Arbitrary-shape object localization using adaptive image grids

Yuan, Junsong; Zhou, Chunluan

2013

Zhou, C., & Yuan, J. (2013). Arbitrary-shape object localization using adaptive image grids. Proceedings of the 11th Asian conference on Computer Vision (ACCV12), pp.71-84.

<https://hdl.handle.net/10356/100685>

https://doi.org/10.1007/978-3-642-37331-2_6

© 2013 Springer-Verlag Berlin Heidelberg. This is the author created version of a work that has been peer reviewed and accepted for publication by Proceedings of the 11th Asian conference on Computer Vision (ACCV12), Springer-Verlag Berlin Heidelberg. It incorporates referee's comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [http://dx.doi.org/10.1007/978-3-642-37331-2_6].

Downloaded on 26 Apr 2023 20:33:41 SGT

Arbitrary-Shape Object Localization using Adaptive Image Grids

Chunlun Zhou and Junsong Yuan

School of EEE, Nanyang Technology University, Singapore

Abstract. Sliding-window based search is a widely used technique for object localization. However, for objects of non-rectangle shapes, noises in windows may mislead the localization, causing unsatisfactory results. In this paper, we propose an efficient bottom-up approach for detecting arbitrary-shape objects using image grids as basic components. First, a test image is partitioned into $n \times n$ grids and the object is localized by finding a set of connected grids which maximize the classifier’s response. Then, graph cut segmentation is used to improve the object boundary by utilizing local image context. Instead of using bounding boxes, the proposed approach searches connected regions of any shapes. With the graph cut refinement, our approach can start with coarse image grids and is robust to noises. To make image grids better cover the object of arbitrary shape, we also propose a fast adaptive grid partition method which takes image content into account and can be efficiently implemented by dynamic programming. The use of adaptive partition further improves the localization accuracy of our approach. Experiments on PASCAL VOC 2007 and VOC 2008 datasets demonstrate the effectiveness of our approach.

1 Introduction

Given an image containing one or more objects of interest, object localization is to determine the position and scope of each object in the image. Object localization plays an important role in object detection systems [6, 8, 10, 24], which need to determine whether there is any object of interest present in an image, and if any, where it is. Since the object of interest may only occupy a small region, it is usually difficult to judge its presence by the whole image. Generally, object detection first performs localization to obtain several candidates by assuming that the object is present, followed by a recognition step to verify each candidate. In such a case, localization accuracy is of fundamental importance to the subsequent recognition.

Sliding window search is widely used for object localization [1, 8, 11, 12]. This localization approach searches all possible windows in an image, evaluating a score for each window, and then selects the windows with the highest scores as candidates for further verification. Since the object size is unknown in advance, for an $m \times n$ image, there are $O(m^2n^2)$ possible windows to be checked, leading to a high computational cost. Another drawback of this approach is that it

Fig. 1. A failed example of ERS. (a) Original image with positive-contribution features (red dots) and negative-contribution features (blue dots). (b) Super-pixel score map. Light regions and dark regions respectively denote positively scored and negatively scored super-pixels. (c) Optimal connected region. Some positively scored background super-pixels are included, while a few object super-pixels are lost due to their negative scores.

approximates the object shape as a rectangle. For non-rectangle objects, noises in the windows may misguide the localization [27].

To solve the above problems of sliding window search, efficient region search (ERS) [25] was recently proposed for localizing arbitrary-shape objects. This approach first segments a test image into super-pixels and each super-pixel is assigned a score by summing the contributions of its features. The object region is localized by selecting a set of connected super-pixels with the highest total score. The selection of super-pixels is converted to a graph search problem which is efficiently solved by a branch-and-cut algorithm. However, this approach largely depends on the quality of generated super-pixels. If some super-pixels cover both object and background areas, part of the object region will be lost or part of the background region will be included in the localization result. The presence of noisy super-pixels, i.e. positive-score background super-pixels and negative-score object super-pixels, may also cause unsatisfactory results. Figure 1 shows a failed example of this approach. To exclude background super-pixels, the authors proposed to introduce contours into ERS (ERS-C). However, ERS-C depends on the robustness of contour evaluation, which makes it difficult to guarantee the performance.

To address the limitations of ERS, we propose an efficient bottom-up approach using image grids. An overview of our approach is shown in Fig. 2. A test image is uniformly partitioned into $n \times n$ grids and the branch-and-cut search is applied to these grids, instead of super-pixels, to obtain an optimal connected region (Fig. 2(a-c)). Then, the obtained region is enlarged and graph cut segmentation is applied to the enlarged region for refinement, i.e. retrieving lost object pixels and excluding background pixels (Fig. 2(d-f)). The introduction of graph cut refinement in our approach has two advantages. First, this refinement step makes it possible to use coarse basic components, e.g. image grids, which can be easily obtained. This is important to speed up the localization process as it is usually time-consuming to obtain high-quality super-pixels, like the segmentation method [2] used in ERS. Second, it makes our approach more robust to noises. To improve the localization accuracy, we also propose an efficient adaptive grid

Fig. 2. Overview of our approach. (a) Original image with local features. (b) Grid score map. (c) Optimal connected region. (d) Initial background and foreground seeds for graph cut segmentation. Blue grids are newly added and selected as background seeds. Red grids are positive-score grids in the optimal region and selected as foreground seeds. Yellow grids are negative-score grids included by grid search to form the optimal region. The yellow grids may belong to either foreground or background and are left unlabeled initially. Graph cut segmentation relabels all the pixels in the image except the black area. (e) Result of graph cut segmentation. (f) Final object region.

partition method which takes image content into account. We demonstrate the effectiveness of our approach on PASCAL VOC 2007 and VOC 2008 datasets.

2 Related Work

Several approaches have been proposed for localizing arbitrary-shaped objects. In [26], extensions are made to efficient subwindow search [16] for localizing non-rectangle regions, k -composite boxes and k -side polygons, respectively. However, the parameter k needs to be pre-determined, and to make the search efficient, its value cannot be too large, thereby limiting the shapes of objects the approach can detect. Voting approaches [17, 18, 21] are suitable for localizing objects of various shapes, but they may fail when the objects to be localized differ largely in shape and viewpoint from those used for training. Free-shape subwindow search [27] locates the object by connecting edge segments to form a closed region with its boundary and inner features maximizing the localization objective function. The approaches proposed in [13, 23, 25] use super-pixels as basic components and localize the object region by selecting a subset of super-pixels. Different from [13, 23, 25, 27], our approach starts with coarse image grids, which are also used for object search [14, 15] recently.

There are many object localization/detection approaches using segmentation for further processing [7, 13, 17, 19, 20, 22, 28]. In [17, 20, 22], more accurate

segmentation is inferred from each localized object candidate and then used to verify the localized candidate. Several approaches [7, 13, 28] use segmentation to exclude background pixels from the localized object region and retrieve lost object pixels, as we do in this paper. With the observation that animals like cats and dogs have homogenous color and texture, segmentation is used to detect the rest parts of the body after the head is detected [19].

3 Grid-based Object Localization

In this section, we first describe efficient region search [25], and then introduce our image-grid based object localization approach.

3.1 Efficient Region Search (ERS)

First, bag-of-visual-words representation and linear SVM are used to train a classifier. A set of features (e.g. SURF [3]) are collected from training images and quantized into a codebook of K visual words. Using the codebook, an image region R containing features d_i , $i = 1, \dots, M$, is represented by a K -bin histogram $h(R)$. The classification function f is learned by linear SVM:

$$f(R) = \sum_{j=1}^N \alpha_j \langle h(R), h(R_j) \rangle + \beta \quad (1)$$

where N is the number of training examples, R_j is the j -th training example, and α_j and β are the weights and the bias respectively.

Let $h_k(R_j)$ be the count of the k -th bin of the histogram $h(R_j)$ and the expression of $f(R)$ is rewritten as

$$f(R) = \sum_{k=1}^K h_k(R) \sum_{j=1}^N \alpha_j h_k(R_j) + \beta = \sum_{i=1}^M c_{w_i} + \beta \quad (2)$$

where $c_k = \sum_{j=1}^N \alpha_j h_k(R_j)$ represents the contribution of the k -th visual word and w_i is the index of the visual word associated with d_i . Therefore, the score of region R is the sum of the contributions of its features plus the bias β .

Given a test image, the goal is to localize an optimal region that maximizes the learned classification function f . The image is first segmented into several super-pixels and each super-pixel is assigned a score according to the features extracted from the super-pixel by Eq. (2). And then object region is obtained by identifying a set of connected super-pixels with the highest total score. By viewing each super-pixel as a graph node with the weight its score and adding edges between adjacent super-pixels, the localization problem is converted to the following maximum-weight subgraph search (MWCS) problem:

MWSC Problem Given an undirected, vertex-weighted graph $G = (V, E)$

Fig. 3. An example of uniform partition (a) and adaptive partition (b).

with weights $w : V \rightarrow \mathbb{R}$, find a connected subgraph $T = (V_T, E_T)$ which maximizes the score $W(T) = \sum_{v \in V_T} w(v)$.

With both positive and negative node weights, the MWCS problem is NP-complete, but in practice it can be efficiently solved by a branch-and-cut algorithm. See [25] for details.

3.2 Region Search using Image Grids

For ERS, the quality of generated super-pixels has a considerable impact on the final localization result. The authors of ERS adopted a semantic segmentation method to obtain high-quality super-pixels but sacrificed computational efficiency. To speed up the component initialization, we start with a coarse grid partition. First, a test image is uniformly partitioned into $n \times n$ grids and each grid is assigned a score as described in section 3.1. By regarding grids as nodes and adding edges between each grid and its neighboring grids, we construct an undirected vertex-weighted graph. Then, the object region is localized by applying the branch-and-cut algorithm to the constructed graph. Figure 2(a-c) shows the search process. We refer to this process as grid search.

With uniform partition, some grids may be located at the object boundary, thus affecting the score evaluation of these grids. To make image grids better cover the object region, we propose an adaptive grid partition method: first, the image is evenly divided into n bands in the horizontal direction; then, each band is vertically divided into n grids satisfying the following two requirements:

1. the color distribution in each grid are as concentrated as possible;
2. the color difference between each two adjacent grids is as large as possible.

The first requirement forces grids to move off the object boundary and the second one prevents a region of homogeneous colors from being broken into many grids. Figure 3 shows an example of uniform partition and adaptive partition.

Now, we describe how to partition a band, denoted by B , of width w into n grids. A partition of the band into n grids is represented by $P = \{p_i\}_{i=1}^{n+1}$, where p_i is the position of the i -th vertical line and satisfies (1) $p_1 = 1$, (2) $p_{n+1} = w + 1$, and (3) $p_i < p_{i+1}$ for $1 < i \leq n$. We denote a sub-band starting

from position s to position t (s and t are included) by $B_{s,t}$ with $1 \leq s \leq t \leq w$. So, the i -th grid of the partition is the sub-band $B_{p_i, p_{i+1}-1}$, i.e. $g_i = B_{p_i, p_{i+1}-1}$. We obtain an optimal partition by minimizing the following cost function:

$$E(P) = \sum_{i=1}^n C(g_i) + \gamma \sum_{i=1}^{n-1} D(g_i, g_{i+1}) \quad (3)$$

where $C(g_i)$ is the color concentration of g_i , $D(g_i, g_{i+1})$ is the color difference between g_i and g_{i+1} , and γ is a parameter.

To calculate $C(g_i)$, we divide the RGB color space into $M = 16 \times 16 \times 16$ bins and compute for g_i a color histogram $h(g_i)$ with $h_j(g_i)$ the count of the j -th bin. By normalizing $h(g_i)$ to sum to 1, we define the color concentration of g_i using information entropy:

$$C(g_i) = \sum_{j=1}^M -h_j(g_i) \log(h_j(g_i)) \quad (4)$$

With this definition, the more concentrated the colors of pixels in g_i are, the smaller the value of $C(g_i)$ is. Typically, when the grid contains only one type of color, $C(g_i) = 0$.

We represent the color of a grid by the average color of the pixels in the grid. Some other representations are also applicable, e.g. color histogram. The color difference between g_i and g_{i+1} is defined as

$$D(g_i, g_{i+1}) = \exp\left(-\frac{\|c_{g_i} - c_{g_{i+1}}\|^2}{2\sigma^2}\right) \quad (5)$$

where $c_{g_i} \in \mathbb{R}^3$ is the color of g_i and $\sigma = 32$ is used in our experiments.

Let $S(i, t)$ be the cost of the optimal partition of the sub-band $B_{1,t}$ into i grids and $T(i, s, t)$ be the minimum cost of partitioning $B_{1,t}$ into i grids with the last grid $g_i = B_{s,t}$. With these two definitions, minimizing the cost function in Eq. (3) is equivalent to compute $S(n, w) = \min_{p_n} T(n, p_n, w)$. By fixing p_n and enumerating the $(n-1)$ -th grid, $g_{n-1} = B_{p_{n-1}, p_n-1}$, we have

$$T(n, p_n, w) = \min_{p_{n-1}} T(n-1, p_{n-1}, p_n-1) + C(g_n) + \gamma D(g_{n-1}, g_n) \quad (6)$$

Generally, $T(i, s, t)$ has the following recursive solution:

$$T(i, s, t) = \begin{cases} C(g_1) & i = 1 \\ \min_r T(i-1, r, s-1) + C(g_i) + \gamma D(g_{i-1}, g_i) & i > 1 \end{cases} \quad (7)$$

where r , s and t determine the last two grids: $g_{i-1} = B_{r, s-1}$ and $g_i = B_{s, t}$. For the special case $i = 1$, the partition consists of only one grid, i.e. $g_1 = B_{1, t}$, so the partition cost is the color concentration of g_1 .

According to Eq. (7), we design a dynamic programming algorithm for partitioning a band into n grids (see Alg. 1). With $C(g_i)$ and $D(g_{i-1}, g_i)$ pre-computed, this algorithm runs in $O(nw^3)$ time. As there are n bands to be

Algorithm 1 Adaptive partition for one band**Require:** a band B , band width w and partition parameters n and γ ;**Ensure:** the optimal partition $P^* = \{p_1^*, p_2^*, \dots, p_{n+1}^*\}$;

```

1:  $T(i, s, t) \leftarrow \infty, \forall i, s, t$ ;
2: for all  $t \in [1, w]$  do
3:    $g_1 \leftarrow B_{1,t}, T(1, 1, t) \leftarrow C(g_1)$ ;
4: end for
5: for  $i \leftarrow 2$  to  $n$  do
6:   for all  $t \leq w$  do
7:     for all  $s < t$  do
8:        $g_i \leftarrow B_{s,t}$ ;
9:       for all  $r < s$  do
10:         $g_{i-1} \leftarrow B_{r,s-1}, c \leftarrow T(i-1, r, s-1) + C(g_i) + \gamma D(g_{i-1}, g_i)$ ;
11:        if  $c < T(i, s, t)$  then
12:           $T(i, s, t) \leftarrow c$ ;
13:        end if
14:      end for
15:    end for
16:  end for
17: end for
18:  $p_1^* \leftarrow 1, p_{n+1}^* \leftarrow w + 1$ ;
19: for  $i \leftarrow n$  to  $2$  do
20:    $p_i^* \leftarrow \arg \min_{p_i} T(i, p_i, p_{i+1}^*)$ ;
21: end for

```

partitioned, the total running time for adaptive partition is $O(n^2 w^3)$. To make adaptive partition robust to cluttered images, we restrict the width of each grid to the range $[\frac{1}{2}w_G, \frac{3}{2}w_G]$, where $w_G = \frac{w}{n}$ is the width of a grid obtained by uniform partition. With this restriction, the time needed to partition one band is reduced to $O(n \cdot w \cdot \frac{w}{n} \cdot \frac{w}{n}) = O(\frac{w^3}{n})$ with the total partition time $O(w^3)$. For a high-resolution image (e.g. 1280×1024 pixels), we first scale it down (e.g. to 320×256 pixels) and then perform the partition on the low-resolution image. Therefore, our adaptive partition is fast in practice.

3.3 Refinement with Graph Cut Segmentation

Due to noisy local features, a background grid may be assigned a positive score and included in the localized region. Similarly, some object grids possibly have negative scores and may therefore be excluded. We need to exclude the background pixels from the result region and retrieve the lost object pixels. To achieve this, we first enlarge the localized region to exploit local image context: the grids adjacent to the localized region are added to form a larger region (Fig. 2(d)). Then, graph cut segmentation [4] is applied to the enlarged region to refine the object boundary.

Graph cut segmentation divides the enlarged region into two parts, foreground (the object region) and background. Let X be the set of pixels in the

enlarged region and Y be the set of adjacent pixel pairs. The goal is to seek a labeling $L = \{l_1, \dots, l_x, \dots, l_{|X|}\}$ such that $l_x = 1$ if pixel x belongs to the foreground; otherwise, $l_x = 0$. The labeling L is obtained by minimizing the following cost function:

$$E(X) = \sum_{x \in X} U(x) + \lambda \sum_{(x,y) \in Y} V(x,y) \cdot \delta(l_x, l_y) \quad (8)$$

where

$$\delta(l_x, l_y) = \begin{cases} 0 & \text{if } l_x = l_y \\ 1 & \text{otherwise} \end{cases} \quad (9)$$

Here $U(x)$ is the cost of assigning pixel x to the foreground or the background, $V(x, y)$ is the cost incurred when two adjacent pixels x and y have different labels and λ is a parameter. In this paper, we will focus on how to define $U(x)$ and $V(x, y)$. We refer readers to [4] for a fast solution to the minimization of Eq. (8).

To calculate $U(x)$, we select the positive-score grids in the localized region as foreground seeds and the newly added grids as background seeds (Fig. 2(d)). By dividing the RGB color space into $16 \times 16 \times 16$ bins, a foreground color histogram h^{fg} and a background color histogram h^{bg} are computed according to the foreground seeds and the background seeds, respectively, and each histogram is normalized with a summation of 1. With these two histograms, we estimate $p(x|l_x = 1) = h_{b_x}^{fg}$ and $p(x|l_x = 0) = h_{b_x}^{bg}$, where b_x is the bin index of pixel x in the histogram. By assuming $P(l_x = 1) = P(l_x = 0) = \frac{1}{2}$, we have the following two distributions

$$p(l_x = 1|x) = \frac{p(x|l_x = 1)P(l_x = 1)}{P(x)} = \frac{h_{b_x}^{fg}}{h_{b_x}^{fg} + h_{b_x}^{bg}} \quad (10)$$

$$p(l_x = 0|x) = \frac{p(x|l_x = 0)P(l_x = 0)}{P(x)} = \frac{h_{b_x}^{bg}}{h_{b_x}^{fg} + h_{b_x}^{bg}} \quad (11)$$

Using Eq. (10) and Eq. (11), we define $U(x)$ as

$$U(x) = -\log(p(l_x|x)) \quad (12)$$

As a result, the more likely a pixel x belongs to the foreground(background), the less the cost is incurred if l_x is set to 1(0).

We define $V(x, y)$ as

$$V(x, y) = \exp\left(-\frac{\|c_x - c_y\|^2}{2\sigma^2}\right) \quad (13)$$

where c_x is the color of pixel x , and as in Eq. (5), σ is set to 32. If two adjacent pixels of similar colors are labelled differently, the value of $V(x, y)$ is large. Therefore, $V(x, y)$ imposes a smoothness constraint on the segmentation result.

To speed up the segmentation process, we scale the original test image to a lower resolution of 120×120 pixels. Graph cut segmentation is run on the smaller image to get the foreground which is then scaled up to the original size. After the graph cut segmentation, the obtained foreground may consist of several isolated regions (Fig. 2(e)). We select the region which maximizes the learned classification function as the final result (Fig. 2(f)).

3.4 Partition Parameter Selection

A coarse grid partition (n is small) cannot cover the object region precisely, while a fine grid partition (n is large) may make the grids too small to be sensitive to noisy features. In addition, a large n causes a high computational cost for grid search. Therefore, the value of n should not be too small or too large. Furthermore, since the objects to be localized can be of any sizes and at any positions, a single fixed partition may not work well for all cases. Without the prior knowledge of the object size and position, we apply to a test image several partitions which cover different scales, e.g. 9×9 , 11×11 and 13×13 . From these partitions, the optimal one is selected: first, for each partition, grid search and graph cut refinement are performed to get an object region; then, from these regions the one with the highest score is selected as the final result.

4 Experiments

To demonstrate the effectiveness of the proposed approach, we conduct object localization on PASCAL VOC 2007 and VOC 2008 datasets.

4.1 Experiment on VOC 2007 Dataset

We take cat and dog in the PASCAL VOC 2007 localization dataset as test categories. In this dataset, cats and dogs change in pose and viewpoint, and take on various shapes in images, so they are suitable for testing arbitrary-shape object localization. We compare our approach (GS-GC), i.e. grid search (GS) followed by graph cut (GC) refinement, with two state-of-the-art methods: efficient subwindow search (ESS) [16] and efficient region search (ERS) [25]. The former produces bounding-box localization results, and as ours, the latter produces object regions of any shapes.

The image set of each category consists of three parts: training set, validation set and testing set. We use the training set and the validation set for training and the testing set for evaluation. For local features, we extract SURF [3] from each image.¹ A codebook of 2000 visual words is generated for each category by collecting 500,000 features and quantizing them by k -means. As positive examples, we use the segmented training examples provided by the authors of [25]. We randomly sample some rectangular patches from image regions that are

¹ We keep all the features, while in [25] only features at Canny edge points are used.

Fig. 4. Localization accuracy at bounding-box level. **Fig. 5.** Localization accuracy at pixel level.

labelled as background or negative classes. The parameter C for training linear SVM is selected by cross-validation. The parameter λ for graph cut segmentation and the parameter γ for adaptive partition are tuned so that our approach performs best on the validation set. For ESS, we use the authors' published code. The super-pixels used in ERS are kindly provided by the authors. To make a direct comparison, the same features and SVM models are used for ESS, ERS and our approach. Following [25], we apply each method to localize the optimal region in each test image and the confidence of the localized region is evaluated by a χ^2 -kernel SVM learned from the same training examples.

For evaluation, we use two types of precision-recall curves at bounding-box and pixel levels, respectively. At bounding-box level, a localized region is considered positive if the intersection area of the ground-truth bounding box and the bounding box that tightly surrounds the region divided by their union area is greater than 0.5. At pixel level, a pixel in the localized region is considered positive if it belongs to the true object region. Figures 4 and 5 show the results of ESS, ERS and our approach. For the two implementations of our approach, one with uniform partition (UP) and the other with adaptive partition (AP), we apply to each test image there different grid partitions, 9×9 , 11×11 and 13×13 . Both ERS and our approach outperform ESS, which indicates the advantage of arbitrary-shape region search over rectangle region search. Compared with ERS, our two implementations yield better localization accuracy with GS-

Table 1. Bounding-box level average precision of our approach using a single partition

Category	cat							dog						
n	8	9	10	11	12	13	14	8	9	10	11	12	13	14
GS(UP)	32.6	31.2	29.1	30.0	28.6	29.4	29.1	19.5	20.7	18.9	19.5	20.2	18.8	18.3
GS-GC(UP)	41.8	41.8	37.9	41.9	38.4	45.3	39.8	27.4	27.3	28.7	31.7	31.0	29.6	30.1
GS(AP)	33.6	30.4	29.6	30.1	29.5	30.4	28.9	21.0	20.1	21.2	19.4	20.1	18.2	18.3
GS-GC(AP)	42.9	45.0	41.3	44.6	44.0	42.3	42.7	28.6	29.2	30.9	35.3	29.9	30.8	30.1

Table 2. Pixel level average precision of our approach using a single partition

Category	cat							dog						
n	8	9	10	11	12	13	14	8	9	10	11	12	13	14
GS(UP)	32.6	31.3	30.5	30.5	28.7	28.0	28.7	28.6	28.0	26.8	26.5	26.5	25.3	24.6
GS-GC(UP)	43.3	44.0	44.4	44.8	44.4	45.9	44.4	41.7	41.8	42.3	43.1	43.1	43.1	42.7
GS(AP)	33.3	32.0	30.5	30.7	29.7	28.9	28.2	29.4	28.8	27.7	27.3	26.8	25.5	25.1
GS-GC(AP)	45.8	45.1	44.4	45.7	45.6	47.6	45.4	42.7	42.4	41.1	44.7	42.8	42.9	43.0

GC(AP) performing best. Tables 1 and 2 give the results of our approach using a single grid partition with $8 \leq n \leq 14$. Due to the coarse initialization of image grids, GS(UP) and GS(AP) perform worse than ERS, but for all different n 's, GS-GC(UP) and GS-GC(AP) outperform ERS at both bounding-box and pixel levels, showing the effectiveness of the graph cut refinement. The advantage of using adaptive partition can be seen from Tabs. 1 and 2 that GS-GC(AP) improves the localization accuracy over GS-GC(UP) for most partitions. In addition, the use of multiple partitions further improves the performance, because it reduces the risk of single-partition failure and makes our approach more stable. Figure 6 shows some localization examples of ESS, ERS and our approach.

We run all the three methods on a PC with a 3.2 GHz CPU. On average, the optimal region search takes about 0.06s for ESS and 0.26s for ERS (the time for segmentation is not counted). The average runtime of GS using a single partitions is listed in Tab. 3. When $n \leq 11$, the runtime of GS and that of ERS are similar, as in one test image the number of super-pixels (about 100) is close to that of image grids ranging from 64 to 121. The runtime of GS gets longer as n increases and we restrict $n < 15$. For graph cut segmentation, we use an efficient implementation of a min-cut/max-flow algorithm [5]. The average time to run graph cut segmentation on a test image is about 0.12s.

4.2 Experiment on VOC 2008 Dataset

We further compare our approach with ERS on 20 object categories in the VOC 2008 segmentation dataset. Following the experimental setting in [25], we split the *trainval* set of each category into three parts and report the 3-fold cross-validation accuracy using the PASCAL segmentation criterion [9], overlap accuracy, defined by $\frac{TP}{TP+FP+FN}$, where TP , FP and FN are total true positives, false positives and false negatives, respectively. For each test image, 9×9 , 11×11 and 13×13 grid partitions are used. We set $\lambda = 1.0$ and $\gamma = 3.0$ for all the 20

Table 3. Average runtime of grid search using single partitions

Method	GS(UP)							GS(AP)						
n	8	9	10	11	12	13	14	8	9	10	11	12	13	14
Average time (s)	0.13	0.18	0.28	0.55	1.12	2.52	4.90	0.16	0.22	0.33	0.56	0.94	1.76	4.96

Fig. 6. Localization examples of different approaches. Results of EES, ERS and our approach are given in Row 1, Row 2 and Row 3 respectively.

categories. Table 4 shows the results of 20 object categories. GS-GC(UP) and GS-GC(AP) outperform ERS on 16 and 17 categories respectively and improve the mean overlap accuracy on 20 categories: 0.274 for ERS, 0.295 for GS-GC(UP) and 0.3 for GS-GC(AP). Moreover, GS-GC(AP) improves the overlap accuracy over GS-GS(UP) on 14 categories. Some localization examples of our approach are given in Fig. 7.

5 Conclusions

We present an efficient bottom-up approach for arbitrary-shape object localization. Based on coarse image grids, the object region in an image is localized by searching an optimal connected region composed of image grids using the branch-and-cut algorithm, followed by graph cut segmentation for refinement. We also propose an efficient adaptive grid partition method which helps further improve the localization accuracy of our approach. We demonstrate the advantage of our approach on PASCAL VOC 2007 and VOC 2008 datasets.

Acknowledgement. This work was supported in part by the Nanyang Assistant Professorship (SUG M58040015) to Dr. Junsong Yuan. We thank Gangqiang Zhao for the help to proofread the paper.

References

1. Agarwal, S., Roth, D.: Learning a sparse representation for object detection. European Conference on Computer Vision. (2002) 113-130

Table 4. Overlap accuracy on 20 categories of VOC 2008 dataset

Category	aerop.	bicyc.	bird	boat	bottle	bus	car	cat	chair	cow
ERS	0.324	0.109	0.268	0.262	0.121	0.405	0.244	0.389	0.120	0.324
GS-GC(UP)	0.380	0.154	0.286	0.218	0.148	0.459	0.264	0.435	0.148	0.332
GS-GC(AP)	0.361	0.158	0.295	0.236	0.152	0.483	0.273	0.448	0.140	0.343
Category	dinin.	dog	horse	motor.	person	potte.	sheep	sofa	train	tvmon.
ERS	0.300	0.288	0.280	0.337	0.257	0.119	0.394	0.224	0.453	0.259
GS-GC(UP)	0.319	0.320	0.278	0.412	0.284	0.134	0.394	0.206	0.459	0.263
GS-GC(AP)	0.312	0.330	0.278	0.397	0.290	0.143	0.409	0.234	0.452	0.267

Fig. 7. Localization examples of our approach.

- Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From Contours to Regions: An Empirical Evaluation. IEEE Conference on Computer Vision and Pattern Recognition. (2009) 2294-2301
- Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding **110** 2008 346-359
- Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. IEEE International Conference on Computer Vision. (2001) 105-112
- Boykov, Y., Kolmogorov, V.: An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. IEEE Transaction on Pattern Analysis and Machine Intelligence **26** (2004) 1124-1137
- Chum, O., Zisserman, A.: An exemplar model for learning object classes. IEEE Conference on Computer Vision and Pattern Recognition. (2007) 1-8
- Dai, Q., Hoiem, D.: Learning to localize Detected Objects. IEEE Conference on Computer Vision and Pattern Recognition. (2012) 3322-3329
- Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. IEEE Conference on Computer Vision and Pattern Recognition. (2005) 886-893
- Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2008 Results. <http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html>
- Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object Detection with Discriminatively Trained Part-Based Models. IEEE Transaction on Pattern Analysis and Machine Intelligence **32** (2010) 1627-1645
- Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. IEEE Transaction on Pattern Analysis and Machine Intelligence **14** (2008) 36-51

12. Fritz, M., Schiele, B.: Decomposition, discovery and detection of visual categories using topic models. *IEEE Conference on Computer Vision and Pattern Recognition*. (2008) 1-8
13. Fulkerson, B., Vedaldi, A., Soatto, S.: Class Segmentation and Object Localization with Superpixel Neighborhoods. *IEEE Conference on Computer Vision and Pattern Recognition*. (2009) 670-677
14. Jiang Y., Meng, J., Yuan, J.: Grid-based Local Feature Bundling for Efficient Object Search. *IEEE International Conference and Image Processing*. (2011) 113-116
15. Jiang Y., Meng, J., Yuan, J.: Randomized Visual Phrases for Object Search. *IEEE Conference on Computer Vision and Pattern Recognition*. (2012) 3100-3107
16. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Efficient Subwindow Search: A Branch and Bound Framework for Object Localization. *IEEE Transaction on Pattern Analysis and Machine Intelligence* **31** (2009) 2129-2142
17. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision* **77** (2008) 259-289
18. Opelt, A., Pinz, A., Zisserman, A.: A boundary-fragment-model for object detection. *European Conference on Computer Vision*. (2006) 575-588
19. Parkhi, O.M., Vedaldi, A., Jawahar, C.V., Zisserman, A.: The truth about cats and dogs. *IEEE International Conference on Computer Vision*. (2011) 6-13
20. Ramanan, D.: Using segmentation to verify object hypotheses. *IEEE Conference on Computer Vision and Pattern Recognition*. (2007) 18-23
21. Razavi, N., Gall, J., Van Gool, L.: Scalable Multi-class Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition*. (2011) 1505-1512
22. Rihan, J., Kohli, P., Torr, P.H.S.: OBJCUT for face detection. *Computer Vision, Graphics and Image Processing*. (2006) 13-16
23. Russakovsky, O., Ng, A.Y.: A Steiner tree approach to efficient object detection. *IEEE Conference on Computer Vision and Pattern Recognition*. (2010) 1070-1077
24. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple Kernels for Object Detection. *IEEE International Conference on Computer Vision*. (2009) 606-613
25. Vijayanarasimhan, S., Grauman, K.: Efficient Region Search for Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition*. (2011) 1401-1408
26. Yeh, T., Lee, J.J., Darrell, T.: Fast Concurrent Object Localization and Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*. (2009) 280-287
27. Zhang, Z., Cao, Y., Salvi, D., Oliver, K., Waggoner, J., Wang, S.: Free-Shape Subwindow Search for Object Localization. *IEEE Conference on Computer Vision and Pattern Recognition*. (2010) 1086-1093
28. Zhao, L., Davis, L.S.: Closely coupled object detection and segmentation. *IEEE International Conference on Computer Vision*. (2005) 454-461