Segmenting natural images with the least effort as humans

Qiyang Zhao
zhaoqy@buaa.edu.cn

NLSDE, Beihang University
Beijing, China

Abstract

Great approaches to natural image segmentation have been made in recent years by learning from human segmentations, however little attention is paid to the behavior of human subjects in segmenting images. The paper investigates the effort made by human subjects and proposes an empirical method to estimate the boundary tracing loads, then establishes a model for natural image segmentation based on *the least effort principle*. We sort the hierarchies exhibited in human segmentation processes which use the BSDS tool, together with the monotonicity observed in the region merging processes, into two constraints on our model. Then an algorithm is established to segment natural images from scratch with pretty high efficiency thanks to the monotonic merging strategy.

The experiment on BSDS500 shows our method obtains the state-of-the-art performance on both boundary and region measures. The average time consumption is only 1s and far less than those of its competitors. We also propose a new integrating evaluation measure, on which the performance of our method is noticeably worse than that of human subjects, indicating it is still a long run to build a perfect segmentation method.

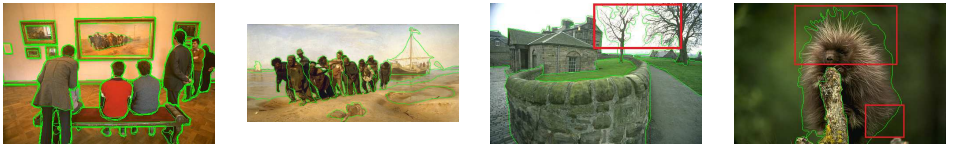
1 Introduction

Given that natural image segmentation is well-known as an ill-posed problem, then how can we design an algorithm to obtain good performance as human subjects? A choice is to learn from human segmentations, as in [5][20][11][12][24]. Amongst them, MCG [12] obtains the state-of-the-art performance and fairly high computational efficiency in all segmentation methods, indicating the necessity of learning.

Then a question arises here: what should we learn? All literatures learning from human segmentations focus on either local cues of contours [20][5], or global cues for good segmentations [4][24]. They predict contours from hand-tuned [5][20] or auto-constructed features [21] by classifiers including the emerging structured forests [13][14][17], then compute segmentations in the way independent of that of human subjects, such as with the ucm method [10].

The way they exploit human segmentations ignores a basic fact that human segmentations are produced by human operations. As a kind of human activities, the human segmentation process would inevitably exhibit the universal characteristics of human behaviors. In 1940's, Zipf published his famous study, **the least effort principle (LEP)**, stating that a human *will strive to solve his problem in such a way as to minimize the total work that he must expend* [6]. It has been widely investigated in from evolutionary biology to web-page design, however is seldom mentioned in the computer vision researches.

Pages 110.1-110.12
DOI: https://dx.doi.org/10.5244/C.29.110



(a) (b) (c) (d)

Figure 1: Images and human-annotated boundaries. (a), (c) and (d) are all from the BSDS. The painting in (a) are an entire image in (b) itself. All boundaries are shown in green. In (c) and (d), the mis-located boundaries are bounded by red boxes. Zoom in for the better view.

The principle gives us a new insight into our problem. There is a small painting at the up-center in fig.1.a, and only one human subject makes little effort to partition the painting. However, the case is totally different if the painting itself forms an entire image, as shown in fig.1.b. Why? The only difference is between their sizes, but the explanation, only large regions deserve refined segmentations, is hardly convincing. From the viewpoint of LEP, the partitions of the embedded copy would be refined only if human subjects under-evaluate their effort of tracing boundaries. With the roughly equal effort, the segmentation of fig.1.a are as refined as in fig.1.b, but the boundaries inside the embedded copy will be far less than those in the entire image. Consider another example in fig.1c-d, the boundaries in red boxes are not accurate at all: none parts of the sky/grass should be put into the tree/hair regions. Notice that these mistaken boundaries do not tamper the segmentation quality severely, but they are much easier to trace, so human subjects choose them to save the tracing loads.

In Sec.2, we model the human segmentation processes in light of the LEP. Besides this, we find the segmentation hierarchy is an explicit characteristic of human operations with the BSDS tools [7]: human subjects are used to refine segmentations in a level-by-level style. We argue, given human subjects can obtain perfect segmentations with the help/constraint of BSDS tools, it should be possible to design a good algorithm by simulating the processes. Therefore we take the hierarchy as one constraint on our model in Sec.3.1. Meanwhile, it should be clarified we are **not** ambitious to claim hierarchies are adherent to human segmentation operations in all scenarios rather than the BSDS experiments.

The popular method for hierarchical segmentation is merging regions based on search trees [10], but its efficiency is low on large images. There is an interesting observation we can exploit: most merging unit costs are monotonically increased during the region merging processes. Then in Sec.3.2 we use the monotonic region merging strategy to improve the computational efficiency promisingly. Finally in Sec.4, we propose a new evaluation measure by integrating all existing popular ones and evaluate our new algorithm on all measures.

2 Human Effort of Segmenting Images

Suppose you are a human subject in the BSDS segmentation experiment, and are required to segment images into pieces under the **ending instruction**: *each piece contains only one single distinguished thing* [4]. Then what effort you should make? Easy to see, the major effort F consists of understanding images (U) to guide following operations, and tracing boundaries (T) by hands with a mouse:

$$F(I,S) = U(I) + T(S) \tag{1}$$

where I is an image, $S \in \{0,1\}^{|E|}$ is a segmentation configuration defined on the set E of all pairs of adjacent pixels (1—two adjacent pixels in the same region, 0—otherwise). Then from the viewpoint of LEP, human subjects would like to find an acceptable segmentation S by minimizing their effort, or formally,

$$\min_S F(I,S), \text{s.t. each piece contains only one thing.} \quad (2)$$

2.1 Formalizing the Ending Instruction

However, as the BSDS document says, the question whether or not a piece contains only one thing, is rather vague and up to subjects themselves [4]. Nevertheless, if you can describe a region easily such as 'a uniformly blue region', then there tend to be only one 'thing'. On the contrary, if you have to describe it with much effort such as 'the left part is uniformly blue while the right part is white and black squares', there will likely be multiple 'things'. In other words, the less effort to describe with, the more likely a region is of 'only one thing'.

We can describe a region with features/concepts in different levels, such as colors, textures and semantic concepts. In this paper, we use the low-level features only, and leave high level concepts to our future work. There are two types of low-level descriptions we should consider. The first is to describe pixels independently, and its effort can be measured by entropies as in [16][22][18]. The second, a 'regularity description' as we call it, is usually adopted by humans but seldom mentioned in literatures. Consider a piece of sky for which a humans says 'varying from dark smoothly to light blue', it is hard to describe pixels independently as there are numerous different colors. Here humans exploit the regularity of the perceptual differences between adjacent pixels. The regularity description effort can be measured by color distances or contour probabilities. We add the two types of description effort up as the total effort to describe a region.

But, is the description effort the only factor we should take into account when deciding whether to partition a region or not? Consider four typical cases:

- the evidence of multiple things is **clear**, and you can trace the boundaries **easily**;
- the evidence of multiple things is **clear**, whereas you have to make **great** effort to trace the boundaries;
- the evidence of multiple things is **vague**, meanwhile the potential boundaries are too **complicated** to trace;
- the evidence of multiple things is **vague**, meanwhile you can trace the potential boundaries with **moderate** effort;

In the first two cases, splitting the region seems the only sound choice. In the third case, it is more sensible to keep the region as a whole, because less effort is made while the instruction is not violated. The last case is much tricky because we must balance the description effort and the tracing effort: we usually choose one of the less effort between 'splitting it' and 'keeping it'. In the following sections, we call a group of regions $\{R_i\}$ is a partition configuration of some region R, if $\cup_i R_i = R$ and $R_i \cap R_j = \phi$ for all i, j's. We call a region R is *RETAINABLE* if it does not need to be partitioned, or formally,

$$RETAINABLE(R) = TRUE \iff \forall \text{ partition } \{R_i\} \text{ of } R, D(R) \leq \sum D(R_i) + T(\{R_i\}) \quad (3)$$

where $D(\cdot)$ stands for the region description effort and $T(\cdot)$ stands for the tracing effort. Then we can sort the ending instruction into the following constraint:

Ending Constraint: We can end a segmentation process with a configuration S, if and only if each region R induced by S is *RETAINABLE*.

2.2 Estimating Tracing Loads

First we clarify the tracing effort T in (3) should be the product of a subjective unit cost (λ) and the tracing load (L), as we find the loads for ordinary people to trace the identical curves are roughly equal, but their subjective evaluations are considerably different. We use 'trace' instead of 'draw' here because the boundaries are 'already' there partitioning distinguished 'things', and all we need is to figure it out. We estimate the tracing loads in an empirical way. We collect several human subjects to trace the boundaries in the BSDS human segmentations, and tell them to do it as precisely as possible. The tracing style is to use many small clicks as recommended by the BSDS tools [7]. We record the time consumptions as the measure of tracing loads, then analyze boundaries to find a way to predict them.

In our experiment, 870 boundaries are traced and their lengths vary from 20 to 2000 pixels. We find that, for those longer than 150, the time consumptions are larger if there are more and/or sharper corners. Dividing 0 to π into 15 bins (the two bins including 0 and π are 1.5 times large as the others), we find the tracing loads are linearly related to the corner amounts of different bins. The following linear regression is subject to two facts: (1) the tracing load of several boundaries should not be less than that of their union, and (2) the sharper corners are harder to trace. The relative square error in our regression is about 0.07, indicating a significant linearity. The normalized coefficients are show in table 1.

Table 1: Normalized coefficients of corner amounts. w_i is the coefficient for the ith bin, w_e is the weight of endpoints. w_0 is the intercept. The values in parentheses are typical angles.

$w_1(0)$	$w_2(\frac{\pi}{8})$	$w_3(\frac{3\pi}{16})$	$w_4(\frac{\pi}{4})$	$w_5(\frac{5\pi}{16})$	$w_6(\frac{3\pi}{8})$	$w_7(\frac{7\pi}{16})$	$w_8(\frac{\pi}{2})$	$w_9(\frac{9\pi}{16})$
1.0000	12.5677	25.0774	25.0774	25.0774	25.0774	28.5677	28.5677	28.5677
$w_{10}(\frac{5\pi}{8})$	$w_{11}(\frac{11\pi}{16})$	$w_{12}(\frac{3\pi}{4})$	$w_{13}(\frac{13\pi}{16})$	$w_{14}(\frac{7\pi}{8})$	$w_{15}(\pi)$	w_e	w_0	
28.5677	28.5677	45.7290	45.7290	45.7290	45.7290	38.1871	30.6387	

The case of short boundaries (especially shorter than 50) is different, because we find the regression coefficients of all corners are roughly equal. The reason is, the sharper corners in short boundaries are not as evident as in long ones, and humans can hardly take the advantage of moving inertia of the mouse to trace short smooth curves. We mix these two cases softly with a Logistic function, to get a unified model to predict the load of tracing a curve ω:

$$L(\omega) = \text{length}(\omega) + \frac{\sum_{i=2}^{15} n_i \cdot (w_i - 1)}{1 + e^{-\alpha \cdot (\text{length}(\omega) - \beta)}} + [\omega \text{ is not closed }] \cdot 2 \cdot w_e + w_0 \qquad (4)$$

where $\alpha = 0.1, \beta = 150$, n_i is the corner amount in the ith bin.

We find corners by estimating the Digital Straight Segments (DSS) on boundaries [19], but in a slightly different way. We calculate the angles of the corners of any two incident DSS's, and put them into corresponding bins. Please see more details in our source codes.

The term U is hard to tackle, because analyzing it with either the psychology or the neural science is far beyond the scope of the paper. Instead, we simply set it as a constant value. Therefore U is irrelevant when minimizing the total effort F. In the following sections, we also let $L(S)$ to represent the load to trace all boundaries induced by a segmentation S, and $L(\{R_i\})$ to represent the load to trace all boundaries between any two regions in a region

group $\{R_i\}$. Then the minimization of the effort F_λ on a unit tracing cost λ is equivalent to

$$\min_S \lambda L(S), \text{s.t. } \forall R \text{ induced by } S, RETAINABLE(R) = TRUE. \quad (5)$$

3 Least Effort-based Segmentation

3.1 Hierarchy Constraint

The BSDS human segmentations are produced hierarchically with the BSDS tool. Easy to see, each time human subjects deciding to partition the image further, they would like to take more tracing loads, or equivalently, their subjective evaluation on the unit cost of tracing loads is reduced. It means the segmentations on low unit costs are produced by refining those on higher unit costs. We sort it into the following constraint on F:

Hierarchy Constraint: If two pixels are separated in the optimal segmentation on an unit tracing cost λ, they will also be separated in the optimal segmentations on costs which are smaller than λ.

Or formally, suppose S_1, S_2 are the optimal solutions to (2) on the unit tracing costs λ_1, λ_2 respectively $(\lambda_1 \leq \lambda_2)$, then we have $S_1 \leq S_2$. The last inequality is defined on vectors: we call $S_1 = \langle s_i^{(1)} \rangle \leq S_2 = \langle s_i^{(2)} \rangle$ if and only if $s_i^{(1)} \leq s_i^{(2)}$ for all i's. Immediately we have:

Lemma 3.1. *For any $\lambda_1 < \lambda_2$, let $S_1 = \arg\min F_{\lambda_1}, S_2 = \arg\min F_{\lambda_2}$. If $S_1 \neq S_2$, and δ^* is the unique solution to $\min_\delta \frac{D(S_1+\delta)-D(S_1)}{L(\delta)}$, then $S_1 + \delta^* \leq S_2$.*

Proof. Let $\lambda_0 = \min_\delta \frac{D(S_1+\delta)-D(S_1)}{L(\delta)}$, then we have $\lambda_1 \leq \lambda_0$ due to the optimality of S_1. Next we will prove $\lambda_0 \leq \lambda_2$. Otherwise, since $S_1 < S_2$, there must exists a region R and its partition $\{R_i\}$ satisfying $\frac{D(R)-\sum_i D(R_i)}{\sum_{i,j} L(R_i,R_j)} \leq \lambda_2 < \lambda_0$. It contradicts the minimality of λ_0, so $\lambda_1 \leq \lambda_0 \leq \lambda_2$.

Let $S_0 = \arg\min F_{\lambda_0}$, then we have $S_1 + \delta^* \leq S_0$. Otherwise it contradicts the minimality and uniqueness of δ^*. Since we have $S_1 \leq S_0 \leq S_2$ according to the hierarchy constraints of F, then $S_1 \leq S_1 + \delta^* \leq S_2$。 □

We denote the value $\frac{D(\bigcup R_i)-\sum D(R_i)}{\sum_{i,j} L(R_i,R_j)}$ by $muc(\{R_i\})$, the unit cost on which the region group $\{R_i\}$ merge into one big region. Suppose we have an optimal solution on the unit cost λ, then based on Lemma 3.1, the optimal solutions to larger unit costs can be found by merging the region group of the minimum *muc*. However, the amount of all region groups is too huge to be handled. A feasible way is to consider adjacent region pairs only, so to optimize F approximately. Here's the outline in Algo. 1, where $\Omega(\cdot,\cdot)$ in step 5 represents the boundary between two regions.

We can demonstrate the ucm-based method with the above framework. If we adopt only the regularity descriptions, and estimate tracing loads only by boundary lengths, then we would have the merging criterion of averaged contour probabilities in [10][11][2][24][12].

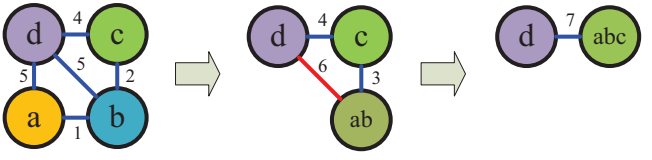
3.2 Monotonic merging

The key idea of Algo. 1 is to maintain a dynamically sorted search tree *BST*. After each mergence, all adjacencies should be updated and re-arranged in *BST*. However, by profiling we find about 80% updates are wasted as they are overwritten by new updates. For example, we have a region R_d adjacent to three regions R_a, R_b and R_c, and the merging sequence are

Algorithm 1: Segmentation based on Naive Merging

Input: Image I, threshold λ_{thr} of the unit cost;
Output: The segmentation result S.

$S \leftarrow \{0\}^{|E|}$; construct an empty search tree BST;
calculate $\langle muc(R_i,R_j),R_i,R_j\rangle$ for all adjacent pixel p_i,p_j and insert it into BST;
$\langle \lambda_{min},R_1,R_2\rangle \leftarrow \min_{\langle\lambda,\cdot,\cdot\rangle\in BST}\lambda$;
while $\lambda_{min} \le \lambda_{thr}$ **do**
  $R \leftarrow R_1 \cup R_2, S \leftarrow S+\Omega(R_1,R_2)$; check new corners and update tracing loads;
  foreach R' *incident to* R **do**
    calculate and insert $\langle muc(R,R'),R,R'\rangle$ into BST, then sort the tree;
  end
  $\langle \lambda_{min},R_1,R_2\rangle \leftarrow \min_{\langle\lambda,\cdot,\cdot\rangle\in BST}\lambda$;
end



(a) (b) (c)

Figure 2: Wasted updates in the toy example. The red line represents the adjacency of *ab* and *d*, for which the *muc* is calculated but never used. The numbers on lines are the *muc*'s.

R_a–R_b and R_{ab}–R_c. The updates involving R_d are corresponding to the potential mergences R_{ab}–R_d and R_{abc}–R_d, while the update for R_{ab}–R_d will be overwritten by that of R_{abc}–R_d, as shown in fig. 2. These wasted updates bring large burdens of calculations. Fortunately, we find almost all mergences are subject to the following constraint:

Monotonicity Constraint: For any three regions R_a,R_b,R_c, let $\lambda_1,\lambda_2,\lambda_3$ be the *muc*'s of three mergences R_a–R_c, R_b–R_c and R_{ab}–R_c respectively, it holds $\min(\lambda_1,\lambda_2) \le \lambda_3$.

In a word, the merging unit costs are monotonically increased. For example in fig.2, the *muc* of merging R_{ab} and R_d is larger than the smallest of $muc(R_a,R_d)$ and $muc(R_b,R_d)$. So it is unnecessary to calculate $muc(R_{ab},R_d)$ before R_a,R_b,R_c merge into R_{abc}. Then we have:

Lemma 3.2. *If the merging unit costs are subject to the monotonicity constraint, then the output of algorithm 1 is the exact solution to (5). Furthermore, the muc's calculated in the step 7 are monotonically increased.*

Proof. For the first part, it is sufficient to prove, for any group of regions, there must exist two regions for which the *muc* is not larger than that of the whole group. Otherwise, we repeatedly choose a pair of regions of the minimum *muc* to merge, until only one region is left. We denote the ith *muc* by λ_i, and the corresponding tracing load by L_i. Since the boundaries of the region group are the union of all boundaries between any two regions, we have the following contradiction

$$D(R) = \sum_i D(R_i) + \sum_i \lambda_i \cdot L_i > \sum_i D(R_i) + \lambda \cdot \sum_i L_i = D(R) \tag{6}$$

The proof of the second part is straightforward. If the new mergence is not related to the last mergence, then according to the minimality of the last *muc*, the new *muc* will be not less than the last one. Otherwise, it would also hold according to the monotonicity constraint. □

In light of the monotonicity, we can replace the search tree *BST* in algorithm 1 with a direct access table [1], *DAT*, to improve the computational efficiency. We divide the range of the merging unit costs uniformly into lots of bins, each of which is corresponding to an entry in *DAT*. On each entry, we store all region pairs of the corresponding merging unit costs. Based on Lemma 3.2, all we need is to check the table entries sequentially and merge those *ACTIVE* and *UPDATED* region pairs. When inserting a new record, we can locate the entry directly according to its merging unit cost. Here's the outline in Algo .2.

Algorithm 2: Segmentation based on Monotonic Merging

Input: Image I, threshold λ_{thr} of the unit cost;
Output: The optimal segmentation result S.

$S \leftarrow \{0\}^{|E|}$; construct an empty direct access table *DAT*;
calculate $\langle muc(p_i,p_j),p_i,p_j\rangle$ for all adjacent pixels p_i,p_j;
insert these records into *DAT*, and label them as *UPDATED* and *ACTIVE*;
for $\lambda \leftarrow 0$ *to* λ_{thr} **do**
    extracts the record $\langle \lambda,R_i,R_j\rangle$ from *DAT*;
    if *its labels are UPDATED and ACTIVE* **then**
        $R \leftarrow R_i \cup R_j, S \leftarrow S+\Omega(R_i,R_j)$; check new corners and update tracing loads;
        foreach *region* R_k *adjacent to* R_i *and* R_j **do**
            pick the record of the smallest *muc* from $\langle muc(R_i,R_k),R_i,R_k\rangle$ and $\langle muc(R_j,R_k),R_j,R_k\rangle$;
            label it as *OUTDATED*, and the other as *INACTIVE*;
        end
    end
    else
        if *its labels are OUTDATED and ACTIVE* **then**
            calculate and insert $\langle muc(R_i,R_j),R_i,R_j\rangle$ into *DAT*; label it as *UPDATED*;
        end
    end
end

Once an adjacency record is labeled as *OUTDATED* but *ACTIVE*, then it is valid but its merging unit cost should be updated in future. Once a record is labeled as *INACTIVE*, it will be skipped when accessing *DAT* as the record is redundant. Since the merging unit costs are increased monotonically, we do not have to update these records right after any merging operations (step 7 in Algo. 2). Thus most wasted updates in Algo. 1 will be saved by this 'postponed updating' strategy. In our experiments on the BSDS, the computational efficiency is improved to be ten times high as that of naive merging.

4 Evaluation

4.1 Revisit on existing measures

There are several popular measures to evaluate the segmentation algorithms independently: Boundary [10], Cover [10], PRI [10], VOI [10] and the recent F_{op} [8].However in existing literatures, the best performances of a single method on these measures are usually reported on dramatically different thresholds. Here we attempt to integrate the first four measures to get a more objective measure, which can reflect both boundary and region performance synchronously.

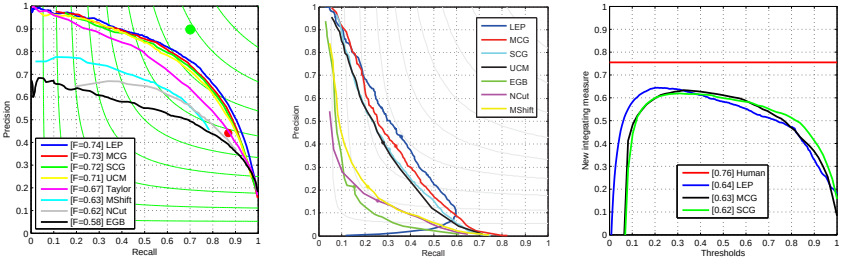
On each image, we let the human segmentations be positive samples, whereas random segmentations be negative ones. We collect the measure scores, then train linear SVMs on d-

ifferent measure combinations to distinguish human segmentations from random ones. Some interesting observations can be done on the coefficients in table 2. The boundary measure is the most discriminative one, as stated in [8]. The coefficient of PRI in the combination B+R is negative, indicating it is partially 'covered' by the boundary measure. Finally, we choose the combination of all measures (B+C+R+V in table 2) as our integrating measure.

Table 2: Coefficients of all existing measures in different combinations. In each combination, all coefficients are normalized by the largest positive coefficient.

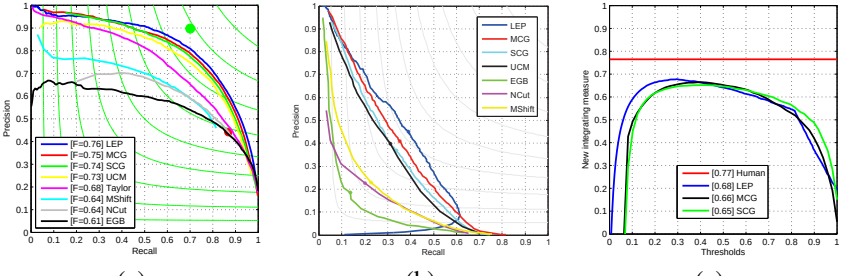
Combination	Boundary(B)	Covering(C)	PRI(R)	VOI(V)	Accuracy(%)
B+C+R+V	1.0000	-0.0045	0.0539	-0.0677	99.92
B+C	1.0000	0.4535	-	-	99.77
B+R	1.0000	-	-0.0463	-	99.28
B+V	1.0000	-	-	-0.0722	99.90
C+R	-	1.0000	0.6822	-	98.33
C+V	-	1.0000	-	-0.0251	96.96
R+V	-	-	1.0000	-0.1205	98.67

4.2 Evaluation Results



(a) (b) (c)

Figure 3: Results on the BSDS500 val set. (a) boundary, (b) F_{op}, (c) the new measure.



(a) (b) (c)

Figure 4: Results on the BSDS500 test set. (a) boundary, (b) F_{op}, (c) the new measure.

We train our models on the BSDS train set and evaluate the new algorithm (LEP in figures and tables) on the test set and val set. We calculate the color entropy in the Lab space and the texture entropy based on a variant of C-LBP [23], to measure the independent descriptions. For the regularity description of a region, we calculate two values on all adjacent pixels: the first is their normalized Euclidean Lab- distances, and the second are the odds being separated in the structured predictions of a random forest [13][14] [1].

[1] All source codes and evaluation results are available on http://gait.buaa.edu.cn/~zqy/lep

Our competitors include eight segmentation methods, MCG [12], SCG [12], UCM [1], Taylor [2], MShift [3], NCut [9], ISCRA[24], EGB [15], and a contour detector SF [13][14]. We evaluate them with the public source codes or their pre-computed results. The evaluation measures include Boundary, Cover, PRI, VOI, F_{op} and our new measure in Sec.4.1. The experiment on BSDS500 shows that our method obtains the state-of-the-art performance, as shown in fig.3, fig.4, fig.5, table 3, and table 4. Especially on its test set, our method is the only one with the precision 0.90+ on the recall 0.50 (see fig.4.a). It runs on all images in less than 1s in average, and is much more efficient than other method which has 0.70+ scores on the boundary measure. Our method outperforms others again on the new measure, however is noticeably worse than that of human subjects (see fig.3.c and fig.4.c).

Table 3: Results on the BSDS500 val set. The best values are shown in bold.

	Boundary			F_{op}		Covering		PRI		VOI		Time
	ODS	OIS	AP	ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS	
Human	0.79	0.79	-	-	-	0.73	0.73	0.87	0.87	1.16	1.16	-
LEP(Ours)	**0.74**	**0.76**	**0.79**	**0.39**	**0.46**	**0.62**	**0.68**	**0.82**	**0.86**	**1.51**	**1.31**	1s
MCG[12]	0.73	0.76	0.74	0.36	0.42	0.61	0.67	0.81	0.86	1.55	1.37	20s+
SCG[12]	0.72	0.73	0.63	0.34	0.40	0.59	0.65	0.80	0.85	1.62	1.44	3.5s+
ISCRA[24]	-	-	-	-	-	0.60	0.66	0.81	0.86	1.62	1.40	240s+
UCM[1]	0.71	0.74	0.73	0.33	0.38	0.59	0.65	0.81	0.85	1.65	1.47	240s
Taylor[2]	0.67	0.72	0.73	-	-	0.56	0.62	0.79	0.84	1.74	1.53	11s
MShift[3]	0.63	0.66	0.54	0.21	0.28	0.54	0.58	0.78	0.80	1.83	1.63	600s+
NCuts[9]	0.62	0.66	0.43	0.19	0.27	0.44	0.53	0.75	0.79	2.18	1.84	600s+
EGB[15]	0.58	0.62	0.53	0.18	0.25	0.51	0.58	0.77	0.82	2.15	1.79	<1s
SF[13][14]	0.73	0.75	0.77	-	-	-	-	-	-	-	-	0.4s

Table 4: Results on the BSDS500 test set. The best values are shown in bold.

	Boundary			F_{op}		Covering		PRI		VOI		Time
	ODS	OIS	AP	ODS	OIS	ODS	OIS	ODS	OIS	ODS	OIS	
Human	0.80	0.80	-	0.56	0.56	0.72	0.72	0.88	0.88	1.17	1.17	-
LEP(Ours)	**0.76**	**0.79**	**0.82**	**0.42**	**0.47**	**0.63**	**0.69**	**0.84**	**0.87**	**1.47**	**1.29**	1s
MCG[12]	0.75	0.78	0.76	0.38	0.43	0.61	0.66	0.83	0.86	1.57	1.39	20s+
SCG[12]	0.74	0.77	0.65	0.35	0.42	0.60	0.65	0.83	0.86	1.63	1.43	3.5s+
ISCRA[24]	0.72	0.75	0.46	-	-	0.59	0.66	0.82	0.85	1.60	1.42	240s+
UCM[1]	0.73	0.76	0.73	0.35	0.38	0.59	0.65	0.83	0.86	1.69	1.48	240s
Taylor[2]	0.68	0.72	0.74	-	-	0.56	0.62	0.81	0.85	1.78	1.56	11s
MShift[3]	0.64	0.68	0.56	0.23	0.29	0.54	0.58	0.79	0.81	1.85	1.64	600s+
NCuts[9]	0.64	0.68	0.45	0.21	0.27	0.45	0.53	0.78	0.80	2.23	1.89	600s+
EGB[15]	0.61	0.64	0.56	0.16	0.24	0.52	0.57	0.80	0.82	2.21	1.87	<1s
SF[13][14]	0.75	0.77	0.80	-	-	-	-	-	-	-	-	0.4s

5 Discussion

The success of our new algorithm shows, when learning human segmentations, it is feasible and necessary to take human behaviors into account to find a good solution. There are lots of human-labeled data in many fields of computer vision, so it should be worthy of investigating related problems from the principles of human behaviors such as LEP, and simulating human behavior to establish new algorithms.

On the other hand, we observe that the performance of our algorithm on the new integrating measure is much worse than that of human subjects. In our opinions, the reason is that human subjects use semantic concepts unconsciously when segmenting images. They are more inclined to merge some perceptually different regions together as the regions are parts of one object. How to introduce semantic concepts into our algorithm is our future work.

Acknowledgement: This work is supported by the State Key Laboratory of Software Development Environment (No. SKLSDE-2015ZX-29).

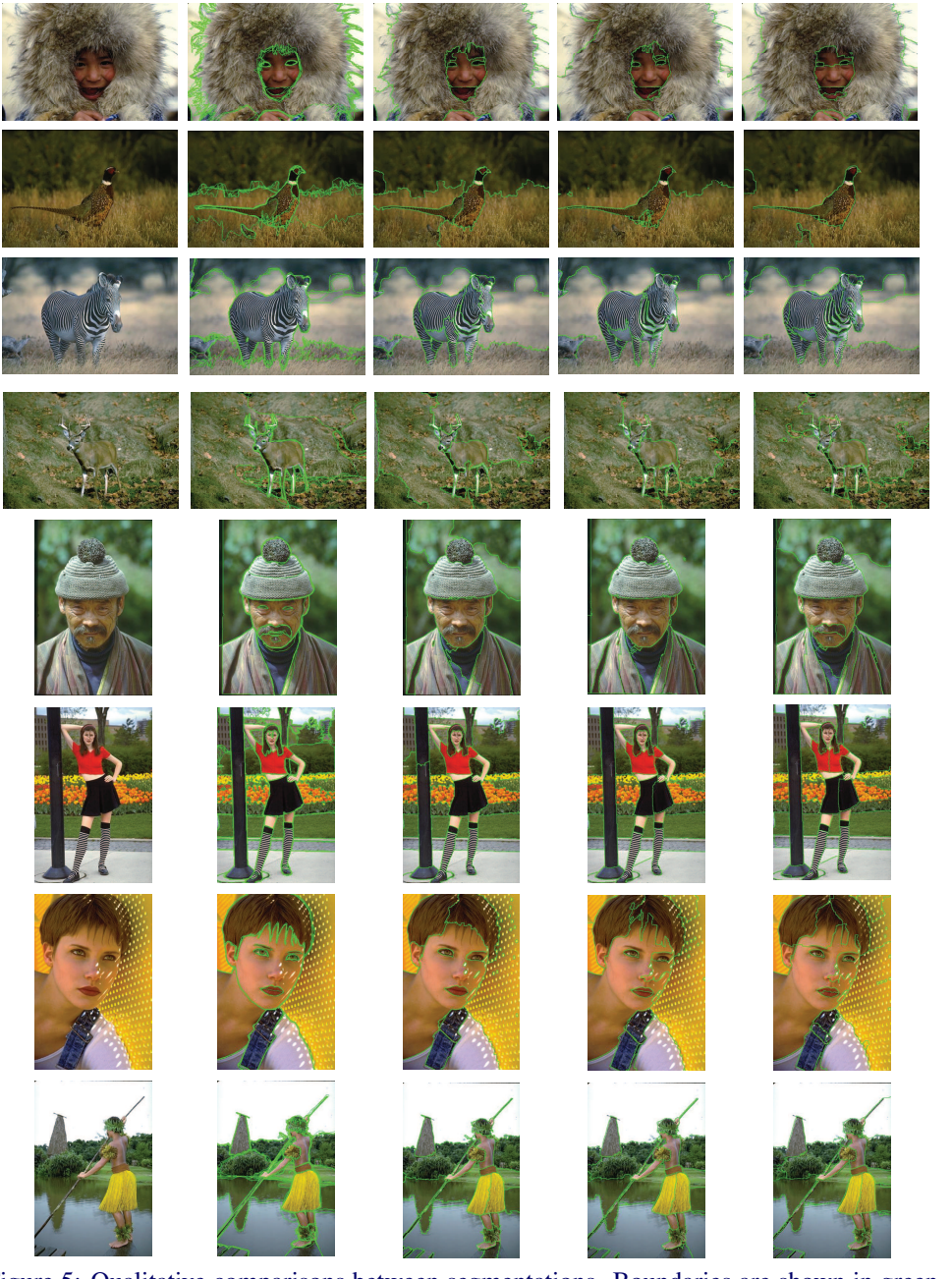


Figure 5: Qualitative comparisons between segmentations. Boundaries are shown in green. From left to right: images, groundtruths, segmentations of our method, MCG and SCG.

References

[1] Stein C., Cormen T., and Rivest R. et al. *Introduction to algorithms*. MIT Press, 2009.

[2] Taylor C. Towards fast and accurate segmentation. In *CVPR*, 2013.

[3] Comaniciu D. and Meer P. Mean shift: a robust approach toward feature space analysis. *IEEE TPAMI*, 2002.

[4] Martin D. and Tal D. et al Fowlkes C. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statics. In *ICCV*, 2001.

[5] Martin D., Fowlkes C., and Malik J. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE TPAMI*, 2004.

[6] Zipf G. *Human behaviour and the principle of least effort*. Addison-Wesley Press, 1949.

[7] Berkeley Vision Group. The berkeley segmentation dataset and benchmark. *http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/*.

[8] Pont-Tuset J. and Marques F. Measures and meta-measures for the supervised evaluation of image segmentation. In *CVPR*, 2013.

[9] Shi J. and Malik J. Normalized cuts and image segmentation. *IEEE TPAMI*, 2000.

[10] Arbelaez P. Boundary extraction in natural images using ultrametric contour maps. In *CVPR Workshop*, 2006.

[11] Arbelaez P., Maire M., and Fowlkes C. et al. Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 2011.

[12] Arbelaez P., Pont-Tuset J., and Barron J. et al. Multiscale combinatorial grouping. In *CVPR*, 2014.

[13] Dollar P. and Zitnick C. Structured forests for fast edge detection. In *ICCV*, 2013.

[14] Dollar P. and Zitnick C. Fast edge detection using structured forests. *IEEE TPAMI*, 2015.

[15] Felzenszwalb P. and Huttenlocher D. Efficient graph-based image segmentation. *IJCV*, 2004.

[16] Bagon S., Boiman O., and Irani M. What is a good image segment? a unified approach to segment extraction. In *ECCV*, 2008.

[17] Hallman S. and Fowlkes C. Oriented edge forests for boundary detection. In *CVPR*, 2015.

[18] Rao S., Mobahi H., and Yang A. et al. Natural image segmentation with adaptive texture and boundary encoding. In *ACCV*, 2009.

[19] Kovalevsky V. New definition and fast recognition of digital straight segments and arcs. In *ICPR*, 1990.

[20] Ren X. and Malik J. Learning a classification model for segmentation. In *ICCV*, 2003.

[21] Ren X. and Bo L. Discriminatively trained sparse code gradients for contour detection. In *NIPS*, 2012.

[22] Ma Y., Derksen H., Hong W., and Wright J. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE TPAMI*, 2007.

[23] Guo Z., Zhang L., and Zhang D. A completed modeling of local binary pattern operator for texture classification. *IEEE TIP*, 2010.

[24] Ren Z. and Shakhnarovich G. Image segmentation by cascaded region agglomeration. In *CVPR*, 2013.