

# Automatic Image Segmentation by Dynamic Region Merging

Bo Peng, Lei Zhang<sup>1</sup>, *Member, IEEE* and David Zhang, *Fellow Member, IEEE*

Department of Computing, The Hong Kong Polytechnic University, Hong Kong

**Abstract:** This paper addresses the automatic image segmentation problem in a region merging style. With an initially over-segmented image, in which the many regions (or super-pixels) with homogeneous color are detected, image segmentation is performed by iteratively merging the regions according to a statistical test. There are two essential issues in a region merging algorithm: order of merging and the stopping criterion. In the proposed algorithm, these two issues are solved by a novel predicate, which is defined by the sequential probability ratio test (SPRT) and the minimal cost criterion. Starting from an over-segmented image, neighboring regions are progressively merged if there is an evidence for merging according to this predicate. We show that the merging order follows the principle of dynamic programming. This formulates image segmentation as an inference problem, where the final segmentation is established based on the observed image. We also prove that the produced segmentation satisfies certain global properties. In addition, a faster algorithm is developed to accelerate the region merging process, which maintains a nearest neighbor graph (NNG) in each iteration. Experiments on real natural images are conducted to demonstrate the performance of the proposed dynamic region merging algorithm.

**Key Words:** Image segmentation, Region merging, Wald's SPRT, Dynamic programming

---

<sup>1</sup> Corresponding author: [cszhang@comp.polyu.edu.hk](mailto:cszhang@comp.polyu.edu.hk). This work is supported by the Hong Kong SAR General Research Fund (PolyU 5330/07E).

## I. Introduction

Image segmentation is a fundamental yet still challenging problem in computer vision and image processing. In particular, it is an essential process for many applications such as object recognition, target tracking, content-based image retrieval and medical image processing, etc. Generally speaking, the goal of image segmentation is to partition an image into a certain number of pieces which have coherent features (color, texture, etc.) and in the meanwhile to group the meaningful pieces together for the convenience of perceiving [1]. In many practical applications, as a large number of images are needed to be handled, human interactions involved in the segmentation process should be as less as possible. This makes automatic image segmentation techniques more appealing. Moreover, the success of many high-level segmentation techniques (e.g. class-based object segmentation [2, 3]) also demands sophisticated automatic segmentation techniques.

Dating back over decades, there is a large amount of literature on automatic image segmentation. For example, the edge detection algorithms [4-7] are based on the abrupt changes in image intensity or color, thus salient edges can be detected. However, due to the resulting edges are often discontinuous or over-detected, they can only provide candidates for the object boundaries. Another classical category of segmentation algorithms is based on the similarities among the pixels within a region, namely region-based algorithms. In order to cluster the collection of pixels of an image into meaningful groups of regions or objects, the region homogeneity is used as an important segmentation criterion. Many cut criteria in graph theory have been studied for this purpose. The most widely used cut criteria include normalized cut [8], ratio cut [9], minimum cut [10] and so on. The aim of these algorithms is to produce a desirable segmentation by achieving global optimization of some cost functions. However, these cost functions only provide a characterization of each cut rather than the whole regions. Another problem is that the optimization processes are often computationally inefficient for many practical applications. In recent years, the success of combinatorial graph cut methods [41-42] has been attracting significant research attention. These methods utilize the user input information along with the cut criteria in optimization and nearly global optima can be achieved in linear computational time. As a matter of fact, for most cut-based energy functionals a single optimal partition of an image is not easy to obtain. It makes a possibility of finding different level-based explanations of an image. From this aspect, there are methods [11-12] tackling the image segmentation as a hierarchical bottom-up problem.

In region-based methods, a lot of literature has investigated the use of primitive regions as a preprocessing step for image segmentation [13-15]. The advantages are twofold. First, regions carry on more information in describing the nature of objects. Second, the number of primitive regions is much fewer than that of pixels in an image and thus largely speeds up the region merging process. Starting from a set of primitive regions, the segmentation is conducted by progressively merging the similar neighboring regions according to a certain predicate, such that a certain homogeneity criterion is satisfied. In previous works, there are region merging algorithms based on statistical properties [16, 22-23, 50-51], graph properties [17-19, 24-25] and spatio-temporal similarity [20]. Although the segmentation is obtained by making local decisions, some techniques [16-21] have shown satisfying results with efficient implementation. Most region merging algorithms do not have some desirable global properties, even though some recent works in region merging address the optimization of some global energy terms, such as the number of labels [43] and the area of regions [44]. As a good representation of morphological segmentation, watershed transform [45] can also be classified as region-based segmentation methods. The intuitive idea comes from geography, where watersheds are the dividing lines of different catchment basins. The major drawback of watershed transform is the over-segmentation of the image. To overcome this problem, one solution is “flooding” from the selected markers [46-48] such that only the most important regional minima are saved for the segmentation. The other [49] is based on a hierarchical process, where the catchment basins of the watershed image are merged until they belong to almost homogeneous regions.

In this paper, we implement the segmentation algorithm in a region merging style for its merit of efficiency, where similar neighboring regions are iteratively merged according to a novel merging predicate. As stated above, homogeneity criteria (cues) are essential to the region merging process. Although a good enough cue is needed in order to obtain a valuable segmentation, our work does not focus on finding a more complex region model. Instead, we model the cues by a function of random variables. In this way, the properties of cues are not mainly concerned, but the reliability of the cues. In many traditional segmentation algorithms (e.g. [8-10]), the reliability is predetermined and thus researchers often try to use more reliable cues for implementation. In contrast, some statistical segmentation methods are able to calculate the reliability of cues, for example, using the parametric probability models [50-51]. But they cannot be used in a general scenario. Another statistical method [16] directly uses the statistical property of image data (e.g., colors) to identify the region border. Particularly, a homogeneity criterion is proposed based on the

expectation of pixel colors inside a region, which naturally leads to a merging predicate. In the recent work of region merging [22], a likelihood ratio test is applied as the measure of region similarity. The probability of adjacent regions and that of their merging are both computed. To minimize both probabilities of error, the optimal merging will take place along with the largest decreasing of the likelihood ratio.

In our work, image segmentation is also formulated as an inference problem, and there is no limitation on the distribution of random variables. The evaluation of the homogeneity of regions follows the grouping principle of Gestalt theory [26-27], which suggests a preference to having consistent elements in the same data set. Unlike the method proposed in [22], we do not directly use the ratio test as a similarity measure, but take it as an enhancement for the merging evidence. The proposed predicate can be therefore interpreted as a combination of the consistency measure and the similarity measure. More specifically, the extent of consistency tells whether or not the tested data belong to the same group. It is measured by two hypotheses according to the Sequential Probability Ratio Test (SPRT) [28]: a null hypothesis  $H_0$  “the tested data are consistent” and an alternative hypothesis  $H_1$  “the tested data are inconsistent”. The proposed similarity measure describes the affinity between two neighboring regions. In each iteration, a region finds its closest neighbor at the lowest value according to some cost function. With the consistency measure and the similarity measure, the problem of searching in a spatial-time space becomes finding an optimal path that mostly satisfies the data consistency. If taking the segmentation as a labeling problem [1], the label of a region will transit to the one of its closest neighbor as iteration goes on. As a result, a merging will happen when two neighboring regions exchange their labels. We can prove that with this merging scheme, a primitive region will be merged into its most similar group at the lowest cost in the final segmentation.

A good segmentation algorithm should preserve certain global properties according to the perceptual cues. This leads to another essential problem in a region merging algorithm: the order that is followed to perform the region merging. Since the merging process is inherently local, most existing algorithms have difficulties to possess some global optimality. However, we demonstrate that the proposed algorithm holds certain global properties, i.e., being neither over-merged nor under-merged, using the defined merging predicate. In addition, to speed up the region merging process, we introduce the structure of nearest neighbor graph (NNG) to accelerate the proposed algorithm in searching the merging candidates. The experimental results indicate the efficiency of the acceleration algorithm.

The rest of this paper is organized as follows. In Section II, a novel region merging predicate is defined.

The consistency test of neighboring regions is defined in Section III by using SPRT. In Section IV, we present the dynamic region merging algorithm and prove some global properties it holds for the segmentation. In Section V, an accelerated algorithm for the merging process is presented and analyzed. In Section VI, we show the segmentation results on real images. Section VII concludes the paper.

## II. The Region Merging Predicate

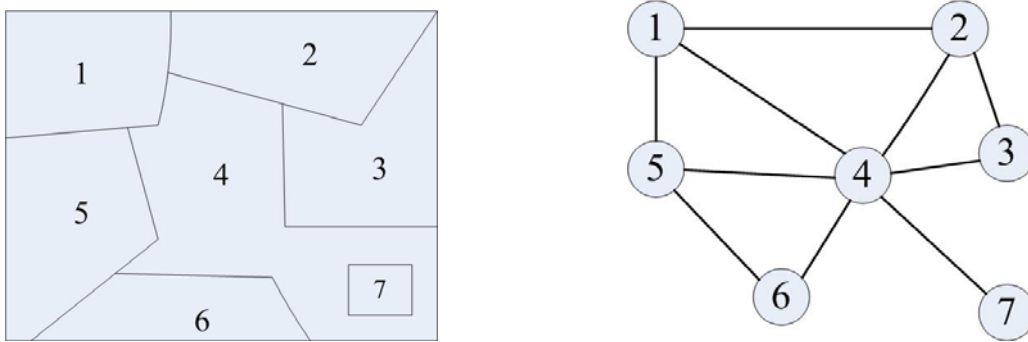
Automatic image segmentation can be phrased as an inference problem [1]. For example, we might observe the colors in an image, which are caused by some unknown principles. In the context of image segmentation, the observation of an image is given but the partition is unknown. In this respect, it is possible to formulate the inference problem as finding some representation of the pixels of an image, such as the label that each pixel is assigned. With these labels, an image is partitioned into a meaningful collection of regions and objects. The Gestalt laws in psychology [26-27] have established some fundamental principles for this inference problem. For example, they imply some well-defined perceptual formulations for image segmentation, such as homogeneous, continuity and similarity. In the family of region merging techniques, some methods have used statistical similarity tests [16, 29] to decide the merging of regions, where a predicate is defined for making local decisions. These are good examples of considering the homogeneity characteristics within a region, from which we can see that an essential attribute for region merging is the consistency of data elements in the same region. In other words, if neighboring regions share a common consistency property, they should belong to the same group. However, most of the existing region merging algorithms cannot guarantee a globally optimal solution of the merging result. As a consequence, the region merging output is over-merged, under-merged or a hybrid case. In this section, we propose a novel predicate which leads to certain global properties for the segmentation result.

The proposed predicate is based on measuring the dissimilarity between pixels along the boundary of two regions. For the convenience of expression, we use the definition of region adjacency graph (RAG) [30] to represent an image. Let  $G = (V, E)$  be an undirected graph, where  $v_i \in V$  is a set of nodes corresponding to image elements (e.g. super-pixels or regions).  $E$  is a set of edges connecting the pairs of neighboring nodes. Each edge  $(v_i, v_j) \in E$  has a corresponding weight  $w((v_i, v_j))$  to measure the dissimilarity of the two nodes connected by that edge. In the context of region merging, a region is represented by a component  $R \subseteq V$ . We

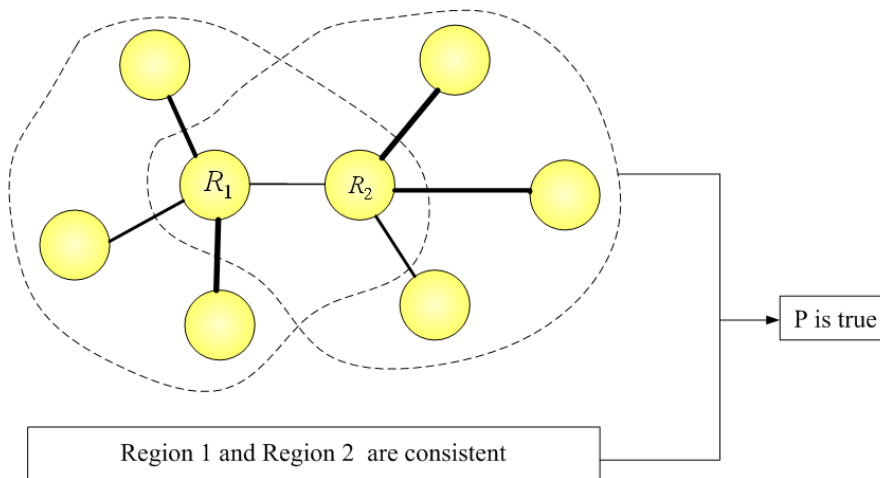
obtain the dissimilarity between two neighboring regions  $R_1, R_2 \subseteq V$  as the minimum weight edge connecting them. That is,

$$S(R_1, R_2) = \min_{v_i \in R_1, v_j \in R_2, (v_i, v_j) \in E} w((v_i, v_j)) \quad (1)$$

The graph structure of an example partition is shown in Fig. 1, where the image has 7 regions and its RAG is shown on the right. The advantage of RAG is that it can provide a “spatial view” of the image.



**Figure 1.** An example of region partition and the corresponding region adjacency graph (RAG).



**Figure 2.** An example that the predicate  $P$  between  $R_1$  and  $R_2$  is true. The thickness of lines indicates the weights of the edges. The most similar pair of regions is connected by an edge with the minimal weight.

Since the merging predicate will decide whether there is an evidence of merging between the pair of regions, it involves two aspects: a dissimilarity measure which is used to determine the candidate region for merging, and the consistency property which checks if the regions are homogenous. We define the following region merging predicate  $P$ :

$$P(R_1, R_2) = \begin{cases} \text{true} & \text{if (a) } S(R_1, R_2) = \min_{R_i \in \Omega_1} S(R_1, R_i) = \min_{R_j \in \Omega_2} S(R_2, R_j); \text{ and} \\ & \text{(b) } R_1 \text{ and } R_2 \text{ are consistent} \\ \text{false} & \text{otherwise} \end{cases} \quad (2)$$

where  $\Omega_1$  and  $\Omega_2$  are the neighborhood sets of  $R_1$  and  $R_2$ , respectively. The merging predicate on regions  $R_1$  and  $R_2$  could thus be “merge  $R_1$  and  $R_2$  if and only if they are the most similar neighbors in each other’s neighborhood and follow the principle of consistency.” The condition (a) is stronger than that of only requiring the connecting edge between  $R_1$  and  $R_2$  to be the minimal one in either of the neighborhood. This leads to an interesting property of the proposed region merging algorithm, i.e., the candidates of the pairs of regions for merging are uniquely decided by the given graph. We shall see hereafter that such a condition uniquely decides the pairs of regions to be merged at a given merging level. Moreover, in Section V we will prove that there is always at least one pair of regions which satisfies condition (a). Clearly, without condition (b), all the regions will be merged into one big region at the end of region merging process. Therefore, condition (b) acts as a stopping criterion. Fig. 2 illustrates an example when the predicate  $P$  between regions  $R_1$  and  $R_2$  is true.

According to the definition of  $P$ , the test of consistency is based on the visual cues extracted from the image data. In the next section, the SPRT method is introduced for a reliable decision on the consistency of regions.

### III. Consistency Test of Cues

In order to obtain the homogenous regions in region merging, the proposed predicate  $P$  in Eq. (2) checks the consistency of regions. Region information is usually presented by the cues extracted from the observed data. The choice of cues can be intensity, color, texture and so on. If we view the cue as a random variable, the distribution of the cue depends on the consistency of pairs of regions. In this paper, we formulate the evaluation of the region consistency as a sequential test process. Suppose parameter  $\theta$  is related to the distribution of random cues  $x$ . More specifically, we gather information of parameter  $\theta$  by observing random variables in successive steps. Since every sample of the cues carries statistical information on parameter  $\theta$ , we may collect the information at the end of observation. This is one of the interesting problems studied in sequential analysis, where  $\theta$  is called a hypothesis. In the context of region merging, two hypotheses are

involved in the evaluation task: a pair of regions is “consistent”, and is “inconsistent”, which are denoted by a null hypothesis  $H_0: \theta = \theta_0$  against an alternative hypothesis  $H_1: \theta = \theta_1$ , respectively. The property of the hypotheses is a hidden state that is not directly observable, but is statistically linked to the observable cues. To decide whether or not a pair of regions belongs to the same group, we look for the solution of its hypothesis test.

An efficient and popular procedure for integrating the statistical evidence is the sequential probability ratio test (SPRT) which was proposed by Wald [28]. SPRT shows that the solution to the hypothesis can be found by making the smallest number of observations and satisfying the bounds on the predefined probabilities of two errors. SPRT is purely sequential in nature, i.e., continuing sampling on the instances of a random variable will eventually lead to a reliable inference about parameter  $\theta$ .

The application of SPRT to the consistency test of cues is described as follows. We observe the distribution of random cues  $x$  in a sequence until a likelihood ratio  $\delta$  goes out of the interval  $(B, A)$  for the first time by a random walk, where the real numbers  $A$  and  $B$  satisfy  $B < 0 < A$ . The sequence of successive likelihood ratio  $\delta_i$  is:

$$\delta_i = \log \frac{P_0(x_i | \theta_0)}{P_1(x_i | \theta_1)}, \quad i = 1, 2, \dots, N \quad (3)$$

where  $P_0(x|\theta_0)$  and  $P_1(x|\theta_1)$  are the distributions of visual cues.  $P_0(x|\theta_0)$  and  $P_1(x|\theta_1)$  should be different so as to make a convincing decision. We use the Gaussian distribution model to approximate the cue distributions. The two conditional probabilities are given as follows:

$$\begin{cases} P_0(x | \theta_0) = \lambda_1 \exp(-(I_b - I_{a+b})^T S_I^{-1} (I_b - I_{a+b})) \\ P_1(x | \theta_1) = 1 - \lambda_2 \exp(-(I_b - I_a)^T S_I^{-1} (I_b - I_a)) \end{cases} \quad (4)$$

where  $I_a$  and  $I_b$  are the average color of sampled data in regions  $a$  and  $b$  respectively, and  $I_{a+b}$  is the average value of samples' union.  $S_I$  is the covariance matrix of the regions, and  $\lambda_1$  and  $\lambda_2$  are scalar parameters. If each test is independent, the composition of the likelihood ratios is the sum of the individual  $\delta_i$ :

$$\delta = \sum_{i=1}^N \delta_i \quad (5)$$

where  $N$  is the first integer for which  $\delta \geq A$  or  $\delta \leq B$ . We can see that the solution to the hypothesis is decided by the relationship between  $\delta$ , an upper limit and a lower limit, denoted by  $A$  and  $B$ , respectively. If  $\delta$  goes out of one of these limits, the hypothesis is made and thus the test stops. Otherwise, the test is carried



on with a new random sampling.

Intuitively, the likelihood ratio  $\delta$  is positive and high when the terminal decision is in favor of  $H_0$ , while it is negative and low when the situation is reversed. In the SPRT theory [28], Wald recommended implementing the test with a practical approximation:  $A=\log(1-\beta)/\alpha$  and  $B=\log\beta(1-\alpha)$ , where  $\alpha$  and  $\beta$  are probabilities of the decision error given by:

$$\begin{aligned}\alpha &= \Pr\{\text{Rejecting } H_1 \text{ when } H_1 \text{ is true}\} \\ \beta &= \Pr\{\text{Accepting } H_1 \text{ when } H_0 \text{ is true}\}\end{aligned}$$

The selection of  $\alpha$  and  $\beta$  affects the region merging quality. Intuitively, as error rates decrease, the region merging quality grows. However, at the same time the computational effort will increase accordingly. In our implementation, both  $\alpha$  and  $\beta$  are set to a fixed value 0.05.

For the uncertainty of the (worst-case) number of tests in SPRT, a truncated SPRT [28] is used here by presetting an upper bound  $N_0$  on the number of tests. In Wald's theory, the expected number of tests is:

$$\begin{cases} E\{n | \theta_0\} = [A(1-\beta) + B\beta] / \eta_0 \\ E\{n | \theta_1\} = [A\alpha + B(1-\alpha)] / \eta_1 \end{cases} \quad (6)$$

where  $\eta_0, \eta_1$  are the conditionally expected number of trails from a single test:  $\eta_0=E\{\delta|\theta_0\}$  and  $\eta_1=E\{\delta|\theta_1\}$ .

We set  $N_0$  to be a constant which is greater than  $\max\{E\{\delta|\theta_0\}, E\{\delta|\theta_1\}\}$ . The proposed SPRT based consistency test of cues is summarized in Table 1.

**Table 1.** Algorithm 1: consistency test of cues

<p><b>Preset</b> <math>\lambda_1</math>;</p> <p><b>Let</b> <math>\lambda_2=1, \alpha=0.05, \beta=0.05</math>;</p> <p><b>Compute parameters:</b></p> <p><math>N_0</math>: be a constant greater than <math>\max\{E\{\delta \theta_0\}, E\{\delta \theta_1\}\}</math>;</p> <p><math>A=\log(1-\beta)/\alpha, B=\log\beta(1-\alpha)</math>;</p> <p><math>P_0(x \theta_0), P_1(x \theta_1)</math> are computed using Eq. (4).</p> <p><b>Input:</b> a pair of neighboring regions.</p> <p><b>Output:</b> the decision <math>D</math> that the two regions are “consistent” (<math>D=1</math>) or “inconsistent” (<math>D=0</math>).</p> <ol style="list-style-type: none"> <li>1. Set evidence accumulator <math>\delta</math> and the trials counter <math>n</math> to be 0.</li> <li>2. Randomly choose <math>m</math> pixels in each of the pair of regions, where <math>m</math> equals the half size of the region.</li> <li>3. Calculate the distributions of visual cues <math>x</math> using Eq. (4) based on these pixels.</li> <li>4. Update the evidence accumulator <math>\delta = \delta + \log \frac{P_0(x_i   \theta_0)}{P_1(x_i   \theta_1)}</math>.</li> <li>5. If <math>n \leq N_0</math> <ul style="list-style-type: none"> <li>If <math>\delta \geq A</math>, return <math>D=1</math> (consistent)</li> <li>If <math>\delta \leq B</math>, return <math>D=0</math> (inconsistent)</li> </ul> </li> </ol>
--

If  $n > N_0$   
     If  $\delta \geq 0$ , return  $D=1$  (consistent)  
     If  $\delta < 0$ , return  $D=0$  (inconsistent)  
 6. Go back to step 2.

## IV. Dynamic Region Merging (DRM)

### IV-A. The dynamic region merging algorithm

In this section, we explain the proposed region merging algorithm as a dynamic region merging (DRM) process, which is proposed to minimize an objective function with the merging predicate  $P$  defined in Eq. (2). As mentioned in Section I, the proposed DRM algorithm is started from a set of over-segmented regions. This is because a small region can provide more stable statistical information than a single pixel, and using regions for merging can improve a lot the computational efficiency. For simplicity and in order to validate the effectiveness of the proposed DRM algorithm, we use the watershed algorithm [31] (with some modification) to obtain the initially over-segmented regions (please refer to Section VI-A for more information), yet using a more sophisticated initial segmentation algorithm (e.g. mean-shift [32]) may lead better final segmentation results.

Given an over-segmented image, there are many regions to be merged for a meaningful segmentation. By taking the region merging as a labeling problem, the goal is to assign each region a label such that regions belong to the same object will have the same label. There are two critical labels for a region  $R_i$ : the initial label  $l_i^0$ , which is decided by the initial segmentation, and the final label  $l_i^n$ , which is assigned to the region when the merging process stops. In our problem, the final label  $l_i^n$  for a given region is not unique, which means that the same initialization  $l_i^0$  could lead to different solutions. This uncertainty mainly comes from the process of SPRT with a given decision error. The test of consistency/inconsistency depends on the error probabilities of the cue decisions  $\alpha$  and  $\beta$ . In general, these decisions are precise for homogenous regions. If a region contains a small part of non-homogenous data, the SPRT might add a few more times of tests to verify its decision. With reasonably small error probabilities, the segmentation results will be more reliable. According to our observation, in most cases, the segmentation result is stable for a given image and it can be guaranteed that all the results satisfy the merging predicate  $P$  defined in Eq. (2). In the process of region merging, the label of each region is sequentially transited from the initial one to the final one, which

is denoted as a sequence  $(l_i^0, l_i^1, \dots, l_i^n)$ .

To find an optimal sequence of merges which produce a union of optimal labeling for all regions, the minimization of a certain objective function  $F$  is required. According to predicate  $P$ , the transition of a region label to another label corresponds to a minimal edge weight that connects the two regions. In this case, a sequence of transitions will be defined on a set of local minimum weights, i.e., in each transition the edge weight between the pair of merged regions should be the minimal one in the neighborhood. As a result, the objective function  $F$  used in this work is defined as the measure of transition costs in the space of partitions. In other words, as the whole image is a union of all regions,  $F$  is the sum of transition costs over all regions. That is:

$$F = \sum_{R_i} F_i \quad (7)$$

where  $F_i$  is the transition costs of one region  $R_i$  in the initial segmentation. Minimizing  $F$  in Eq. (7) is a combinatorial optimization problem and finding its global solution is in general a hard task. Since the exhaustive search in the solution space is impossible, an efficient approximation method is desired. The solution adopted here is based on the stepwise minimization of  $F$ , where the original problem is broken down into several sub-problems by using the dynamic programming (*DP*) technique [33].

The *DP* is widely used to find the (near) optimal solution of many computer vision problems. The principle of *DP* is to solve a problem by studying a collection of sub-problems. Indeed, there have been some works in image segmentation that benefit from this efficient optimization technique, such as *DP* snake [34-35]. In the proposed DRM algorithm, we apply *DP* on discrete regions instead of line segments. The minimization problem for region  $R_i$  starting at labeling  $l_i^0$  is defined as:

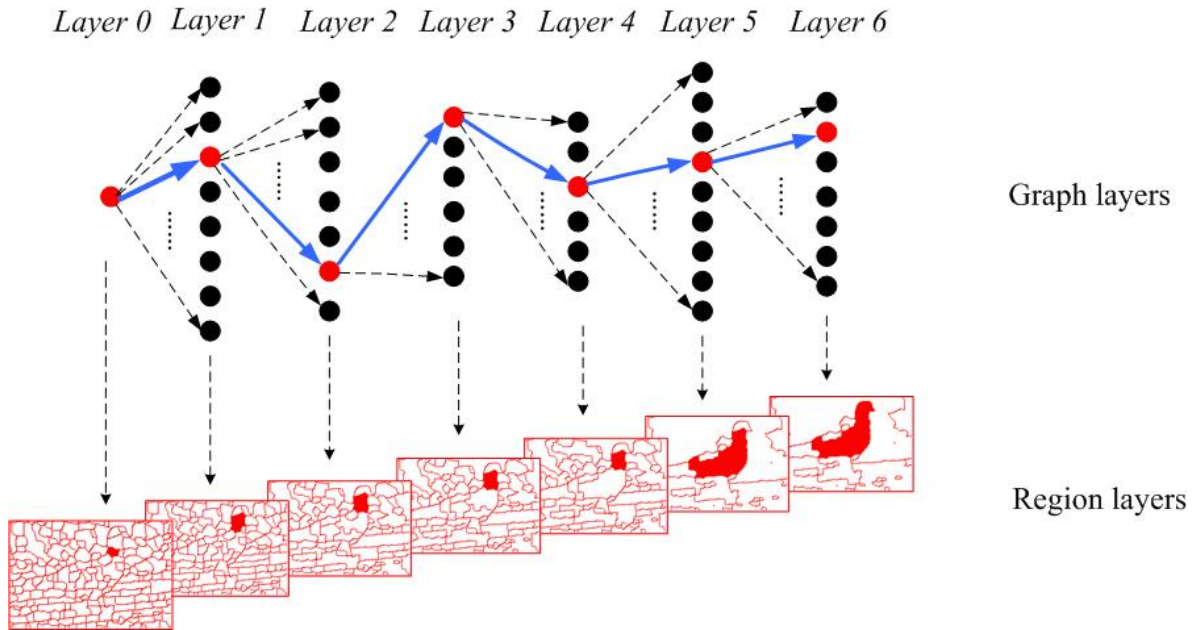
$$\begin{aligned} \min F_i(l_i^0, \dots, l_i^n) &= \min F_i(l_i^0, l_i^{n-1}) + d_{n-1, n} \\ &= \min F_i(l_i^0, l_i^{n-2}) + d_{n-2, n-1} + d_{n-1, n} \\ &= \dots \\ &= \sum_{k=0}^{n-1} d_{k, k+1} \end{aligned} \quad (8)$$

where  $F_i(l_i^0, l_i^1, \dots, l_i^n)$  is the transition cost from  $l_i^0$  to  $l_i^n$ ,  $d_{k, k+1}$  is the minimal edge weight between the regions with labeling  $l_i^k$  and  $l_i^{k+1}$ , respectively. In conjunction with Eq. (1), we have

$$d_{k, k+1} = \min_{R_{k+1} \in \Omega_k} S(R_k, R_{k+1}) \quad (9)$$

The overall path length from  $l_i^0$  to  $l_i^n$  is the sum of minimum edges  $d_{k,k+1}$  for each node in that path. This problem reduces to a search for a shortest path problem, whose solution can be found at a finite set of steps by the Dijkstra's algorithm in polynomial time. At this point, a minimization process of objective function  $F$  is exactly described by the predicate  $P$  defined in Eq. (2), where  $P$  is true if the nodes are connected by the edge with the minimal weight in their neighborhood. It means that the closest neighbors will be assigned to the same label, which is the cause of the merging.

In Fig. 3, an example process of region merging is shown by embedding it into a 3D graph. Between two adjacent layers there is a transition which indicates the costs of a path. Clearly, this is also a process of label transitions. The neighborhood of the highlighted region (in red) is denoted as the black nodes in the graph and the closest neighbor is denoted as the red nodes. The directed connections with the lowest cost between adjacent layers are made (shown as blue arrows). Note that the connectivity between regions in the same layer is represented by the RAG, which is not explicitly shown in Fig. 3.



**Figure 3.** The dynamic region merging process as a shortest path in a layered graph. The upper row shows the label transitions of a graph node. The lower row shows the corresponding image regions of each label layer. Starting from layer 0, the highlighted region (in red) obtains a new label from its closest neighbor (in red). If the region is merged with its neighbor, they will be assigned to the same label. The shortest path is shown as the group of the directed edges (in blue).

The proposed DRM algorithm is summarized in Table 2. Minimization of the objective function follows the principle of  $DP$ , which is exactly expressed in terms of predicate  $P$ . In the  $k$ -th decision step, a merging occurs when two neighboring regions are connected by the minimum weight edge in each other's

neighborhood. With the observation of this characteristic, we will propose an accelerated region merging algorithm in Section V.

**Table 2.** Algorithm 2: segmentation by dynamic region merging

<p><b>Input:</b> the initially over segmented image <math>S_0</math>.</p> <p><b>Output:</b> region merging result.</p>
<ol style="list-style-type: none"> <li>1. Set <math>i=0</math>.</li> <li>2. For each region in segmentation <math>S_i</math>, use <b>Algorithm 1</b> to check the value of predicate <math>P</math> with respect to its neighboring regions.</li> <li>3. Merge the pairs of neighboring regions whose predicate <math>P</math> is true, such that segmentation <math>S_{i+1}</math> is constructed.</li> <li>4. Go back to step 2 until <math>S_{i+1} = S_i</math>.</li> <li>5. Return <math>S_i</math>.</li> </ol>

#### IV-B. Properties of the proposed DRM algorithm

Although the proposed DRM scheme is conducted in a greedy style, some global properties of the segmentation can be obtained. More specifically, it can be proved that the proposed DRM algorithm can produce a segmentation  $S$  which is neither over-merged nor under-merged according to the proposed predicate  $P$ . Similar to the definitions of over-segmentation and under-segmentation in [18], we define the concepts of over-merged segmentation and under-merged segmentation as below.

**Definition 1. Under-merged segmentation.** A segmentation  $S$  is under-merged if it contains some pair of regions for each the predicate  $P$  is true.

**Definition 2. Over-merged segmentation.** A segmentation  $S$  is over-merged, if there is another segmentation  $S_r$  which is not under-merged and each region of  $S_r$  is contained in some component of  $S$ . Or saying,  $S_r$  can be obtained by splitting one or more regions of  $S$ . We call that  $S_r$  is a refinement of  $S$ .

**Definition 3. The evidence of boundary.** There is an evidence of boundary between a pair of regions  $R_1$  and  $R_2$  if the predicate  $P$  is false with condition (a) being satisfied but not condition (b) in Eq. (2).

**Lemma 1.** If two adjacent regions are not merged for an evidence of a boundary between them in the  $k$ -th iteration, they will be in different regions in the final segmentation. Denote by  $R_i^k$  and  $R_j^k$  two neighboring regions in the  $k$ -th iteration. Then  $R_i = R_i^k$  and  $R_j = R_j^k$ , where  $R_i$  is the region whose label is  $L_i$  and  $R_j$  is the region whose label is  $L_j$  in the final segmentation  $S$ .

**Proof:** If there is an evidence of a boundary between  $R_i^k$  and  $R_j^k$ , according to **Definition 3**, we have

$S(R_i, R_j) = \min_{a \in \Omega_i} S(R_i, R_a) = \min_{b \in \Omega_j} S(R_j, R_b)$ , where  $\Omega_i$  and  $\Omega_j$  are the neighborhoods of  $R_i^k$  and  $R_j^k$ , respectively. Since  $R_i^k$  and  $R_j^k$  are not merged, they will not be merged with any other regions in the remaining steps before they are merged with each other. So we have  $R_i = R_i^k$  and  $R_j = R_j^k$ . ■

**Lemma 1** holds because our merging predicate relies on comparing the minimal weight edge between regions. We can prove in the following two theorems that with this measure, some global properties of the segmentation can be easily obtained.

**Theorem 1.** Given the predefined error bounds, the segmentation  $S$  by **Algorithm 2** is not under-merged according to **Definition 1**.

**Proof.** If  $S$  is under-merged, there must be some pair of regions  $R_i^k$  and  $R_j^k$  which have a true value of the predicate, but do not cause a merge in the merging process. Therefore, the evidence of a boundary does not hold for  $R_i^k$  and  $R_j^k$ . According to **Lemma 1**, if  $R_i^k$  and  $R_j^k$  are not merged,  $R_i = R_i^k$  and  $R_j = R_j^k$ . This implies that **Algorithm 2** does not merge  $R_i^k$  and  $R_j^k$ . The evidence of a boundary holds for them, which is a contradiction. ■

**Theorem 2.** Given the predefined error bounds, the segmentation  $S$  by **Algorithm 2** is not over-merged according to **Definition 2**.

**Proof.** If  $S$  is over-merged, there must be a proper refinement  $T$  that is not under-merged. Let a region  $C \in S$ , and there are two adjacent regions  $A \subset C$ ,  $B \subset C$ , such that  $A$  and  $B$  satisfy the refinement  $T$ . According to **Algorithm 2**,  $A$  and  $B$  will not merge with any other regions in  $C$  before they merge with each other. Then  $C$  does not contain  $A$  and  $B$ , which is a contradiction. ■

In **Theorem 1** and **Theorem 2**, we claim that the proposed DRM method brings global properties to the segmentation result. The under-merging and over-merging issues are essential problems to deal with in many region merging algorithms. Although the predicate  $P$  is locally defined in Eq. (2), it can lead to some desirable properties which are held in the mediate segmentation after several iterations, and especially in the final segmentation. The under-merging error in **Definition 1** is easier to overcome, because once two regions are merged, there is no chance to judge them again. While for the over-merging error in **Definition 2**, the proposed DRM method forces the predicate  $P$  to make the same decision on the evidence of a boundary, and therefore it can bring a global property for the final segmentation result. This is not a trivial problem in region merging techniques. Many existing algorithms have difficulties to preserve the global property. For

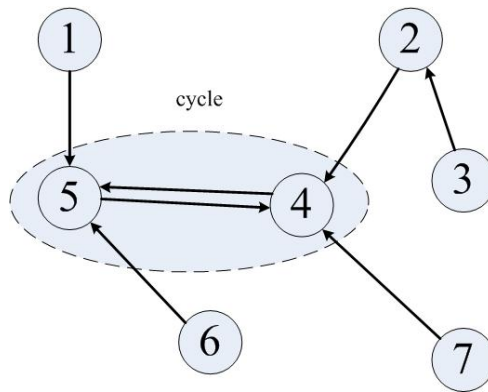
example, the method in [16] cannot overcome the over-merging error, which means that the merging of a pair of regions may contradict to some previous judgments on them, and in [22-23], the under-merging and over-merging problems are not solved with theoretical guarantees.

## V. Algorithm Acceleration by Nearest Neighbor Graph (NNG)

The DRM process presented in **Algorithm 2** depends on adjacency relationships between regions. At each merging step, the edge with the minimal weight in a certain neighborhood is required. This requires a scan of the whole graph by which the relations between neighboring regions are identified. The edge weights and nodes of RAG are calculated and stored for each graph layer. Since the positions of the edges are unknown, the linear search for these nodes and edges requires  $O(|E|)$  time. After each merging, at least one of the edges must be removed from RAG, the positions and edge weights are updated. Then a new linear search is performed for constructing the next graph layer. If the number of regions to be merged is very large, the total computational cost in the proposed DRM algorithm will be very high.

Based on the observation that only a small portion of RAG edges counts for the merging process, we can find an algorithm for accelerating the region merging process. The implementation of the algorithm relies on the structure of nearest neighbor graph (NNG), which is defined as follows. For a given RAG, where  $G = \langle V, E \rangle$ , the NNG is a directed graph  $G_m = \langle V_m, E_m \rangle$ , where  $V_m = V$ . If we define a symmetric dissimilarity function  $S$  to measure the edge weights, the directed edge is defined as:

$$E_m = \{(v_i, v_j) \mid w(v_i, v_j) = \min S(v_i, v_k), (v_i, v_k) \in E\}$$



**Figure 4.** A possible NNG of the RAG in Fig. 1 and a cycle in the NNG.

Fig. 4 shows an example of NNG with respect to the RAG in Fig. 1. From the definition we can see that the out-degree of each node equals to one and the number of edges in an NNG is  $\|V\|$ . The edge starting at a node points toward its most similar neighbor. For a sequence of graph nodes, if the starting and ending nodes coincide, we call it a cycle (see Fig. 4).

It is easy to verify that the NNG has the following properties [36]:

- (1) Along any directed edge in NNG, the weights are non-increasing.
- (2) The maximum length of a cycle is two.
- (3) NNG contains at least one cycle.
- (4) The maximum number of cycles is  $\lfloor \|V\|/2 \rfloor$ .

From the definition of merging predicate in Eq. (2), we see that when the predicate value between two regions is true, there is exactly a cycle between them. This demonstrates the stop criterion for the proposed DRM algorithm, i.e., if there is no cycle in the NNG, the region merging will stop. In other words, before the process stops, we can always have at least one pair of regions to merge according to the above property (3). This suggests that we can keep the NNG cycles during the region merging process instead of searching over the whole RAG. The original RAG is constructed from the over-segmented image, and the NNG is formed by searching for the most similar neighbors of each graph node. The NNG cycles are identified by a scan of the NNG. The accelerated region merging process is described in Table 3. In the graph theory, the NNG is also called a Minimal Spanning Tree (MST). For a general MST algorithm, the edge cost is defined in the statistic graph where the minimal cost is chosen for a single path node. While in ours, the shortest path is based on a “dynamically changed graph”, in which the graph edges are re-calculated in each iteration.

Now we have the following theorem to demonstrate that the region merging process will go on until the predicate is false for all pairs of regions in the image.

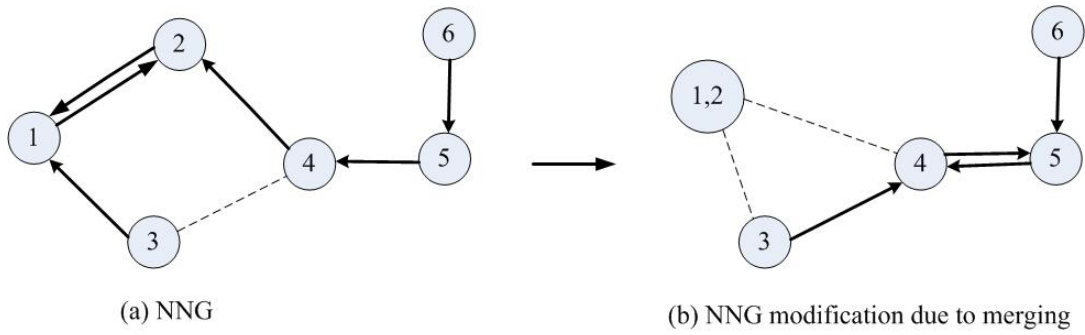
**Theorem 3.** In the DRM algorithm, there is at least one pair of regions to be merged in each iteration before the stopping criterion is satisfied.

**Proof.** According to the property (3) of the NNG, there is always at least one cycle in the graph (the number of graph nodes should be greater than 1). Therefore, the region merging process will continue until the condition (b) in Eq.(2) is not satisfied. ■



**Table 3.** Algorithm 3: Accelerating the dynamic region merging process.

<p><b>Input:</b> the initial RAG and NNG of the image.</p> <p><b>Output:</b> The region merging result (in the form of RAG).</p>
<ol style="list-style-type: none"> <li>1. Set <math>i=0</math>.</li> <li>2. For the NNG in the <math>i</math>-th graph layer, find the minimum weight edge of the RAG using the cycles.</li> <li>3. Use <b>Algorithm 1</b> to check the value of the predicate <math>P</math>. If <math>P</math> is between the cycle is true, merge the corresponding pair of regions.</li> <li>4. Update the RAG, NNG and the cycles.</li> <li>5. Set <math>i=i+1</math>.</li> <li>6. Go back to step 2 until no cycle can be found.</li> <li>7. Return RAG.</li> </ol>



**Figure 5.** An example of NNG modification. Dotted lines represent the RAG edges, while directed lines represent NNG edges.

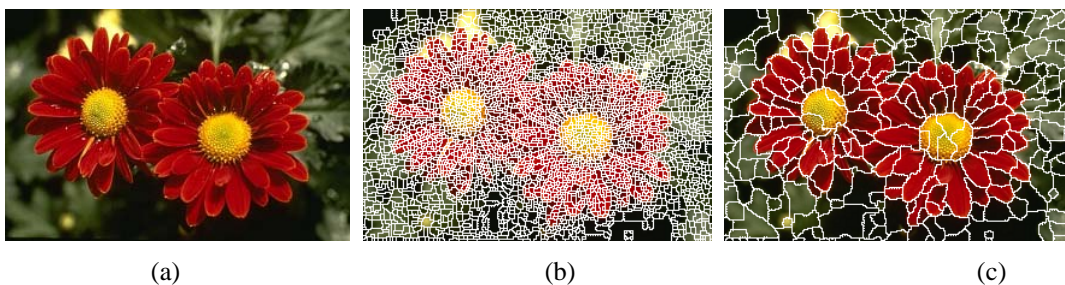
If only use RAG in the merging process, update of the RAG requires  $O(\|V\|)$  searching time. The computation will be expensive if there are a large number of nodes in RAG. In **Algorithm 3**, after the nodes of a cycle are merged, the weights of the neighboring RAG and the structure of NNG are modified. There is an observation that the new cycles can only form in the second order neighborhood of the merged nodes, which is illustrated in Fig. 5. After merging nodes 1 and 2, the closest neighbor of node 3 becomes node 4, and that of node 4 becomes node 5. A new cycle is formed between nodes 4 and 5, and by no means to be between nodes 5 and 6. In such a way, the causes of new NNG cycles can only be detected in the second order neighborhood of merged nodes. Hence, the computational effort for updating the NNG at each merging of region pair depends on the distribution of the second order neighborhood size in the RAG. The computation time for a merge of NNG cycle is  $O(\gamma^{(2)}+1)$ , where  $\gamma^{(2)}$  is the size of the second order neighborhood of the new node. In most cases,  $\gamma^{(2)}$  is far less than  $\|V\|$ , which indicates the reduction of computation time by the accelerating algorithm.

## VI. Experimental Results and Discussions

In this section, we evaluate the proposed DRM algorithm on the Berkeley Segmentation Dataset<sup>2</sup> (BSDS), which contains 100 test images with 5-10 human segmentations on each one of them as the ground-truth data. In Section VI-A, we test the DRM algorithm with several representative examples. In Section VI-B, we compare the well known mean-shift algorithm [32] and the graph-based region merging method [18] with the DRM algorithm. In Section VI-C, the performance of the accelerated DRM algorithm is evaluated. There are several free parameters in the proposed algorithm, and we test the effects of them in Section VI-D. Some discussions on the DRM method and its potential extensions will be referred in Section VI-E. The code and software of the proposed method can be found at <http://www4.comp.polyu.edu.hk/~cslzhang/DRM.htm>.

### VI-A. Analysis on DRM by representative examples

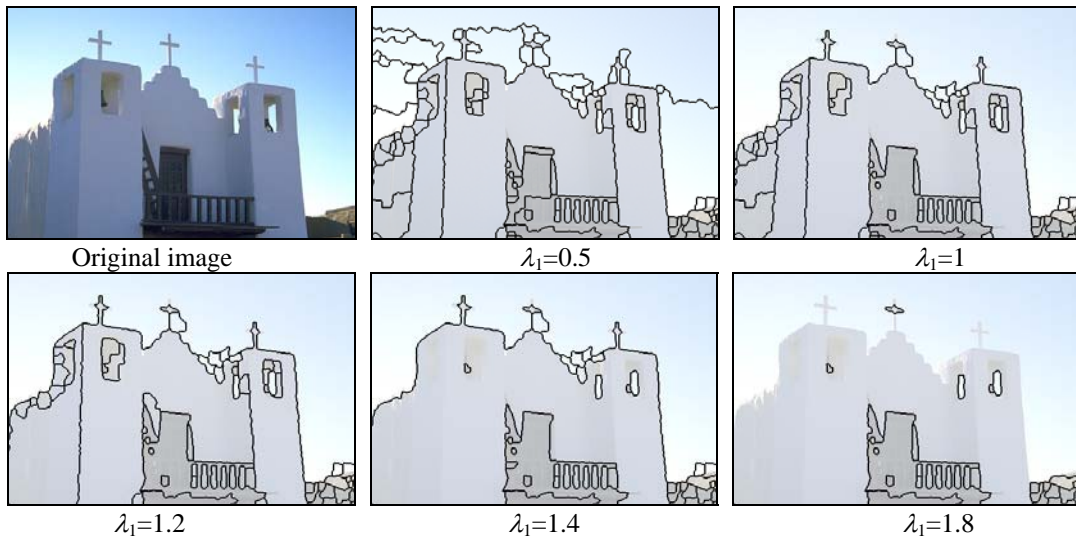
The proposed DRM algorithm starts from an initially over-segmented image. For simplicity, we use the watershed algorithm to obtain the initial segmentation. Certainly, a more sophisticated initial segmentation method, such as the mean-shift algorithm [32], may lead to a better final segmentation result. Since the standard watershed algorithm is very sensitive to noise and hence leads to severe over-segmentation (see Fig 6(b) for an example), to reduce noise and trivial structures we apply median filtering on the gradient image before using the watershed algorithm. Fig. 6(c) shows the result of the modified watershed segmentation, where the over-segmentation is reduced a lot while preserving the desired the object boundaries.



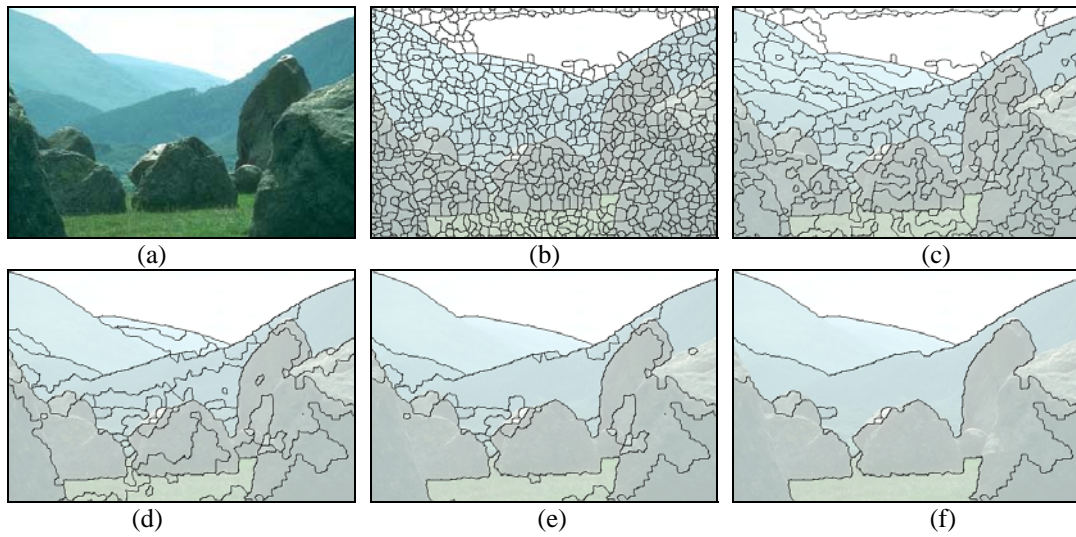
**Figure 6.** (a) Original image; (b) initial segmentation by standard watershed algorithm; and (c) initial segmentation by modified watershed algorithm with median filtering on the gradient image.

---

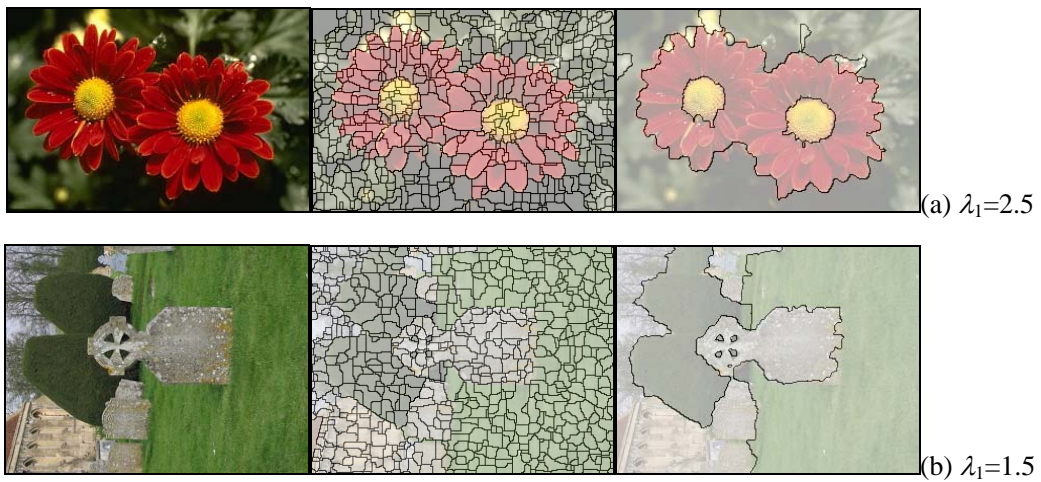
<sup>2</sup> <http://www.cs.berkeley.edu/projects/vision/grouping/segbench/>

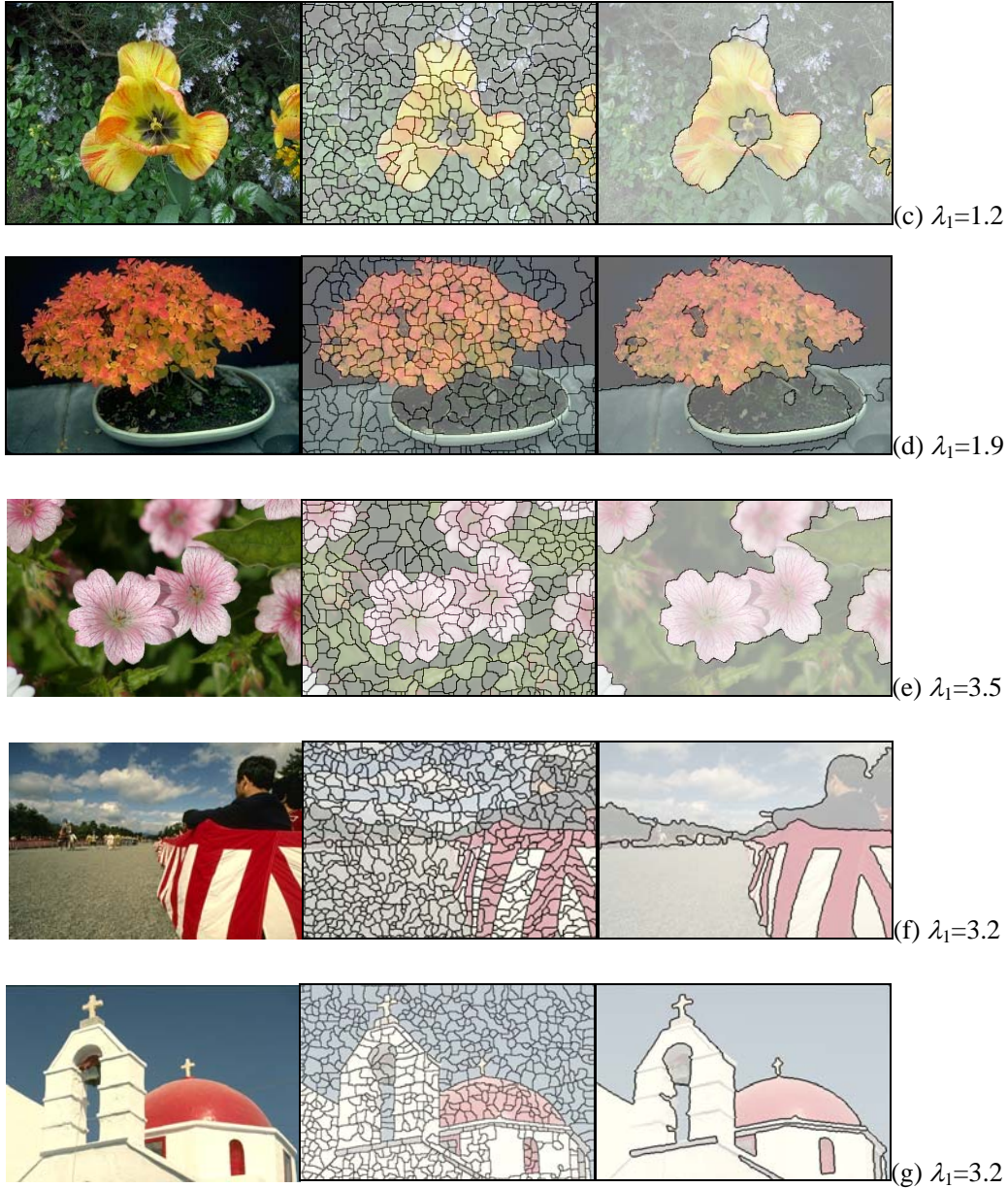


**Figure 7.** Segmentation results with  $\lambda_1=0.5, 1, 1.2, 1.4, 1.8$ , respectively.



**Figure 8.** Region merging process. (a) the original image; (b) the result of initial watershed segmentation; (c)-(f) the merging results in different stages. ( $\lambda_1=0.8$ )





**Figure 9.** Segmentation results by the proposed algorithm. From left to right, the first column shows the original images. The second column shows the over-segmentation produced by watershed algorithm. The third column shows our segmentation results.

As stated in Section III, the values of  $\lambda_1$  and  $\lambda_2$  in the cue distribution functions (refer to Eq. (4)) control the degree of consistency between two regions, and hence decide when to terminate the region merging. In practice, we set  $\lambda_2$  to be a constant value 1, and hence there is only one parameter  $\lambda_1$  to be set. By experimental experience, we set  $\lambda_1$  between 0.1 and 5. Fig. 7 shows the segmentation results of an example image with different values of  $\lambda_1$ . When  $\lambda_1$  is large, the algorithm is more likely to take two neighboring regions as consistency. Therefore it may produce over-merged regions (see the result where  $\lambda_1=1.8$ ). When  $\lambda_1$  is small, the case is inverse and as a result produces under-merged regions (see the results where  $\lambda_1=0.5$ ,

1). To clearly show the region merging process, we give an example in Fig.8, in which primitive regions (in Fig. 8(b)) are merged iteratively until the stop criterion is satisfied (in Fig. 8(f)).

We illustrate the proposed DRM algorithm using more example images in Fig. 9. It is clear that neighboring regions with coherent colors are merged into one, whereas the boundaries are well located on the reasonable places. Some of the large regions have substantial variations inside (Figs. 9 (c), (e), (f)) however, with relatively slow changes of colors along the boundaries. This indicates that the DRM algorithm can tolerate some variations for grouping meaningful regions in an image.

## VI-B. Comparison with some well-known methods

In this section we compare the segmentation results of DRM with the mean-shift algorithm [32] and the graph-based algorithm [18], which are among the well-known works in automatic image segmentation. Mean-shift performs the segmentation by clustering the data into several disjoint groups. The cluster centers are computed by first defining a spherical window and then shifting the window to the mean of the data points with a radius  $r$ . This shifting process repeats until convergence. The segmented image is constructed using the cluster labels. The mean-shift algorithm takes the feature (range) bandwidth, the spatial bandwidth, and the minimum region area (in pixels) as input. We need to adjust these three parameters for a good segmentation. In the graph-based algorithm [18], the predicate is based on both the external and the internal intensity differences of the regions. More specifically, the external difference is calculated as the minimum weight edge between two neighboring regions and the internal difference of a region is the largest weight in the minimum spanning tree of that region. The predicate is then defined by comparing these two values to make an adaptive decision with respect to the local characteristics of an image. The graph-based algorithm [18] also contains three parameters, which are used to smooth the image ( $\sigma$ ), to control the minimum region size ( $min$ ) and to set the expected number of regions ( $K$ ). All of these parameters are in different ranges, making the tuning of parameters tricky. As stated in Section III, the proposed DRM algorithm contains five free parameters  $\{m, \lambda_1, \lambda_2, \alpha, \beta\}$  in the consistency test. In implementation, we fix  $\lambda_2$  to be 1, both  $\alpha$  and  $\beta$  to be 0.05 and  $m$  to be half size of the region. Hence only the value of  $\lambda_1$  is needed to tune. In this sense, DRM is much more convenient to control than the mean-shift algorithm [32] and the graph-based algorithm [18].

Figs. 10(b)-(d) show the segmentation results by the graph-based method [18], the mean-shift method [32] and the proposed DRM method. Segmentations are obtained by choosing the results with roughly the same number of regions from the three methods. By this setting, we can compare the performance of them in the same segmentation granularity. In Figs. 10(b)-(c), some major boundaries in the ground truth are missed, and some boundaries are over-detected. In Fig. 10(d), we can see that the proposed method can retain good localization of boundaries. In Fig. 10(e), the human labeled segmentation is used as the reference (ground truth) for visual evaluation of the segmentation quality. For each image, 5-10 segmentations produced by different human observers are chosen, and the consistency among different human segmentations is indicated by the probability-of-boundary ( $P_b$ ) images (Fig. 10(e)), where the darker pixels indicate more people marked them as a boundary. For more information about the generation of human labeled ground truth segmentation, please refer to [37].

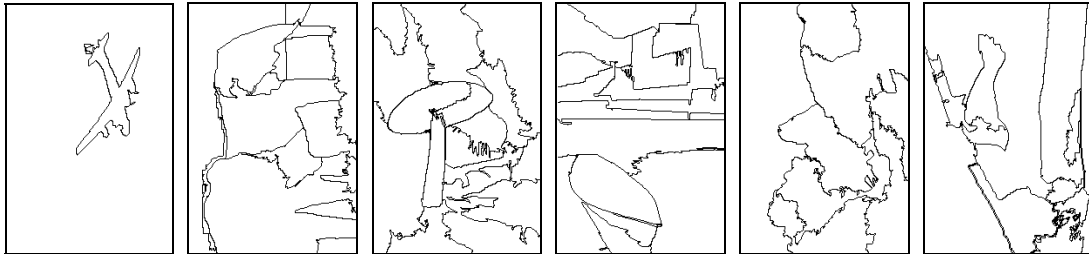
Let's then discuss about the quantitative measure of segmentation quality. This task might be as hard as segmentation itself. In the past decades, many researchers have been studying the supervised evaluation methods, where the segmented images are compared with a ground truth image. However, in unsupervised image segmentation, often there is no ground truth available. In order to build a common platform for researchers to evaluate various image segmentation methods, a group of human segmented images for each test sample are provided in the Berkeley Segmentation Dataset (BSDS), and a boundary-based evaluation method was proposed by Martin *et al.* [38]. In [38], the precision-recall framework is applied in conjunction with the human-marked boundaries. The precision-recall is a well-accepted measure in image segmentation, which considers two aspects of boundary qualities: *precision* is the fraction of detections that are true positives rather than false positives, while *recall* is the fraction of true positives that are detected rather than missed. A combination of these two quantities is defined as the *F*-measure in [39]:

$$F=PR/(\tau R+(1-\tau)P)$$

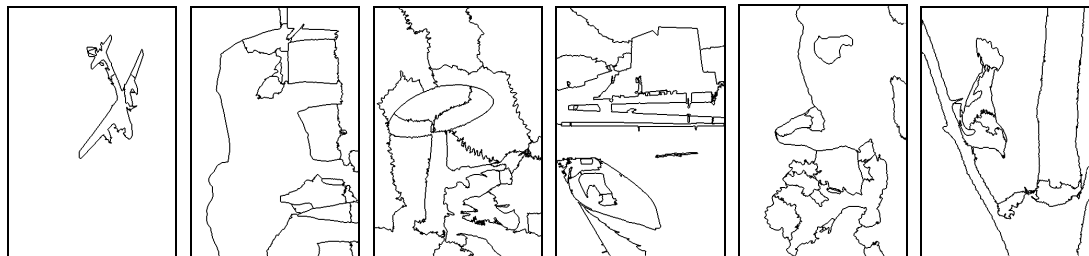
where  $\tau$  is a relative cost between  $P$  and  $R$ . An *F*-measure curve can be obtained by changing the algorithm parameters. Since the *F*-measure curve is usually unimodal, the maximal one may be taken as a summary of the algorithm's performance in the sense that large value of *F*-measure indicates high quality of image segmentation. In our experiment, the optimal parameter is chosen for each image and the corresponding *F*-measure is recorded for the overall evaluation.



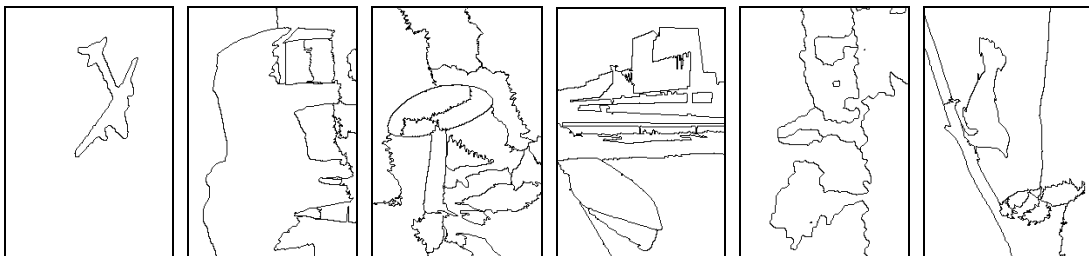
(a) Original images.



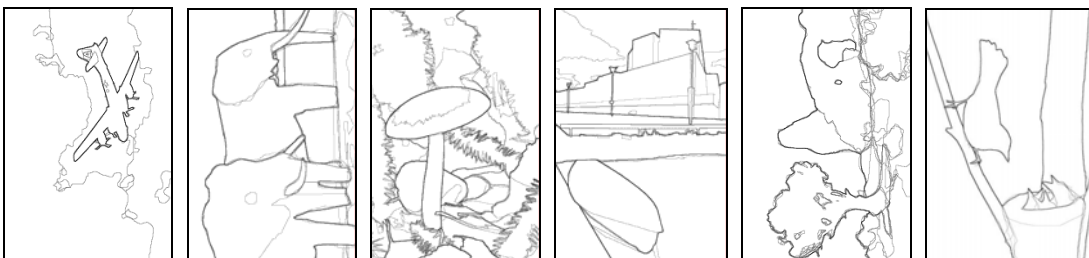
(b) Segmentation results by the graph-based region merging method in [18].



(c) Segmentation results of mean-shift method in [32].



(d) Segmentation results of the proposed method with watershed initializations.



(e) Ground truth.

**Figure 10.** Segmentation results by different methods. The first row shows the original images. The second row shows the results by the graph-based image segmentation method [18]. Parameter settings ( $[\sigma, K, min]$ ) for images (from the left to the right) are (0.5, 1200, 400), (0.5, 800, 200), (0.5, 400, 200), (0.5, 1500, 700), (0.5, 800, 300) and (0.5, 500, 200), respectively. The third row shows the results by the mean-shift method [32], with the spatial bandwidth and color bandwidth being set as (14, 13), (13, 12), (18, 17), (14, 13), (14, 13) and (12, 11), respectively. The fourth row shows the proposed method. The values of  $\lambda_1$  are 0.6, 0.5, 2, 1.1, 1.1 and 0.9, respectively. The fifth row shows probability-of-boundary images of the human segmentation, where the darker boundaries indicate more subjects marked a boundary [38].

There are two issues to address when use BSDS for evaluating the segmentation results. First, the ground truth is defined by a collection of 5 to 10 human segmentations for each image. Simply uniting the humans' boundary maps is not a reasonable choice because it ignores the high overlapping boundaries among different humans. Second, when matching boundary pixels, one should avoid over-penalizing the slightly mislocalized boundaries. We use the method proposed by Martin *et al.* to compute the  $F$ -measure for all algorithms. In this method, the segmentation results are compared with each human segmentation separately. False positives are counted as the boundary pixels that do not match any human boundary. The recall depends on all the human segmentations, i.e., averaged over different human data. One merit of Martin *et al.*'s method is that it can tolerate some localization errors in the matching process, which is a reasonable consideration in the correspondence of ground-truth and segmentation results. In Table 4, we compared the  $F$ -measures of the Canny edge detector [5], the graph-based method [18], the mean-shift method [32] and the DRM algorithm. The best parameter is chosen for each of these methods, and we have the  $F$ -measures as 0.62 for Canny edge detector, 0.62 for the graph-based method [18], 0.65 for mean-shift method [32] and 0.65 (with watershed as initial segmentation) and 0.66 (with mean-shift as initial segmentation) for the proposed DRM method.

**Table 4.** The best  $F$ -measures by the competing methods on the BSDS dataset. The average  $F$ -measure for the human subjects is 0.79 [40], which represents the human performance for the segmentation task. The DRM algorithm is tested when initialized by watershed and mean-shift respectively.

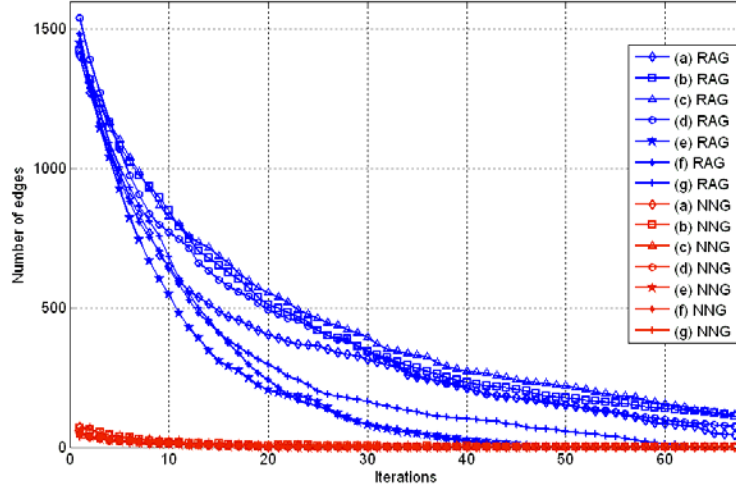
Method	Human	Graph-based method [18]	Canny [5]	Mean-shift [32]	DRM (initialized by watershed)	DRM (initialized by mean-shift)
$F$ -measure	0.79	0.62	0.62	0.65	0.65	0.66

### VI-C. The performance of accelerated DRM algorithm

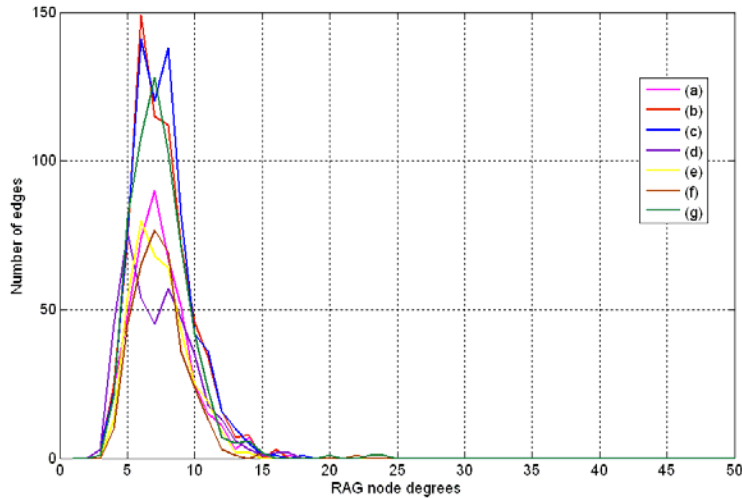
In Fig. 11, we show the number of RAG regions and the number of NNG cycles for images Fig. 9(a)-(g) at different region merging stages. Apparently, the number of cycles (in red) is much smaller than that of the RAG edges (in blue). By using the accelerated region merging algorithm, the computational effort is largely saved. The update of NNG cycles depends on the size of RAG in the second order neighborhood. In Fig. 12, we show the histogram of RAG node degrees for the over-segmented images in Figs. 9(a)-(g) (in the second column). We can see that most of the nodes have the degree less than 15; therefore the computational cost for updating the NNG cycles is not expensive. This acceleration algorithm would be very helpful in the high-resolution environment where a massive graph containing billions of vertices will exist. In our test of



100 Berkeley images, the running time of the proposed algorithm ranges from 5 to 15 seconds on a PC with Intel Core 2 Duo 2.66 GHz CPU, 2GB memory. It is largely depended on the number of regions in the initial segmentation and that in the final segmentation.



**Figure 11.** The number of RAG edges (in blue) vs. the number of NNG cycles (in red) for images in Figs. 9(a)-(g) at different region merging stages.



**Figure 12.** Histograms of the RAG node degrees for images in Figs. 9(a)-(g) (shown in different colors).

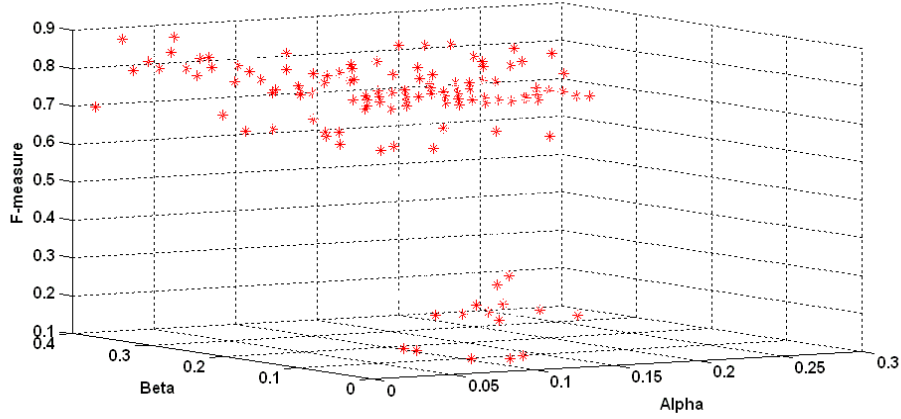
#### VI-D. The choices for parameters

In the proposed DRM method, there are five free parameters  $\{m, \lambda_1, \lambda_2, \alpha, \beta\}$  which control the consistency hypothesis evaluation. In implementation, we fixed four of them (i.e.,  $m, \lambda_2, \alpha, \beta$ ) to be constant. However, it is still worth investigating their impact on the stability and robustness of the whole segmentation process.

In SPRT,  $m$  is used to decide the amount of data selected for the random test. We empirically choose it to be the half size of the region, as in the experiments we found that the segmentation results are not sensitive to the moderate changes of  $m$ .

Then we test the choices of parameters  $\alpha$  and  $\beta$ , which represent the probability of accepting an “inconsistent” model as “consistent” and rejecting a “consistent” model as “inconsistent”, respectively. In Fig.13, we show 120 segmentation results of Fig. 9(a) in the form of  $F$ -measures, which are produced by different pre-specified  $(\alpha, \beta)$  values. In the experiment,  $\alpha$  ranges from 0.01 to 0.26 and  $\beta$  ranges from 0.01 to 0.4. In general, we can see that the lower the test error is, the higher the obtained segmentation quality is. In our implementation, both  $\alpha$  and  $\beta$  are set to a fixed value 0.05. Theoretically, the values of  $\alpha$  and  $\beta$  will also affect the number of tests needed for the SPRT, as shown in Eq. (6) proposed by Wald. The upper bound of the number of tests  $N_0$  is set based on the *Truncated* SPRT [28]. This has been explained in Section III.

The last experiment is performed to test the choices of parameter  $\lambda_1$ , which is the only user input parameter in the implementation. As defined in Eq. (3) and Eq. (4), if we set  $\lambda_2=1$ ,  $\lambda_1$  can be used to balance the relative weight of  $P_0$  and  $P_1$  so that only  $\lambda_1$  needs to be tuned. For a small value of  $\lambda_1$ , it requires stronger evidence for a boundary. In other words, the over-merging is not likely to happen with small values of  $\lambda_1$ . Generally, reducing the value of  $\lambda_1$  will produce more regions in the segmentation result. The control of this parameter leads to a hierarchy of segmentations at different scales. Table V shows the number of segmented regions under different values of  $\lambda_1$  (the rest parameters are fixed in this experiment). As  $\lambda_1$  increases, the number of regions tends to decrease. In our experiments, the value of  $\lambda_1$  is chosen from 0.1 to 5. Notice that it is hard to find a generally appropriate  $\lambda_1$  for all the images, since the variety and multiscale interpretation of the image content.



**Figure 13.** The  $F$ -measures of Fig. 9(a) with 120 pre-specified  $(\alpha, \beta)$  values. In the experiment,  $\alpha$  ranges from 0.01 to 0.26 and  $\beta$  ranges from 0.01 to 0.4.

**Table V.** The number of regions w.r.t. different values of  $\lambda_1$ . Indices of images are corresponding to the original images in Fig. 10 (from the left to the right)

Fig.10-1	$\lambda_1$	0.3	0.4	0.8	1.0	1.01	1.02
	# of regions	28	27	24	15	7	2
Fig.10-2	$\lambda_1$	0.3	0.8	0.92	0.96	0.98	1.0
	# of regions	167	150	134	117	82	67
Fig.10-3	$\lambda_1$	0.9	0.98	1.2	1.4	2.0	2.3
	# of regions	375	285	130	126	93	72
Fig.10-4	$\lambda_1$	1.0	1.2	1.4	2.1	3.5	4.0
	# of regions	235	124	95	91	80	17
Fig.10-5	$\lambda_1$	1.0	1.1	1.2	1.5	1.9	2.0
	# of regions	445	225	178	148	92	21
Fig.10-6	$\lambda_1$	0.8	0.9	0.94	1.0	1.2	1.3
	# of regions	192	181	174	121	102	27

## VI-E. Discussions

This paper presents a framework for region merging. The simple cue, i.e., region color, is used in the implementation. Naturally, different cues (e.g., color, texture, shape, etc.) or a combination of them can be explored by changing the definition of conditional probabilities in SPRT (Eq. (4)). The region merging predicate uses the minimal weight edge between two regions to measure the difference between them. It is guaranteed that some global properties can be obtained under this predicate. However, the predicate might lead to a limitation for capturing the perceptual differences between two neighboring local regions. Some failure examples are given in Fig. 14. We can see that the proposed DRM algorithm may miss some long but weak boundaries (1<sup>st</sup> example), and it may also merge the regions with short but high contrast boundary (2<sup>nd</sup> example).



**Figure 14.** Failure examples of the proposed DRM method. The first column shows the original images, the second column shows the initial segmentation by watershed algorithm, and the third column shows the region merging results by the proposed method. We see that DRM methods may miss some weak boundaries (in the 1st example) and merge the regions with short but high contrast boundary (in the 2nd example).

Another problem one might concern is that the randomness nature of SPRT might lead to non-unique partitions to an image. To quantitatively evaluate its influence on the segmentation results, we randomly select 20 images and run our algorithm for 10 times on each one of them. Segmentations of the same image are compared by calculating the ratio of discrepancy to the amount of boundary pixels for an image. The result turns out to be 0.88% on average, which indicates an irrelevance of the randomness of SPRT.

There are several potential extensions to DRM to solve the above problems. For example, we can add a global refinement step to correct the miss-classified regions due to local decisions. Many merging errors in DRM are due to the insufficient local (perceptual) information used to making the merging decision. However, if we view the outputted labels of DRM as initial markers of the image content, then these initial markers can be used to calculate some global statistics of the image so that a global refinement can be defined to correct the DRM errors according to some criterion. Such a DRM with refinement scheme can exploit both local and global image features, and hence better results can be expected.

Another potential extension is the introduction of user interaction. With some user guidance, the initial label of part of the image regions can be assigned beforehand, which will provide useful information for the region merging process. An interactive DRM algorithm can then be developed to accomplish the image segmentation.

## VII. Conclusions

In this paper, we proposed a novel method for segmenting an image into distinct components. The proposed

algorithm is implemented in a region merging style. We defined a merging predicate  $P$  for the evidence of a merging between two neighboring regions. This predicate was defined by the sequential probability ratio test (SPRT) and the maximum likelihood criterion. A dynamic region merging (DRM) was then presented to automatically group the initially over-segmented many small regions. Although the merged regions are chosen locally in each merge stage, some global properties are kept in the final segmentations. For the computational efficiency, we introduced an accelerated algorithm by using the data structure of region adjacency graph (RAG) and nearest neighbor graph (NNG). Experiments on natural images showed the efficiency of the proposed algorithm. There are several potential extensions to this work, such as the introduction of global refinement and user interaction, etc. Those will be further investigated in our future work.

## References

- [1] D.A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall, 2002
- [2] L. Ladicky, C. Russell, P. Kohli, P. Torr. Associative Hierarchical CRFs for Object Class Image Segmentation. In: *ICCV 2009*.
- [3] F. Lecumberry, A. Pardo and G. Sapiro. Simultaneous object classification and segmentation with high-order multiple shape models. *IEEE Transactions on Image Processing*. pp: 625 - 635, 2010.
- [4] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison Wesley, Reading, MA, 1992.
- [5] J. Canny. A Computational Approach to Edge Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679-698, 1986.
- [6] B. Paul, L. Zhang and X. Wu, "Canny edge detection enhancement by scale multiplication," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1485-1490, Sept. 2005.
- [7] L. Zhang, B. Paul, et al, "Edge detection by scale multiplication in wavelet domain," *Pattern Recognition Letters*, vol. 23, pp. 1771-1784, 2002.
- [8] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 2000*.
- [9] S. Wang, J. M. Siskind. Image Segmentation with Ratio Cut, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):675-690, 2003.
- [10] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. November 1993.
- [11] H. D Cheng, Y. Sun. A hierarchical approach to color image segmentation using homogeneity. *IEEE Transactions on Image Processing*. Volume: 9 , Issue: 12, page(s): 2071-2082, 2000.
- [12] S. Lee; M.M. Crawford. Unsupervised multistage image classification using hierarchical clustering with a bayesian similarity measure. *IEEE Transactions on Image Processing*. Page(s): 312 -320, 2005.
- [13] A. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel lattices. *CVPR*, 2008
- [14] A. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, and G. Jones. Scene shape priors for superpxiel segmentation. *ICCV*, 2009.
- [15] A. Moore, S. Prince . "Lattice Cut" - Constructing superpixels using layer constraints. *CVPR 2010*.

- [16] R. Nock and F. Nielsen. Statistic region merging. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol 26, pages 1452-1458, 2004.
- [17] K. Haris and S. N. Estradiadis and N. Maglaveras and A. K. Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on Image Processing*, vol 7, pp. 1684-1699, Dec. 1998.
- [18] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient Graph-Based Image Segmentation *International Journal of Computer Vision*. Vol. 59, Number 2, September 2004.
- [19] B. Peng, L. Zhang and J. Yang, Iterated Graph Cuts for Image Segmentation. In *Asian Conference on Computer Vision*, 2009.
- [20] Moscheni, F. Bhattacharjee, S. Kunt, M. Spatio-temporal segmentation based on region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 20, pages: 897-915. Sep 1998.
- [21] J. Ning, L. Zhang, D. Zhang and C. Wu. Interactive Image Segmentation by Maximal Similarity based Region Merging. *Pattern Recognition*, vol. 43, pp. 445-456, Feb, 2010
- [22] F. Calderero, F. Marques. Region merging techniques using information theory statistical measures. *IEEE Transactions on Image Processing*. Volume: 19 , Issue: 6. page(s): 1567-1586, 2010.
- [23] F. Calderero, F. Marques. General region merging approaches based on information theory statistical measures. *The 15th IEEE International Conference on Image Processing (ICIP)*. pp: 3016-3019, 2008.
- [24] H. Liu, Q. Guo, M, Xu, I. Shen. Fast image segmentation using region merging with a k-Nearest Neighbor graph. *IEEE Conference on Cybernetics and Intelligent Systems*. Page(s): 179-184, 2008.
- [25] Y. Shu, G. A. Bilodeau, F. Cheriet. Segmentation of laparoscopic images: integrating graph-based segmentation and multistage region merging. *The 2nd Canadian Conference on Computer and Robot Vision*, 2005.
- [26] I.E. Gordon, *Theories of Visual Perception*, first ed. John Wiley and Sons, 1989.
- [27] K. Koffka. *Principles of Gestalt Psychology*. New York: Harcourt, Brace and World. 1935.
- [28] A. Wald. *Sequential Analysis*. Wiley Publications in Statistics, Third ed. Wiley, 1947.
- [29] Z. Wu, Homogeneity testing for unlabeled data: A performance evaluation, *CVGIP: Graph. Models Image Process.*, vol. 55, pp. 370-380, Sept. 1993.
- [30] A. Trémeau and P. Colantoni. Regions adjacency graph applied to color image segmentation. *IEEE Transactions on Image Processing*. Vol. 9, pp. 735-744, 2000.
- [31] L. Vincent, P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (6) 583–598, 1991.
- [32] D. Comanicu, P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24, 603-619, May 2002.
- [33] Bellman, Richard, *Dynamic Programming*, Princeton University Press, 1957.
- [34] A. Mishra, P. Fieguth, D. Clausi. Accurate boundary Localization using Dynamic Programming on Snakes. *Proceedings of the Canadian Conference on Computer and Robot Vision*, pp 261-268, 2008.
- [35] A. A. Amini, T.E. Weymouth, and R.C. Jam. Using dynamic programming for solving variational problems in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(9):855-867, 1990.
- [36] D. Eppstein, M. S. Paterson, and F. F. Yao. On nearest-neighbor graphs. *Discrete & Computational Geometry* 17(3):263–282, Apr 1997.
- [37] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *ICCV* 2001.
- [38] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.

- [39] C. Van Rijsbergen. Information Retrieval, second ed. Dept. of Computer Science, Univ. of Glasgow, 1979.
- [40] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From Contours to Regions: An Empirical Evaluation. IEEE Conference on Computer Vision and Pattern Recognition, 2009.
- [41] Yuri Boykov, Olga Veksler, Ramin Zabih. Fast Approximate Energy Minimization via Graph Cuts. In IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222-1239, 2001.
- [42] Yuri Boykov, Vladimir Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. In IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1124-1137, Sept. 2004.
- [43] Andrew DeLong, Anton Osokin, Hossam Isack, Yuri Boykov. Fast Approximate Energy Minimization with Label Costs. In IEEE conference on Computer Vision and Pattern Recognition, San Francisco, California, June, 2010.
- [44] Ben Ayed and Mitiche, A Region Merging Prior for Variational Level Set Image Segmentation, IEEE Transactions on Image Processing. Vol. 17, issue 12, pp. 2301-2311, 2008.
- [45] Serge Beucher and Christian Lantuéjoul. Use of watersheds in contour detection. In International workshop on image processing, real-time edge and motion detection. 1979.
- [46] M. Fernand. Sequential algorithms for cell segmentation: maximum efficiency? In proceedings, International symposium on clinical cytometry and histometry. Scholoss Elmau, 1986.
- [47] S. Beucher. Watershed, hierarchical segmentation and waterfall algorithm. In Mathematical Morphology and Its Applications to Image Processing. Boston, MA: Kluwer, 1994, pp. 69-76.
- [48] B. Marcotegui, S. Beucher. Fast implementation of waterfall based on graphs. Volume 30 of Computational Imaging and Vision, pages 177-186. Springer-Verlag, Dordrecht, (2005).
- [49] M. Fernand. Hierarchies of partitions and morphological segmentation. In IEEE Workshop on scale-space and Morphology in computer vision & ICCV 2001.
- [50] T. Wan, N. Canagarajah, and A. Achim, "Statistical multiscale image segmentation via alpha-stable modeling," in IEEE Int.Conf. Image Processing (ICIP), vol. 4, pp. IV-357-IV-360, 2007.
- [51] S. LaValle and S. M. Hutchinson, "Bayesian region merging probability for parametric image models," in Proc. IEEE Computer Soc.Conf. Computer Vision Pattern Recognition, Jun. 15-17, pp. 778-779, 1993.