches are based on the initial estimation results. Therefore, a good defocused background, which means the defocused regions should have high blurring degree, is required for the existing methods [6], [7]. Otherwise, over estimation of the focused objects may be introduced.

Manuscript received December 3, 2009; revised April 25, 2010 and July 22, 2010; accepted October 14, 2010. Date of publication March 17, 2011; date of current version November 2, 2011. This work was supported in part by the NSFC, under Grant 60972109 and in part by the Program for New Century Excellent Talents in University, under Grant NCET-08-0090. This paper was recommended by Associate Editor E. Izquierdo.

H. Li is with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: hlli@ee.uestc.edu.cn).

K. N. Ngan is with the Department of Electronic Engineering, Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: knngan@ee.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2129150

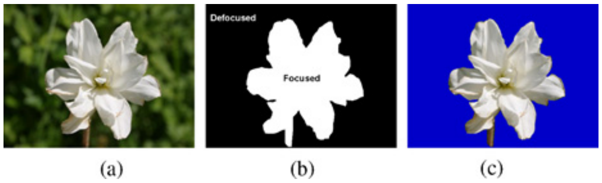


Fig. 1. Example of focused object extraction. (a) Original image. (b) Ground truth mask. (c) Focused object of interest.

Unlike the spatial approaches, some multiscale approaches based on high frequency wavelet coefficients are presented in [4] and [8], which detect the sharply focused objects in terms of the statistics of these coefficients. These methods are motivated by the observation that focused object regions have more high-value wavelet coefficients in the high frequency bands of the transform. As we know, the important statistical properties of wavelet coefficients are spatial-frequency localization and energy compaction. This decomposition process on an image will effectively compact the energy into few wavelet coefficients. Therefore, many useful details of the focused objects are often discarded together with the defocused background. It is difficult to get good detection accuracy for the object of interest by using few wavelet coefficients.

Recently, an unsupervised segmentation algorithm is presented to model the original low-DOF image as a matting problem [3]. This method consists of three stages. A reblurred version of the input image is first generated by a point-spread function. To reduce the noise effect and perform the clustering, a bilateral filter and morphological operator are then employed to smooth and merge the focused regions. The final segmentation and boundary refinement are achieved by using an adaptive error control matting approach. In addition, an automatic focus area estimation method for a single image is proposed in [9], which produces relative focus maps by localized blind deconvolution and a new residual error-based classification.

Most existing methods perform focus detection based only on the responses of high-frequency contents. Because of the failure to take advantage of all the spatial frequency components of the image, some methods generate higher focus values at object edges instead of producing uniform maps that cover the whole object. To identify focused regions, many existing methods extract features from a low-DOF image in terms of intensity changes, while neglecting other useful information, such as color or geometrical feature. In addition, hard thresholds obtained by empirical observation are usually employed to perform focus decision, which may reduce the robustness of the algorithm to noise.

In this paper, we propose a method to segment focused objects from low-DOF images. Unlike existing methods that perform focus decision based on the measurement of the amount of high-frequency components from gray level image, our method learns to identify focused objects by using a set of color training images. Fig. 2 shows the framework of our proposed method, which consists of three parts, namely training, testing, and post-processing. For the training stage, each training image is partitioned into focused objects and background regions manually or in semi-supervised manner (e.g., lazy snapping or GrabCut). To achieve the goal of focused object identification, an important step is to decompose the image into multiple regions that are described by three types of visual descriptors according to the texture, color, and geometrical properties. The focus detection of a region can be achieved by the classification of extracted visual descriptors based on a binary classifier, which allows us to avoid hard thresholding during focus objects detection for the existing methods. Finally, a post-processing scheme is applied to improve the segmentation result, which includes region-level and pixel-level corrections. Compared with the classic and well-established segmentation models given in [10] that take into account motion or disparity fields as additional information, this paper concentrates on the specific object segmentation, i.e., focused objects, by using the learning model.

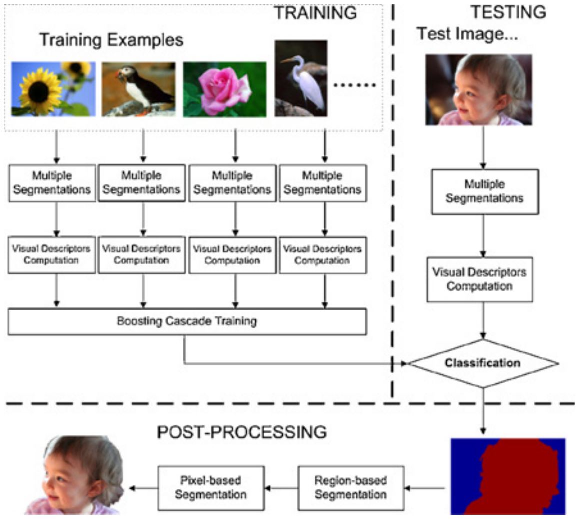


Fig. 2. Framework of our proposed method.

This paper is organized as follows. Section II introduces our proposed segmentation algorithm. Experimental results are provided in Section III to demonstrate the effectiveness of our low-DOF segmentation algorithm. Finally, Section IV concludes this paper.

II. Proposed Method

A. Ambiguity Analysis in the Description of Focus Object

Once some visual primitives such as high frequency points or regions are obtained to highlight the local image change, the focus object can be extracted by evaluating the saliency properties of those visual features. However, such an attention description tends to be ambiguous because continuous visual data generally exhibit much larger variabilities and uncertainties [11]. The same visual features are likely to belong to different visual objects due to under-representations. As illustrated in Fig. 1, we can select some small blocks from the focused object (e.g., a 16 × 16 block at the top-left leaves for the white flower) and the defocused background, respectively. If we remove the mean intensity value from each block by adjusting the mean value of each block to zero, two blocks tend to have the same class label due to the similar appearance feature. However, they belong to different objects. To avoid such ambiguity, one possible solution may be to put them into a spatial context and integrate other attention features. For example, we can use the spatial dependency and discover more co-occurrence regions (e.g., whole white flower) to reduce the uncertainty. In addition, more features such as the color feature can be considered to improve the detection performance. In this paper, we present a new solution for extracting attention objects from low-DOF images by incorporating spatial coherency and attention features.

B. Generating Multiple Segmentations for Input Image

The problem of focus polysemy becomes apparent when we consider how to evaluate the blurring degree of an image using the local contrast, such as pixel-based or patch-based method. Saliency measurement is performed on each pixel/patch, which can ignore the spatial and neighborhood relationships. For example, in order to cluster saliency patches into focused object, morphological operations may be explored to reclassify those misclassified patches [3], [7]. However, it is still unlucky for those focused patches that exceed the size of structure element to avoid false detection. Thus, what we need is to find a way to merge pixels/patches spatially and make them more descriptive.

As a first step of our algorithm described in Fig. 2, an initial over-segmentation of an image is used by partitioning an image into multiple regions. This idea sounds simple in theory, where we consider utilizing a segmentation tool to assign each segment to a coherent object. However, it is still an unsolved problem for image segmentation to provide constituent objects based on current approaches [12]. The idea of creating multiple segmentations by varying the parameters of segmenting algorithms to discover the good ones was discussed in [12] and [13]. Here, we choose to use a segmentation method [14], which computes image segmentation based on pairwise region comparison. The algorithm is highly efficient, which runs in $O(n \log n)$ time for n image pixels. Typical parameters in this method include sigma used to smooth the input image and the threshold θ. An example of the segmentation result by [14] is illustrated in Fig. 3. The original and the ground truth mask are shown in Fig. 3(a) and (b), respectively. The multiple segmentation result is presented in Fig. 3(c) by using default parameters. It can be seen that the original image has been partitioned into a lot of sub-regions. Different segmentation results can be obtained by varying the parameters for the input image. It is worth mentioning that there is no specific requirement for the over-segmentation algorithm. Other segmentation algorithms such as Normalized Cuts [15] can also be applied.

C. Obtaining Region Representation

In our work, we develop three types of visual descriptors to describe an image region, i.e., intensity visual descriptor, color visual descriptor, and geometrical visual descriptor. Each visual descriptor aims to discover the region property from that special aspect. It means that each segmented region results in one descriptor with respect to intensity, color, and geometrical feature. Once the visual descriptors are computed from an image, the segmentation of focused object is converted to the classification of visual descriptors.

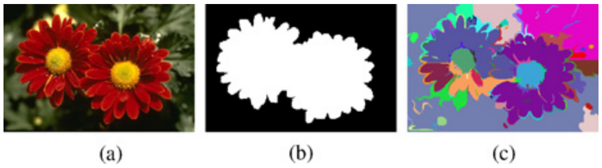


Fig. 3. Example of multiple segmentations. (a) Original image. (b) Ground truth mask. (c) Multiple segmentations by [14] with default parameters.

1) *Intensity Visual Descriptor (IVD):* After the over-segmentation step, we can describe the appearance of a region using an intensity or texture feature. Some approaches represent an image using affine covariant regions described by scale invariant feature transform descriptors [12], or compute the average of texture features over all the pixels within this region [16]. These approaches are dependent on the calculation of interest points. Instead of computing the interest points, we construct the region descriptors (i.e., a set of filter responses) using the convolution of a filter bank with the given image. Note that this visual descriptor aims to describe intensity change in the region. This descriptor is a combination of 12 focus saliency functions (FSF) [3], 8 rotationally symmetric Laplacian of Gaussian filters (LoG), and 4 2-D Laplacian operators. The last two types of filters are usually employed to generate the local descriptors, and were shown to have good performance for object categorization [17] and scene understanding [18]. The four Laplacian operators are applied to intensity channel, thus producing four filter responses. The eight LoGs are applied to the same channel, thus generating eight filter responses. The 12 FSFs are also employed to the intensity channel, thus producing 12 filter responses. The parameter values for these filters are shown in Table I. Therefore, each pixel is associated with a 24-D feature vector (or intensity visual descriptor) when all filter responses are concatenated into a single 24-D vector after absolute operation. A region descriptor can be obtained by computing the mean value of feature vectors of all pixels in this region. Once local appearance descriptors are computed for a region, each region can be represented by an IVD according to the filtering outputs.

2) *Color Visual Descriptor (CVD):* Where intensity visual descriptors are used to describe the region appearance in terms of texture features, color visual descriptors are designed to describe the region appearance from the aspect of color variations. Fig. 1 has illustrated the effect of color component on the focus object detection. In this paper, we choose the RGB color space to extract color features. Of course, other color spaces such as CIE, L*a*b*, or HSI can also be used. To compute the color visual descriptor of a region, we consider using two types of color features. The first is the second statistical moment, i.e., color covariance, which is used to measure the strength of the correlation between R, G, and B color channels. It is known that if a region becomes out of focus, the blurring progress can be modeled as the convolution of a point spread function over this region. Only low-frequency energy is preserved after the blurring operation, which means

TABLE I
PARAMETER VALUES FOR THE FILTER BANK OF THE IVD

Filter Type	Number	Parameters
Laplacian Operator	4	Control parameter $\alpha = 0, 0.3, 0.6, 0.9$ for 3×3 filter size
LoG	8	$\sigma =$0.5, 1.0, 1.5 and 2.0 for both 3×3 and 5×5 filter sizes
FSF	12	$\sigma =$0.4, 0.9, 1.4, 1.9 for 3×3 filter size, $\sigma =$0.5, 1.0, 1.5, 2.0 for 5×5 filter size, and $\sigma =$0.8, 1.3, 1.8, 2.3 for 7×7 filter size

that most of the pixels will share similar values. Thus, the defocused region is more likely to present small covariances than the original (i.e., focused) region. This property can provide useful information to distinguish them. Since the 3×3 covariance matrix is symmetric, we have to select only six covariance values for the CVD construction.

The second color feature is the color contrast feature, which is built from a center-surround operation. It is usually employed to extract the early attention feature by using the center-surround difference between fine and coarse scales [19]. In this paper, the average color of a region is first computed for each channel. We then divide them by the mean value of the whole image to compute the contrast ratio with respect to each color channel, which shows better performance than local color contrast from dyadic Gaussian pyramids. Finally, a 9-dimension (9-D) color visual descriptor can be generated by combining 6-D color covariance and 3-D color contrast features.

It is noticed that two features, namely the covariance and contrast, are employed to generate the CVD in our work. The motivation of choosing the covariance for the CVD construction arises out of a concern about the different responses to focused and defocused regions, which enables us to distinguish them by computing the covariance values. The motivation for introducing the contrast feature is based on the concern about the fact that focused objects usually represent the attention objects. Contrast response, as an important cue, has been successfully applied to extract attention objects from a given image. In addition, other color descriptors can also be designed and used for the CVD construction.

3) *Geometrical Visual Descriptor (GVD):* Unlike intensity and color visual descriptors that are used to describe regions from the appearance nature, the goal of geometrical visual descriptor is to extract the geometrical information from those regions. In this paper, five regional features are employed to construct this visual descriptor. Each feature measures a certain property of this region, which is described as follows.

1) *Region position*, which is computed from the offset of the center of mass for the region to the image center. The offset is defined as the Euclidean distance between two centers coordinates. It is based on a reasonable assumption that focused objects tend to be close to the image center. The position feature has been successfully used in attention object detection [20] and face segmentation [21].
2) *Region area*, which aims to describe the region size by computing the total number of pixels in the region.
3) *Region Euler-number*, which is defined as the difference between the number of objects in the region and the number of holes in those objects based on 8-connectivity measurement. More details on Euler-number can be referred to [29]. The motivation of using this feature is based on the fact that focused objects are usually surrounded by the defocused regions, which may generate the holes for the defocused background.

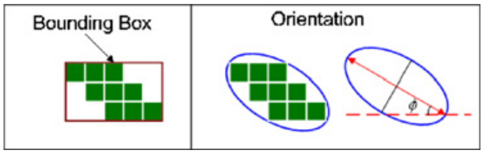


Fig. 4. Illustration of region features “bounding box” and “orientation.”

4) *Region extent*, which is defined as the proportion of the pixels in the region with respect to the bounding box. It can be obtained by dividing the region area by the area of the bounding box. Note that the term “bounding box” represents the smallest rectangle containing the region. An example is shown in the left part of Fig. 4, where the rectangle corresponds to the bounding box of the nine-pixel region.
5) *Orientation*, which denotes the angle between the horizontal axis and the major axis of the ellipse that has the same second moments as the region. The right part of Fig. 4 shows an image region and its corresponding orientation ϕ between the dotted line and the major axis.

We calculate the output of each feature from a region, and combine them into one vector. Thus, we can construct a 5-D geometrical visual descriptor using above regional features. Of course, more region description such as shape number and region contour can be combined into GVD construction.

In summary, we construct a 24-D IVD, 9-D CVD, and 5-D GVD in our work, which results in the total dimensionality of the descriptor to 38 for each region. We consider the IVD, CVD, and GVD separately in the training process.

D. Training Visual Descriptors Classifier

In this section, we discuss how to train a classifier to assign each visual descriptor to a class label, i.e., focused object or background. Many supervised learning methods can be employed to train a classifier, such as support vector machines, neural network, and k-nearest neighbors algorithms. In this paper, we aim to use a boosting algorithm to perform classifier training. To make the representation clearer, we briefly review the boosting approaches.

Recently, many boosting approaches have been presented to detect the object of interest, such as AdaBoost [22] and FloatBoost [23]. The AdaBoost learning algorithm can be

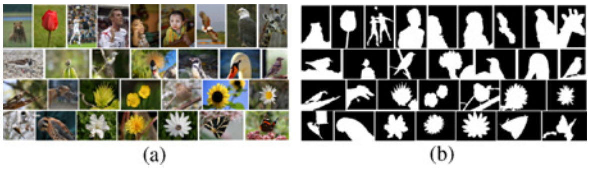


Fig. 5. Training examples of focused object. (a) Original images. (b) Reference masks.

Algorithm 1 The Adaboost Classifier Training for Visual Descriptors

```
1. For all training images, compute training descriptors $(\mathbf{w}_1, l_1)$, ..., $(\mathbf{w}_n, l_n)$ where class labels $l_i = 0, 1$ for background descriptors and focus object descriptors, respectively.
2. Initialize sample weights $w_{1,i} = \frac{1}{2p}$ and $\frac{1}{2q}$ for $l_i = 0, 1$, respectively, where $p$ and $q$ denote the number of focus and background descriptors, respectively.
3. Repeat for $t$=1,..., $T$:
    (a) Normalize visual descriptor weights $w_{t,i}$.
    (b) Assume $N_h$ is the number of weak classifiers, calculate the classification error for all weak classifier $h_k, k = 1, 2, ..., N_h$, namely
        $\varepsilon_t(k) = \sum_i w_{t,i}|h_k(\mathbf{w}_i) - l_i|, \quad \text{for} \quad k = 1, 2, ..., N_h.$
    (c) Choose the best weak classifier $h_t$ with the lowest error $\varepsilon_t$.
    (d) Renew weight $w_{t,i} \leftarrow w_{t,i}\left(\frac{\epsilon_t}{1-\epsilon_t}\right)_i^v$, $v_i = 1 - |h_t - l_i|$.
5. Output the combined classifiers: Focus label if $\sum_{t=1}^T \gamma_t h_t - \frac{1}{2}\sum_{t=1}^T \gamma_t \geq 0$ with $\gamma_t = \log(\frac{1-\epsilon_t}{\epsilon_t})$, Background label otherwise.
```

interpreted as a greedy feature selection process, which selects a small set of classifiers with the lowest errors and their associated weights. Although each weak classifier cannot provide good classification for training images, a weighted combination of weak classifiers called strong classifier can improve the performance of the final classification significantly.

In our work, we use the Adaboost learning algorithm to yield a strong classifier for training examples, which is employed for the attention object detection. In order to obtain training images, we first manually segment each image into focused object and defocused background, which are served as a reference mask. Some segmented masks are illustrated in Fig. 5. We then implement the initial over-segmentation using the method [14], which partitions each image into a lot of regions. We compute visual descriptor for each region and assign each visual descriptor with a label based on the overlap between the region and the mask. It is difficult to expect a segmentation algorithm to partition an image into its constituent objects [12]. Therefore, if over 80% of the region is covered by the focused object, the label of the corresponding visual descriptor will be set to the value one, representing an object. On the contrary, zero will be assigned to the descriptor if less than 20% of focus areas are observed, representing background.

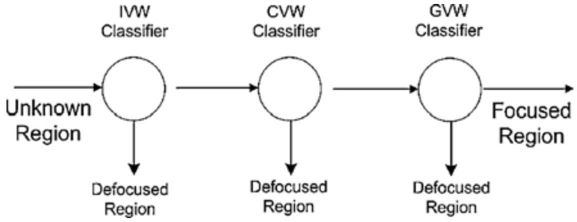


Fig. 6. Illustration of the focus detection cascade.

The detailed steps of this algorithm are described in Algorithm (1), where a small number of robust features are selected to yield a superior classifier. During the learning stage, each best weak classifier will be obtained by the greedy search from the total feature set of visual descriptors extracted from the training images. In our work, a weak classifier h_k is implemented as a decision stump that basically thresholds the distance between the received vector and the kth vector. Each weak classifier consists of a feature $\psi_k(\mathbf{w})$, a threshold τ, and a direction indicator λ, which is defined as follows:

$$h_k(\mathbf{w}) = \begin{cases} 1, & \text{if} \quad \lambda\psi_k(\mathbf{w}) < \lambda\tau \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

with

$$\psi_k(\mathbf{w}) = \psi(\mathbf{w}_k, \mathbf{w}) = \|\mathbf{w}_k - \mathbf{w}\| \tag{2}$$

where $\mathbf{w}$ denotes a descriptor, and k represents the descriptor index. From (1) and (2), we can obtain N features from training descriptors, which can be used as weak classifiers. The decision stumps perform the binary classification using the distance between feature vectors.

E. Cascade of Classifiers

We use a cascade of classifiers to reject as many defocused descriptors as possible at the earliest stage. Fig. 6 shows the constructed detection cascade. The first stage is the intensity classifier that is trained using intensity visual descriptors. The color strong classifier is employed as the second stage based on color visual descriptors. The third stage is constructed by using the geometrical visual descriptors. In our work, we use a simple framework to train the cascade classifier. Each layer of the cascade is trained using the AdaBoost learning algorithm (as described in Algorithm 1) with the given number of features. Each strong classifier is then adjusted to have a high detection rate (e.g., 99%), but a moderate false positive rate (50%) after the AdaBoost learning.

After cascade boosting training, we get the final strong classifier. Given a test image, we first perform the multiple segmentations and region descriptions. For each region, three types of visual descriptors are computed according to different image features. The total descriptor is classified by the decision cascade, resulting in the object/background label. A region can be identified as the focused area when its visual descriptor is classified into the positive class.

F. Post-Processing Method

After boosting classification, we can rapidly classify unlabeled regions and access focused regions from the input image.

However, the spatial correlation between neighboring regions is usually ignored when a classifier only works on each region independently. In order to correct possible false detections or missed detections, we apply a post-processing technique based on the region grouping and pixel-level refinement, to improve the object segmentation results.

1) *Region Grouping:* The basic idea of our region grouping algorithm is to merge similar regions using color information. The overall process of this algorithm is summarized in Algorithm 2. Starting from results given by the boosting classification, if a region is classified into the focused object, we have region label one, zero otherwise. The algorithm repeats region grouping decision for all input regions, and stops if no region labels change. The outputs of this algorithm are updated region labels for segmentation.

As shown in Algorithm 2, for an input region k, we do not change its label if the pixel number P_k in this region is larger than $P_{\max} = 0.5 \cdot N_i$. Otherwise, we search the neighboring regions (i.e., $\mathcal{N}_k^f$ and $\mathcal{N}_k^b$) of region k using the morphological dilation operation with a 21×21 square structuring element. Here N_i represents the total number of pixels in an image, $\mathcal{N}_k^f$ and $\mathcal{N}_k^b$ denote the regions belonging to focus object and background, respectively. In order to perform the region grouping, the minimal distances $d_{\min}^b$ and $d_{\min}^f$ between region k and neighboring focus/background regions should be computed respectively, which are defined as follows:

$$d_{\min}^f = \min(\|C_k - C_j\|, j \in N_k^f) \tag{3}$$

$$d_{\min}^b = \min(\|C_k - C_j\|, j \in N_k^b) \tag{4}$$

where C_k and C_j denote the mean color values for regions k and j, respectively. If both minimal distances are smaller than $d_{\min}$ or larger than $d_{\max}$, we classify this region into a distinct region and keep the original label. For other case, the region label will be updated in terms of the comparison result between two distances. It is noted that parameters $d_{\max}$ and $d_{\min}$ need to be defined at the initialization of this algorithm.

Algorithm 2 Region Grouping

```
Input: Region labels
        Number of regions $N_r$
        Thresholds $P_{\max}$, $d_{\max}$, $d_{\min}$
Output: New region labels
 1. Repeat steps (2) until no region labels change
 2. For $k = 1, ..., N_r$
        (a) Skip to next region if the area of region $k$ is larger than $P_{\max}$
        (b) Get region $R_k$, and compute the mean color value $C_k$
        (c) Find the neighboring regions $\mathcal{N}_k^f$ and $\mathcal{N}_k^b$ of region $k$ by performing the region dilation in terms of a $21 \times 21$ square structuring element. Here $\mathcal{N}_k^f$ and $\mathcal{N}_k^b$ denote the regions belonging to focus object and background, respectively.
        (d) Compute the minimal distance $d_{\min}^f = \min(\|C_k - C_j\|, j \in N_k^f)$ between $R_k$ and regions $\mathcal{N}_k^f$, $d_{\min}^b = \min(\|C_k - C_j\|, j \in N_k^b)$ between $R_k$ and regions $\mathcal{N}_k^b$.
        (e) No label change if $\min\{d_{\min}^f, d_{\min}^b\} > d_{\max}$ or $\max\{d_{\min}^f, d_{\min}^b\} < d_{\min}$
        (f) Assign focus label if $d_{\min}^f < d_{\min}^b$, otherwise background label.
 3. Output region labels
```

2) *Pixel Level Segmentation:* To refine the object boundary, we introduce a pixel-level segmentation method by minimizing the energy cost within a trimap, which consists of three regions, namely "focus region," "background region," and "unknown region." The focus regions are defined as the set of pixels belonging to the focus object obtained from the region segmentation stage, while the background regions consist of those pixels that are classified into the background. The trimap can be obtained based on the erosion and dilation operations of the initial focused object boundary using a 5×5 square structuring element. Note that the larger size of the structure element, the more computation load will be needed to group the pixels within the unknown region.

This paper uses the graph cut optimization method [24] to perform the pixel-level segmentation within the unknown region, which is based on a defined graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$ with a set of nodes $\mathcal{V}$ and a set of undirected edges $\mathcal{E}$ that connect these nodes. The segmentation can be achieved by the min-cut optimization, which can be expressed by solving an energy function based on two cost functions, i.e., the data cost U_1 and the smoothness cost U_2

$$U(Z) = \sum_{i \in \mathcal{V}} U_1(z_k) + \lambda_1 \sum_{\{k,l\} \in \mathcal{E}} U_2(z_k, z_l). \tag{5}$$

The data cost U_1 aims to assign each node to the focus or background label based on its distances to the focus and the background terminal nodes in RGB color space. In order to model the color distributions for background/foreground separation, we choose to use Gaussian mixture models (GMMs) that were already used in GrabCut [25]. The Gaussian mixture models with five and three components are employed to describe the focus and background colors, respectively. The mean and covariance of a component can be estimated based on the K-means algorithm. As described in the previous section, the obtained mean color values of merged regions can be used to determine the GMMs mean components for reducing the computational load.

The smoothness cost U_2 is used to measure the similarity between two nodes, which can be obtained from the local intensity gradient. The details of computing two costs can be referred to our previous work [26].

Of course, many other clustering methods are also available in this refinement procedure. For example, instead of using graph cut for the pixel-based refinement, we can determine the probabilities of unlabeled pixels with respect to pixels of known regions and assign them with class labels using random walker optimization.

III. Experiments

In this section, we evaluate our proposed segmentation method on several low depth-of-field images that were obtained from World Wide Web. Some subjective and objective assessment of segmentation results are reported.

A. Segmentation Results of Low-DOF Images

We collected 206 low-DOF images, which consist of human objects, flowers, trees, various animals, and so on. All image data are first scaled to the maximal width or height with 400 pixels. We then manually segmented all images into focused objects and defocused backgrounds, which are labeled as one and zero in the ground-truth mask, respectively. We split these images into two parts: 89 training images and 117 test images. Fig. 5 shows some training images and reference binary masks. In order to partition an image into multiple homogeneous regions, we choose to use a segmentation algorithm [14] with the threshold value $\theta = 1000$. Larger values for θ result in larger components in the result. Other parameters such as smoothness coefficient σ and minimum component size are set to the default values given by authors. A total of 980 and 468 positive and negative regions were obtained from 89 training images by using multiple segmentations.

As stated in the previous section, we extract three types of visual descriptors for each region. As shown in Fig. 6, we use three layers of classifiers in the cascade structure. Each layer includes 1448 descriptors for classifier training. The negative descriptors of each layer are obtained from the false outputs of the last classifiers. The final cascade classifier has about 92.23% detection rate for the training set. The number of weak classifiers for IVD, CVD, and GVD layers are set to 80, 60, and 60, respectively, which results in a total of 200 weak classifiers for the focus detection cascade.

In our experiments, three control parameters in the region grouping stage are set to $P_{\max} = 0.5 \times \text{Num}$, $d_{\max} = 50$, and $d_{\min} = 30$, respectively. Fig. 7(d) shows some results of running our method on several low-DOF images, including *Bee*, *Racoon*, *Dragonfly*, *Red-flower*, *Frog*, and *Judge* images. The original images are given in Fig. 7(a). The corresponding segmentation results are illustrated in Fig. 7(e), where the background regions are displayed in blue color. It can be seen that most of focused objects are detected and extracted from test images except for some small boundary artifacts such as the back of the animal in the image *Racoon*.

B. Comparison with Other Methods

We then compared our method with the existing methods [3] and [7]. The experimental parameters in the implementation were selected based on the original algorithms, such as the rectangular structuring element with the size of 31 × 31 in [7]. The segmentation results using method [7] are shown in Fig. 7(c), which contains incomplete segmentations for some test images, such as the head of the animal and the bottom part of the frog in the second and fifth rows in Fig. 7. The main reason is that some focused parts cannot be merged when there are not enough overlapping boundaries between two different focused regions [3]. Compared with the method [7], the missed detection is reduced by using the method [3], which is shown in Fig. 7(d). False detections are introduced in some images, such as the boundary of the flower image.

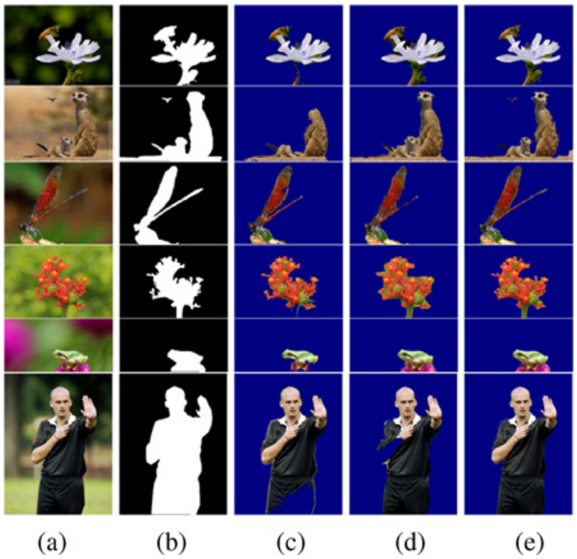


Fig. 7. Comparison results for test images, namely *Bee*, *Racoon*, *Dragonfly*, *Red-flower*, *Frog*, and *Judge* from top to bottom, respectively. (a) Original images. (b) Ground truth masks. (c) Results for method [7]. (d) Results for method [3]. (e) Results for our method.

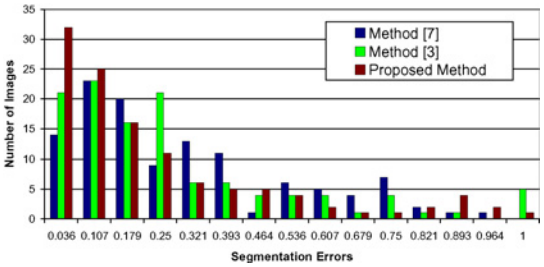


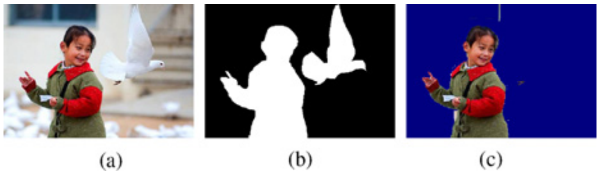
Fig. 8. Distribution of segmentation errors for 117 test images.

In order to evaluate the quality of our proposed method, we perform an objective comparison by computing the non-overlap pixels between the extracted masks and our hand-annotated ground-truth masks, which is usually employed to evaluate objective quality in [3] and [7]. We first manually segmented the reference maps (or ground truths) for the test images, some of which are shown in the second column of Fig. 7. We then define the measurement criterion by dividing the non-overlapping pixels by the total number of pixels in the binary reference mask, which represents the percentage of focus object pixels that were not detected.

The distribution of the segmentation errors for 117 test images is displayed in Fig. 8. This figure shows our algorithm peaking at a segmentation error of 0.0357, ahead of 0.1071 for the methods [3] and [7]. Table II shows that the segmentation errors of the displayed images in Fig. 7, which includes the results by methods [3] and [7]. In order to evaluate the performance of our method at different stages, we compute the results before pixel refinement and after this operation separately. From Table II, it is evident that our method outperforms the state-of-the-art.

TABLE II
PERFORMANCE EVALUATION BY OBJECTIVE CRITERION

Image	Method [7]	Method [3]	Proposed Algorithm (Before Pixel Refinement)	Proposed Algorithm (After Pixel Refinement)
Bee	0.2482	0.2173	0.1282	**0.0674**
Racoon	0.3223	0.1368	0.1134	**0.1095**
Dragonfly	0.1563	0.1754	0.1515	**0.0984**
Red-flower	0.1606	0.2310	0.2136	**0.1047**
Frog	0.3145	0.0676	0.0595	**0.0560**
Judge	0.1817	0.1035	0.0340	**0.0287**



(a) (b) (c)

Fig. 9. Example of missed detection for the focused object. (a) Original image. (b) Ground truth mask. (c) Segmentation result.

C. Discussions

There are still some shortcomings that we hope to address in future work. The original image and the ground truth mask are shown in Fig. 9(a) and (b), respectively. The segmentation result is illustrated in Fig. 9(c), in which a focused pigeon is not extracted in terms of the proposed algorithm. It is shown that our algorithm may be confused by a low saliency degree with respect to the defocused background. We hope to integrate the knowledge of object class and image enhancement technique to solve this problem. In addition, our algorithm sometimes confuses focused objects with some blurred parts.

It is noticed that three types of filters are considered to extract the texture feature in our work. These features exhibit good performance on focus detection [3], object discovery [17], [27], and texture classification [28]. In this subsection, we investigate the performance for different filter banks, which are partitioned into four groups. The first group only consists of four Laplacian operators, while eight LoGs are used in the second group. The third includes four Laplacian operators and eight LoGs. The filter bank defined in our proposed algorithm is evaluated as the final group. Based on the training data described in Section II-D, the fourth group achieves the lowest classification error after the boosting training. For the same training error, few weak classifiers are required for our filter bank. For example, given a fixed training error of 0.1195 for training images, about 119, 87, 70, and 54 weak classifiers are needed for groups 1, 2, 3, and 4, respectively. In addition, many other features, such as the first or second order derivatives of Gaussians, can also be used, which may result in more computational load.

We then investigate the performance of some MPEG-7 color descriptors on the CVD construction. These descriptors have very good performance for image/video retrieval in multimedia databases. There are a total of four color descriptors in MPEG-7 based on different color spaces, i.e., dominant color descriptor (DCD), scalable color descriptor (SCD), color layout descriptor, and color structure descriptor. Based on the training data given in Section II-D, our CVD achieves the lowest training error for 200 weak classifiers, which has 0.038 and 0.178 decreases in errors with respect to the SCD-based and DCD-based color visual descriptors, respectively. Here, 16 lowpass coefficients are used to generate a SCD.

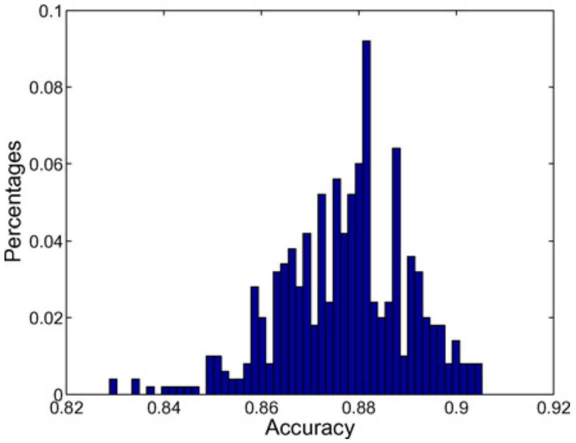


Fig. 10. Distribution of classification percentages when randomly choosing 45 training images from the training set of 89 images.

TABLE III
RESULTS OVER 500 RUNS OF CLASSIFICATION EXPERIMENT WHEN THE TRAINING IMAGES WERE CHOSEN RANDOMLY

Training Images	Mean (%)	Std (%)	Max (%)	Min (%)
45	87.8467	1.3908	91.2043	83.5589
22	84.6292	2.5771	90.0541	75.9134
11	79.2843	4.7487	88.2950	64.0054

We now investigate the effect of choice of training set on the classification performance. It could be argued that the given results may be biased by the selection of training and test images. We employ the method presented in [28] to address this issue, which repeats the classification experiment but with the training images chosen randomly. To avoid over-fitting, we first split the sample data into two non-overlapped parts: 89 images for training and 117 images for testing. The test images are held out and not looked at during training. We repeat the experiment 500 times by randomly selecting 45 images to form the training set. The distribution of classification results is illustrated in Fig. 10. The statistics for varying sizes of the training set is described in Table III, where the mean value of classification accuracy was 87.85% when the 45 images were chosen randomly. The result is very close to the 89.72% obtained when the all training images were chosen. This shows that our experimental setup is not biased and does not suffer from over-fitting to the data.

In addition, the computational complexity of the proposed method was evaluated by using the displayed images in Fig. 7. The testing computer has a Quad CPU 2.66 GHz, and 3.00 GB of RAM. The Graph Cut algorithm for the pixel-level refinement was run from the executable file. The time of the multiple segmentations by [14] is not included in the running time. The proposed method was implemented using MATLAB version R2007b, which has an average computation time of 4.635 s comparable to the existing methods such as 7.9967 s on average for the method [7].

IV. Conclusion

In this paper, we developed a method to segment focused objects from low-DOF images. To extract focused regions, an over-segmentation method is employed to generate multiple regions from each input image. Three types of visual descriptors are designed to describe region features. The focus detection of a region can be achieved by classifying visual descriptors into the focus or background class in terms of a cascade structured classifier trained using a boosting algorithm. In order to improve segmentation results, we employ a two-level segmentation method, which includes region grouping and pixel-level segmentation. Experimental results were obtained by applying the proposed method to several low-DOF images. It has been shown that our method outperforms the state-of-the-art methods for the focused object segmentation.

Acknowledgment

The authors would like to thank all the anonymous reviewers for the constructive comments and useful suggestions that led to improvements in the quality, presentation, and organization of this paper.

References

[1] L. Ladický, C. Russell, and P. Kohli, "Associative hierarchical CRFs for object class image segmentation," in *Proc. IEEE ICCV*, Sep.–Oct. 2009, pp. 739–746.

[2] K. N. Ngan and H. Li, "Semantic object segmentation," *IEEE Commun. Soc. Multimedia Commun. Tech. Committee E-Lett.*, vol. 4, no. 6, pp. 6–8, Jul. 2009.

[3] H. Li and K. N. Ngan, "Unsupervised video segmentation with low depth of field," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 12, pp. 1742–1751, Dec. 2007.

[4] J. Z. Wang, J. Li, R. M. Gray, and G. Wiederhold, "Unsupervised multiresolution segmentation for images with low depth of field," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 1, pp. 85–90, Jan. 2001.

[5] D.-M. Tsai and H.-J. Wang, "Segmenting focused objects in complex visual images," *Pattern Recognit. Lett.*, vol. 19, no. 10, pp. 929–940, Aug. 1998.

[6] C. S. Won, K. Pyun, and R. M. Gray, "Automatic object segmentation in images with low depth of field," in *Proc. IEEE ICIP*, vol. 3. Jun. 2002, pp. 805–808.

[7] C. Kim, "Segmenting a low-depth-of-field image using morphological filters and region merging," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1503–1511, Oct. 2005.

[8] Z. Ye and C.-C. Lu, "Unsupervised multiscale focused objects detection using hidden Markov tree," in *Proc. Int. Conf. Comput. Vision Pattern Recognit. Image Process.*, Mar. 2002, pp. 812–815.

[9] L. Kovacs and T. Sziranyi, "Focus area extraction by blind deconvolution for defining regions of interest," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1080–1085, Jun. 2007.

[10] E. Izquierdo and M. Ghanbari, "Key components for an advanced segmentation toolbox," *IEEE Trans. Multimedia*, vol. 4, no. 1, pp. 97–113, Mar. 2002.

[11] J. Yuan, Y. Wu, and M. Yang, "Discovery of collocation patterns: From visual words to visual phrases," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun. 2007, pp. 1–8.

[12] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun. 2006, pp. 17–22.

[13] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," in *Proc. IEEE ICCV*, vol. 1. Oct. 2005, pp. 654–661.

[14] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[15] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[16] L. Cao and L. Fei-Fei, "Spatially coherent latent topic model for concurrent object segmentation and classification," in *Proc. IEEE ICCV*, Oct. 2007, pp. 1–8.

[17] X. Wang and E. Grimson, "Spatial latent Dirichlet allocation," in *Proc. Neural Inform. Process. Syst. Conf.*, Dec. 2007, pp. 1577–1584.

[18] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proc. IEEE ICCV*, Sep.–Oct. 2009, pp. 1–8.

[19] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[20] J. Han, K. N. Ngan, M. Li, and H. J. Zhang, "Unsupervised extraction of visual attention objects in color images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 1, pp. 141–145, Jan. 2006.

[21] H. Li and K. N. Ngan, "Saliency model based face segmentation in head-and-shoulder video sequences," *J. Visual Commun. Image Represent.*, vol. 19, no. 5, pp. 320–333, Apr. 2008.

[22] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. CVPR*, vol. 1. Dec. 2001, pp. 511–518.

[23] S. Z. Li and Z. Q. Zhang, "FloatBoost learning and statistical face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1112–1123, Sep. 2004.

[24] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *Proc. IEEE ICCV*, vol. 1. Jul. 2001, pp. 105–112.

[25] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," in *Proc. SIGGRAPH*, Aug. 2004, pp. 309–314.

[26] H. Li, K. N. Ngan, and Q. Liu, "FaceSeg: Automatic face segmentation for real-time video," *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 77–88, Jan. 2009.

[27] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *Proc. IEEE ICCV*, vol. 2. Oct. 2005, pp. 1800–1807.

[28] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *Int. J. Comput. Vision*, vol. 62, nos. 1–2, pp. 61–81, Apr. 2005.

[29] S. B. Gray, "Local properties of binary images in two dimensions," *IEEE Trans. Comput.*, vol. C-20, no. 5, pp. 551–561, May 1971.

Hongliang Li (M'06) received the Ph.D. degree in electronics and information engineering from Xi'an Jiaotong University, Xi'an, China, in 2005.

From 2005 to 2006, he joined the Visual Signal Processing and Communication Laboratory, Chinese University of Hong Kong (CUHK), Shatin, Hong Kong, as a Research Associate. From 2006 to 2008, he was a Post-Doctoral Fellow with the same laboratory in CUHK. He is currently a Professor with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, China. His current research interests include image segmentation, object detection and tracking, image and video coding, and multimedia communication systems.

King N. Ngan (M'79–SM'91–F'00) received the Ph.D. degree in electrical engineering from Loughborough University, Loughborough, U.K.

He is currently a Chair Professor with the Department of Electronic Engineering, Chinese University of Hong Kong, Shatin, Hong Kong. He was previously a Full Professor with Nanyang Technological University, Singapore, and with the University of Western Australia, Crawley, Australia. He holds honorary and visiting professorships with numerous universities in China, Australia, and South East Asia. He has published extensively including 3 authored books, 5 edited volumes, and over 300 refereed technical papers, and edited 9 special issues in journals. He holds ten patents in the areas of image/video coding and communications.

Prof. Ngan is an Associate Editor of the *Journal on Visual Communications and Image Representation*, as well as an Area Editor of the *EURASIP Journal of Signal Processing: Image Communication*. He served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the *Journal of Applied Signal Processing*. He chaired a number of prestigious international conferences on video signal processing and communications, and served on the advisory and technical committees of numerous professional organizations. He was a General Co-Chair of the IEEE International Conference on Image Processing, Hong Kong, in September 2010. He is a fellow of IET, U.K., and IEAust, Australia, and was an IEEE Distinguished Lecturer from 2006 to 2007.