Regularized Tree Partitioning and Its Application to Unsupervised Image Segmentation

Jingdong Wang, Huaizu Jiang, Yangqing Jia, Xian-Sheng Hua, Changshui Zhang, and Long Quan

***Abstract*—In this paper, we propose regularized tree partitioning approaches. We study normalized cut (NCut) and average cut (ACut) criteria over a tree, forming two approaches: normalized tree partitioning (NTP) and average tree partitioning (ATP). We give the properties that result in an efficient algorithm for NTP and ATP. Moreover, we present the relations between the solutions of NTP and ATP over the maximum weight spanning tree of a graph and NCut and ACut over this graph. To demonstrate the effectiveness of the proposed approaches, we show its application to image segmentation over the Berkeley image segmentation data set and present qualitative and quantitative comparisons with state-of-the-art methods.**

***Index Terms*—Grouping, image segmentation, graph partitioning, regularized tree partitioning**

I. Introduction

Image segmentation is a fundamental but challenging problem in computer vision and image processing. It could be defined as partitioning the set of pixels forming the image, or clustering its pixels. Many computational vision problems, such as object detection and recognition, stereo and motion estimation, image search and so on, could in principle make good use of segmented images. In this paper, we study clustering approaches to image segmentation and focus on graph-based solutions.

A. Related Work

In the past several years, there has been significant interest in graph-based clustering approaches for unsupervised image segmentation [12], [13], [14], [17], [19], [22], [32], [33], [34], [36], [37], [40], [43], [44], [45], [48], [49]. These approaches represent the image by a weighted graph, where each vertex corresponds to an image pixel or a region and each edge is weighted with the similarity of the pixels or regions connected by that edge. This graph is partitioned into components in a way that minimizes some cost function of the edges within those components and/or the boundary edges between those components.

The pioneer work on graph-based image segmentation is based on a cut criterion, minimum cut proposed by Wu and Leahy [45], which aims to find a graph partitioning so that the similarities over the edges connecting different components (called boundary cut) are minimized. This criterion, however, has a bias toward short boundaries and thus tends to find small components. This bias is addressed with regularized cut, e.g., ratio cut by Cox et al. [12], normalized cut by Shi and Malik [33] and average cut by Soundararajan et al. [34]. Ratio cut defines a weight within a component, and aims to minimize the ratio between the boundary cut and the weight. Normalized cut takes into account the self-similarities within components, and leads to a cost function, with the summation of ratios between boundary cuts across components and self-similarities within components. Average cut alternatively considers the pixel number within each component and results in a cost function, sums of ratios between boundary cuts across components and pixel numbers within components. Isoperimetric cut by Grady and Schwartz [17] suggests to use a general measure within components, called combinatorial volume, to regularize the boundary cut.

It has been shown from theoretical and practical aspects that the regularized cut criteria are superior over minimum cut in clustering and segmentation. However, the regularized cut criteria, including ratio cut, normalized cut, average cut, and isoperimetric cut, all yield NP-hard computational problems. Although approximate methods for computing minimum regularized cuts, e.g., spectral relaxation [17], [19], [33], [34] and semi-definite relaxation [46], have been developed, for the general cases the accuracy in these approximations is not easily estimated. In practice, they are still fairly hard to compute, limiting the methods to relatively small images or requiring high computational cost.

Some other graph-based methods adopt local criteria and conduct a bottom-up strategy to heuristically aggregate the data points into more and more compact clusters. One of the representative methods is presented in [14]. This method incrementally unions two small clusters (initially a cluster only consists of a single data point) into a bigger one, based on the weights of the edges connecting the two clusters. This method is computationally efficient, but may not get satisfactory segmentation results due to the simple union criterion and the local optimization strategy.

J. Wang is with Microsoft Research, Beijing, P.R. China.
E-mail: jingdw@microsoft.com
H. Jiang is with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Shaanxi, P.R. China.
E-mail: hzjiang@mail.xjtu.edu.cn
Y. Jia is with the computer science division, University of California, Berkeley, CA, USA.
jiayq@berkeley.edu
X.-S. Hua is with Microsoft Research, Redmond, USA.
E-mail: xshua@microsoft.com
C. Zhang is with the department of automation, Tsinghua University, Beijing, P.R. China.
E-mail: zcs@mail.tsinghua.edu.cn
L. Quan is with the department of computer science and engineering, the Hong Kong University of Science and Technology, Hong Kong, P.R. China.
E-mail: quan@cse.ust.hk

Early approaches to image segmentation, region splitting and merging [20], [21], and region growing [2], [7], [28], [38], are very close to graph-based approaches. Shi and Malik [33] present an iterative merging scheme and a global recursive scheme to minimize the normalized cut for k-way cut. From this point of view, region splitting and merging and region growing essentially form the image into a graph and then partition the graph with some local criteria in a greedy manner.

Zahn [49] presents a segmentation approach based on the minimum cut criterion over the maximum weight spanning tree of the image graph. The bipartition with the minimum cut criterion can be easily achieved by cutting the edge with the smallest weight since there is no loop in the tree. The segmentation result, however, suffers from the shortcoming of cutting small regions. This shortcoming is partially removed by Urquhart [36], in which the weight of an edge is normalized using the largest weight incident on the vertices touching that edge.

Recently, image segmentation with a maximum weight spanning tree representation has been also studied. Allene et. al [3] analyze the links between spanning forest, minimum cut and watershed and exhibit some particular cases, where a strong relation exists between these structures. Couprie et. al [10] instead present a power watershed framework, a supervised algorithm, to connect graph-based segmentation approaches including graph cuts, random walker, spanning forests and so on. Najman [29] presents an alternative way of thinking hierarchical segmentation [30] that completes existing ones, e.g., ultrametric distances, minimum spanning tree. Guigues et. al [18] present the image analysis based on scaled sets for hierarchical segmentation. In contrast to the prior works using the maximum weight spanning tree, this paper proposes regularized tree partitioning approaches and shows its application to unsupervised image segmentation by using the maximum weight spanning tree approximating the image graph.

Besides, there are supervised clustering and segmentation approaches, such as graph-cuts [6], label propagation [41], and semi-supervised learning algorithms [51]. We have also proposed supervised tree partitioning approaches for image segmentation [24], [39]. Those approaches require initial labeling over seeds from users and differ from the goal of this paper that aims to do automatic clustering.

B. Our Approach

This paper mainly focuses on improving clustering algorithms that are based on regularized graph partitioning [17], [33], [34]. Regularized graph partitioning enjoys good clustering criteria, but is computationally infeasible, which leads to approximate solutions and thus deteriorated segmentation quality. We propose regularized tree partitioning approaches, which can efficiently optimize good clustering criteria and is able to lead to superior performance.

We first present normalized tree partitioning (NTP) and average tree partitioning (ATP) that optimize normalized cut and average cut over a tree, and give the properties over a tree that lead to an efficient optimization algorithm. Then, we analyze the necessary condition on which NTP and ATP over the MST of a graph achieve the exact bipartition over the graph and the same bipartition to the optimal solution of normalized cut and average cut over the graph. Next, we present new partitioning criteria that also derive NTP and ATP over a tree and give sufficient and necessary conditions on which they can achieve the exact and optimal partition. Last, we extend NTP and ATP and reach two approaches, maximum normalized tree partitioning (MaxNTP) and maximum average tree partitioning (MaxATP). We present efficient optimization algorithms to NTP, ATP, MaxNTP, and MaxATP. Quantitative and qualitative comparison results of image segmentation over the Berkeley image data set demonstrate that regularized tree partitioning achieves competitive results.

To summarize, this paper extends our previous work [40] that first introduces the NTP algorithm, and provides at least two contributions. On the one hand, we give a new interpretation to NTP, which helps understand NTP more deeply. On the other hand, we present a regularized tree partitioning framework and introduce several new tree partitioning algorithms, including ATP, MaxNTP, MaxATP, and so on.

II. Preliminaries

Graph-based grouping methods first represent a set of data points in an arbitrary feature space as a weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where a node $v \in \mathcal{V}$ in the graph corresponds to a point in the feature space, an edge $e \in \mathcal{E}$ is formed between a pair of nodes u and v, and the weight $w \in \mathcal{W}$ on the edge e is a function of the similarity between nodes u and v.

The widely-used graph representation of a set of data points is a neighborhood graph, which is constructed by connecting each point and its nearest neighbors. In general data points, the nearest neighbors can be found as k-nearest neighbors or ϵ-nearest neighbors according to some distance measures, e.g., the Euclidean distance. The weight over the edge is set based on the distance (or similarity) between the associated data points. In the task of image segmentation, a graph for an image is constructed differently. The edges are obtained by connecting the spatially-neighboring pixels, e.g., 4-connected neighbors. Then the weight over an edge is computed as the similarity evaluated based on the appearance features of the associated pixels, and sometimes their spatial distance is also taken into consideration for the similarity evaluation.

Graph-based grouping methods aim to find the grouping, to partition the set $\mathcal{V}$ into disjoint sets, $\{\mathcal{V}_1, \mathcal{V}_2, \cdots, \mathcal{V}_m\}$, by removing some edges so that such a partition satisfies some criterion. In this paper, we assume that $\mathcal{G}$ is a connected graph because partitioning a disconnected graph can be done by partitioning each connected component (subgraph). For convenience, we present the description of graph-based grouping methods using the basic bipartitioning case that splits the graph into two subgraphs. The multi-way partitioning case can be discussed in the similar way. The two subsets in the bipartitioning case are denoted by $\mathcal{A}$ and $\mathcal{B}$. Here, $\mathcal{V} = \mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cap \mathcal{B} = \emptyset$, $\mathcal{A} \neq \emptyset$, and $\mathcal{B} \neq \emptyset$. We call the set of edges between $\mathcal{A}$ and $\mathcal{B}$ the boundary of the two subsets.

In this paper, we are interested in segmenting images using a tree structure to represent an image and study the techniques of partitioning the trees under new criteria. In implementation, we use a maximum weight spanning tree to approximate an image graph. A maximum spanning tree is a spanning tree of a weighted graph having the maximum weight. We compute it by applying the Kruskal's algorithm.

Before introducing cut criteria, we first introduce several basic terms.

Definition 1 (Cut and association)**.** *Given two disjoint sets* $\mathcal{A}$ *and* $\mathcal{B}$*, the cut between the two sets,* $\mathrm{Cut}(\mathcal{A},\mathcal{B})$*, evaluating the inter-similarity, is defined as the summation of the similarities over the boundary,*

$$\mathrm{Cut}(\mathcal{A},\mathcal{B}) = \sum_{u\in\mathcal{A},v\in\mathcal{B}} w(u,v), \tag{1}$$

where $w(u,v)$ *is the weight over the edge* (u,v)*.*

Given two sets $\mathcal{A}$ *and* $\mathcal{V}$ *with* $\mathcal{A}\subseteq\mathcal{V}$*, the association of* $\mathcal{A}$ *and* $\mathcal{V}$*,* $\mathrm{Assoc}(\mathcal{A},\mathcal{V})$*, is defined as follows,*

$$\mathrm{Assoc}(\mathcal{A},\mathcal{V}) = \sum_{u\in\mathcal{A},v\in\mathcal{V}} w(u,v). \tag{2}$$

Definition 2 (Dominant cut)**.** *Given two disjoint sets* $\mathcal{A}$ *and* $\mathcal{B}$*, the dominant cut between the two sets,* $\mathrm{DCut}(\mathcal{A},\mathcal{B})$*, is defined as the maximum one of the similarities over the boundary,*

$$\mathrm{DCut}(\mathcal{A},\mathcal{B}) = \max_{u\in\mathcal{A},v\in\mathcal{B}} w(u,v). \tag{3}$$

A. Minimum Cut

The solution to clustering by minimizing the cut value over the image graph was originally developed in [32], [45]. It has been shown in [45] that the minimum cut criterion leads to a grouping that favors cutting small sets of isolated nodes in the graph.

Let's consider an alternative criterion over a graph that minimizes the dominant cut to partition the set $\mathcal{V}$ into two disjoint nonempty subsets $\mathcal{A}^*$ and $\mathcal{B}^*$,

$$(\mathcal{A}^*,\mathcal{B}^*) = \arg\min_{\mathcal{A},\mathcal{B}} \mathrm{DCut}(\mathcal{A},\mathcal{B}), \tag{4}$$

where $\mathcal{A}\neq\emptyset, \mathcal{B}\neq\emptyset, \mathcal{A}\cup\mathcal{B}=\mathcal{V}, \mathcal{A}\cap\mathcal{B}=\emptyset$.

We present a theorem, showing minimizing the dominant cut can be reduced to minimizing the cut or dominant cut over the maximum weight spanning tree (MST) of the graph, which we call minimum tree partitioning (MTP). Without additional explanation, all the theorems have the assumption that *the MST is unique and only one edge in the MST corresponds to the smallest weight* and the proofs are given in Appendix.

Theorem 1. *The partitioning result of minimizing the dominant cut over a graph is exactly equivalent to splitting the MST* $\mathcal{T}$ *by removing the edge corresponding to the smallest weight.*

The following theorem shows a property about the minimum (dominant) cut criterion over a tree.

Theorem 2. *The minimum (dominant) cut criterion over a tree leads to cutting only one edge and results in two connected subtrees.*

According to the above theorem, partitioning the MST with the minimum cut criterion can be efficiently performed by scanning each edge in the MST, which takes linear time cost. Because of computational advantage, image segmentation by partitioning the MST of an image graph was ever studied early in [49]. Similar to the minimum cut over a graph [45], the segmentation result has a bias to cutting small regions.

B. Regularized Cut

To deal with the problem in the minimum cut criterion, various regularized cut criteria are designed by considering the characteristics within each subset. We present a summary of two representative criteria, normalized cut and average cut.

Normalized cut. The normalized cut criterion [33] takes into account the similarities between the points of each subset and the whole set, and penalizes partitions with subsets of small similarities. It is defined as follows,

$$\mathrm{NCut}(\mathcal{A},\mathcal{B}) = \frac{\mathrm{Cut}(\mathcal{A},\mathcal{B})}{\mathrm{Assoc}(\mathcal{A},\mathcal{V})} + \frac{\mathrm{Cut}(\mathcal{A},\mathcal{B})}{\mathrm{Assoc}(\mathcal{B},\mathcal{V})}. \tag{5}$$

The normalized cut criterion can be interpreted from the isoperimetric perspective [17]. A slightly-modified normalized cut criterion [13] instead replaces $\mathrm{Assoc}(\mathcal{A},\mathcal{V})$ and $\mathrm{Assoc}(\mathcal{B},\mathcal{V})$ using $\mathrm{Assoc}(\mathcal{A},\mathcal{A})$ and $\mathrm{Assoc}(\mathcal{B},\mathcal{B})$.

Average cut. The average cut criterion [34] regularizes the cut value using the size of each subset and penalizes partitions with subsets of small sizes. It is defined as the following,

$$\mathrm{ACut}(\mathcal{A},\mathcal{B}) = \frac{\mathrm{Cut}(\mathcal{A},\mathcal{B})}{|\mathcal{A}|} + \frac{\mathrm{Cut}(\mathcal{A},\mathcal{B})}{|\mathcal{B}|}, \tag{6}$$

where $|\cdot|$ is the cardinality of a subset, i.e., the number of points in the subset. Similarly, the ratio cut criterion [44] also considers the cardinalities of the two subsets, and is defined as follows,

$$\mathrm{RCut}(\mathcal{A},\mathcal{B}) = \frac{\mathrm{Cut}(\mathcal{A},\mathcal{B})}{|\mathcal{A}||\mathcal{B}|}. \tag{7}$$

It can be easily shown that the average cut criterion and the ratio cut criterion are equivalent.

Spectral relaxation. The problems of minimizing normalized cut and average cut are NP-hard. The typical solution is to adopt spectral relaxation and transform the optimization problem to an eigenvalue decomposition problem. The regularized cut criteria to image segmentation are shown theoretically to be capable of achieving good performance [34]. Although some special cases when an exact partition is achieved are analyzed [27], the solution using spectral relaxation generally is not easily analyzed due to the continuous relaxation forming an eigen-decomposition problem and the discretization stage computing the grouping result from the eigenvectors.

There are some other algorithms to optimize the normalized cut and average cut, e.g., through network flow [19] and semi-definite programming [46].

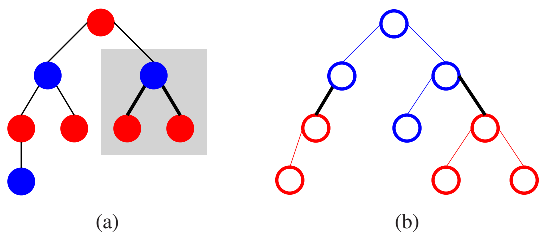


Fig. 1. (a) illustrates the super tree. Each node corresponds to a connected subtree, called super node. (b) gives an example that might form the three super nodes indicated in the gray area in (a): The top four connected blue nodes in (b) form the blue super node, the left two connected red nodes form the left red super node, and the other three connected red nodes form the right red super node.

III. Regularized Tree Partitioning

Let $\mathcal{T} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ be a tree. A partition over it is defined as separating the nodes $\mathcal{V}$ into disjoint nonempty sets, $\{\mathcal{V}_1, \cdots, \mathcal{V}_k\}$. For clarification, we present discussions mainly on the bipartition case. The two disjoint sets are denoted as $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \cup \mathcal{B} = \mathcal{V}$, $\mathcal{A} \cap \mathcal{B} = \emptyset$. In the following, we may use a subset (e.g., $\mathcal{V}_i$) of connected nodes to represent a subtree as in a tree a subset of connected nodes uniquely determines a subtree.

A. Normalized Tree Partitioning

Normalized tree partitioning (NTP) is a method of partitioning the nodes of a tree based on the normalized cut criterion. We present a property of minimizing normalized cut over a tree, which leads to an efficient optimization algorithm. Then we give a necessary condition on which minimizing normalized cut over the maximum weight spanning tree of a graph leads to the exact partition over the graph and the partition corresponding to the optimal solution of minimizing normalized cut over the graph. Last, we show another way to derive NTP.

1) Properties: We first present and prove Theorem 3. Based on it, we can design an efficient algorithm to find the optimal normalized partitioning over a tree by checking the edges in the tree one by one. We will present the algorithm in Section IV.

Theorem 3. *On a tree, the global minimum for the normalized cut criterion must correspond to two subsets: $\mathcal{A}$ and $\mathcal{B}$, and each one forms a connected tree.*

In practical applications, e.g., image segmentation in this paper, we propose to find a maximum weight spanning tree to approximate the graph, and then partition the tree to group the points. To achieve a better understanding of the proposed NTP solution to image segmentation, we analyze the necessary condition on which NTP over the MST gets the exact partition or gets the same solution to that of minimizing normalized cut over the graph, given in Theorem 4. To make the analysis clear, we assume that the MST is unique and the optimum of every partitioning criterion over the graph (and the MST) is unique without special explanation.

Theorem 4. *Suppose $(\mathcal{A}^*, \mathcal{B}^*)$ corresponds to the exact partition of the graph $\mathcal{G}$ or the optimal partition of minimizing normalized cut over $\mathcal{G}$. If normalized tree partitioning over the MST of the graph also yields $(\mathcal{A}^*, \mathcal{B}^*)$, then among the edges across $(\mathcal{A}^*, \mathcal{B}^*)$, only the light edge, the edge whose weight is the maximum of any edge across $(\mathcal{A}^*, \mathcal{B}^*)$, is contained in the MST.*

2) Interpretation: Let's introduce another similarity over a set $\mathcal{V}$, to measure the degree that the points in the set are aggregated into a single group.

Definition 3 (Aggregation)**.** *For a set $\mathcal{V}$ the aggregation is defined as follows,*

$$\begin{aligned}&\mathrm{Aggre}(\mathcal{V})\\ =& \max_{\mathcal{S} \subset \mathcal{V}, \mathcal{S} \neq \emptyset, \mathcal{S} \neq \mathcal{V}} [2\,\mathrm{DCut}(\mathcal{S}, \bar{\mathcal{S}}) + \mathrm{Aggre}(\mathcal{S}) + \mathrm{Aggre}(\bar{\mathcal{S}})],\end{aligned} \tag{8}$$

where $\mathcal{S}$ is a non-empty proper subset of $\mathcal{V}$, and $\bar{\mathcal{S}}$ is the complement of $\mathcal{S}$ with respect to $\mathcal{V}$.

Using the mathematically inductive reasoning, the aggregation can be shown to be equal to the sum of the weights of the MST of the graph corresponding to $\mathcal{V}$. Consider the partitioning criterion regularized by the Aggregation measure, called normalized dominant cut,

$$\begin{aligned}&\mathrm{NDCut}(\mathcal{A}, \mathcal{B})\\ =& \frac{\mathrm{DCut}(\mathcal{A}, \mathcal{B})}{\mathrm{Aggre}(\mathcal{A}) + \mathrm{DCut}(\mathcal{A}, \mathcal{B})} + \frac{\mathrm{DCut}(\mathcal{A}, \mathcal{B})}{\mathrm{Aggre}(\mathcal{B}) + \mathrm{DCut}(\mathcal{A}, \mathcal{B})}.\end{aligned} \tag{9}$$

We present the following lemma and theorem to show the relations between normalized dominant cut and normalized cut.

Lemma 1. *Over a tree, the solution to the normalized dominant cut criterion is exactly equivalent to the solution of NTP, i.e., the normalized cut criterion.*

Theorem 5. *Suppose the solution to minimizing normalized dominant cut over a graph leads to two subsets: $\mathcal{A}^*$ and $\mathcal{B}^*$, the solution to NTP over the MST also results in $\mathcal{A}^*$ and $\mathcal{B}^*$ iff among the edges across $(\mathcal{A}^*, \mathcal{B}^*)$, only the light edge is contained in the MST.*

B. Average Tree Partitioning

Different from normalized tree partitioning, average tree partitioning (ATP) splits the tree so that the cut regularized by the size of each subtree is minimized. Like NTP, ATP has similar properties.

First, we present a theorem that suggests an efficient algorithm to minimize average cut over a tree.

Theorem 6. *On a tree, the global minimum for the average cut (ratio cut) criterion must correspond to two subsets: $\mathcal{A}$ and $\mathcal{B}$, and each one forms a connected tree.*

The following presents the relation between the solution of ATP over the MST and the exact partition and the optimal partition of minimizing average cut over the graph. The proof is similar to that for Theorem 4.

Theorem 7. *Suppose $\mathcal{A}^*$ and $\mathcal{B}^*$ corresponds to the exact partition of the graph $\mathcal{G}$ or the optimal partition of minimizing average cut over $\mathcal{G}$. If average tree partitioning over the MST of the graph also yields $(\mathcal{A}^*, \mathcal{B}^*)$, then among the edges across $(\mathcal{A}^*, \mathcal{B}^*)$, only the light edge is contained in the MST.*

Similar to NTP, we have a new graph partitioning criterion, average dominant cut.

$$\mathrm{ADCut}(\mathcal{A}, \mathcal{B}) = \frac{\mathrm{DCut}(\mathcal{A}, \mathcal{B})}{|\mathcal{A}|} + \frac{\mathrm{DCut}(\mathcal{A}, \mathcal{B})}{|\mathcal{B}|}. \tag{10}$$

According to this criterion, we can derive ATP over the MST in another way, which is guaranteed by the below theorem.

Theorem 8. *Suppose the solution to minimizing average dominant cut over a graph leads to two subsets: $\mathcal{A}^*$ and $\mathcal{B}^*$, the solution to ATP over the MST of the graph also results in $\mathcal{A}^*$ and $\mathcal{B}^*$ iff among the edges across $(\mathcal{A}^*, \mathcal{B}^*)$, only the light edge is contained in the MST.*

C. Extension

Normalized and average tree partitioning sum up the regularized cuts from two subsets together to get an overall criterion. There might be a drawback that the larger regularized cut is diluted by the smaller one, and thus the grouping performance is deteriorated. To handle this issue, we propose to use the larger one among the regularized cuts to evaluate the quality of tree partitioning. This manner is similar to Cheeger constant [8] and can also derive the criterion used in isoperimetric graph partitioning [17].

The two criteria are formulated as follows,

$$\mathrm{MaxNCut}(\mathcal{A}, \mathcal{B}) = \max\{\frac{\mathrm{Cut}(\mathcal{A}, \mathcal{B})}{\mathrm{Assoc}(\mathcal{A}, \mathcal{V})}, \frac{\mathrm{Cut}(\mathcal{A}, \mathcal{B})}{\mathrm{Assoc}(\mathcal{B}, \mathcal{V})}\}, \tag{11}$$

$$\mathrm{MaxACut}(\mathcal{A}, \mathcal{B}) = \max\{\frac{\mathrm{Cut}(\mathcal{A}, \mathcal{B})}{|\mathcal{A}|}, \frac{\mathrm{Cut}(\mathcal{A}, \mathcal{B})}{|\mathcal{B}|}\}. \tag{12}$$

In the case that there are multiple partitions corresponding to the same $\mathrm{MaxNCut}$ (or $\mathrm{MaxACut}$) value, we remedy this problem by selecting the partition that has the smallest NCut (ACut) value. The two extensions are called maximum normalized tree partitioning (MaxNTP) and maximum average tree partitioning (MaxATP).

For MaxNTP and MaxATP, we have a property similar to Theorems 3 and 6, and then can have an efficient algorithm to find the solution.

Theorem 9. *On a tree, the global optimum for maximum normalized cut and maximum average cut must correspond to two subsets: $\mathcal{A}$ and $\mathcal{B}$, and each one forms a connected tree.*

The proof is given in the supplementary material.

Similar to NTP and ATP, we have also a property showing the necessary condition that MaxNTP and MaxATP over the MST achieves the exact partition over the graph or the same partition to MaxNTP and MaxATP over the graph.

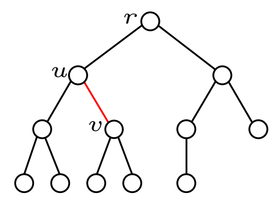


Fig. 2. A tree structure. Note that the tree we aim to partition may not be necessarily a binary tree.

Theorem 10. *Suppose $\mathcal{A}^*$ and $\mathcal{B}^*$ corresponds to the exact partition of graph $\mathcal{G}$ or the optimal partition of minimizing maximum normalized (average) cut over $\mathcal{G}$. If MaxNTP (MaxATP) over the MST of the graph also yields $(\mathcal{A}^*, \mathcal{B}^*)$, then among the edges across $(\mathcal{A}^*, \mathcal{B}^*)$, only the light edge is contained in the MST.*

We can also define the following two criteria,

$$\begin{aligned}&\mathrm{MaxNDCut}(\mathcal{A}, \mathcal{B})\\ &= \max\{\frac{\mathrm{DCut}(\mathcal{A}, \mathcal{B})}{\mathrm{Aggre}(\mathcal{A}) + \mathrm{DCut}(\mathcal{A}, \mathcal{B})}, \frac{\mathrm{DCut}(\mathcal{A}, \mathcal{B})}{\mathrm{Aggre}(\mathcal{B}) + \mathrm{DCut}(\mathcal{A}, \mathcal{B})}\},\end{aligned} \tag{13}$$

$$\begin{aligned}&\mathrm{MaxADCut}(\mathcal{A}, \mathcal{B})\\ &= \max\{\frac{\mathrm{DCut}(\mathcal{A}, \mathcal{B})}{|\mathcal{A}|}, \frac{\mathrm{DCut}(\mathcal{A}, \mathcal{B})}{|\mathcal{B}|}\}.\end{aligned} \tag{14}$$

We can have the conclusion that the above criteria over a graph can lead to maximum normalized and average cuts over the MST, as described below.

Theorem 11. *Suppose the solution to minimizing maximum normalized (average) dominant cut over a graph leads to two subsets $\mathcal{A}^*$ and $\mathcal{B}^*$, the solution to MaxNTP (MaxATP) over the MST also results in $\mathcal{A}^*$ and $\mathcal{B}^*$ iff among the edges across $(\mathcal{A}^*, \mathcal{B}^*)$, only the light edge is contained in the MST.*

IV. Algorithms

This section describes the algorithms to the proposed four tree partitioning schemes. We first present how to get the optimal bipartition and then a recursive bipartition manner to k-way partition.

A. Bipartition

Considering the tree in Figure 2, which is rooted from node r and denoted as $\mathcal{T}_r$, we can just remove edge (u, v), then the tree is partitioned into two parts: one, denoted as $\mathcal{T}_v$, is rooted from node v, and the other one, denoted as $\mathcal{T}_{r\setminus v}$ and is called the complementary subtree of $\mathcal{T}_v$, is still rooted from the original root r but excludes $\mathcal{T}_v$ and edge (u, v). For convenience, the removed edge (u, v) is used to represent such a bipartition. The (connected) tree structure only consists of $n - 1$ edges, where n is the number of the nodes. So it only takes $O(n)$ time to find the optimal edge to be split by traversing all the edges, if the associations (sizes) and cuts for all the possible partitions are pre-computed. The following shows how to efficiently compute the cuts and the associations (sizes).

Cut computation. Because there is only one edge linking two complementary subtrees, the cut for the corresponding partition is just the weight of that edge. In Figure 2, the cut value of the partition, Cut $(\mathcal{T}_{r\setminus v}, \mathcal{T}_v)$, is the similarity $w(u, v)$ of nodes u and v.

Association computation. A naive method is exhaustively calculating the associations for all possible partitions separately. This separate manner is computationally inefficient and and its time complexity is $O(n^2)$ because there are $O(n)$ possible bipartitions and it takes $O(n)$ time to calculate the association for each bipartition.

To speed up association calculation, we propose a recursive method by exhibiting the relations of the associations of different partitions. Specifically, our approach is based on two properties. The first is association complementarity. Let a_r be the association of $\mathcal{T}_r$. From Figure 2, it is obvious that $a_{r\setminus v} = a_r - a_v - 2w(u, v)$, where $a_{r\setminus v} = a_{\mathcal{T}_{r\setminus v}}$ and $a_r = a_{\mathcal{T}_r}$. Hence, it is sufficient to calculate association values for all subtrees $\mathcal{T}_v$.

The second property is overlapping of association evaluation between a subtree and its child trees. According to the definition of the association, it can be easily derived that the association of subtree $\mathcal{T}_v$ is equal to the summation of the associations of the subtrees, rooted from v's child nodes, and double cut values between v and v's child nodes. Mathematically, this overlapping can be written as a recursive formulation:

$$a_u = \begin{cases} \sum_{v \in C_u} (a_v + 2w(u, v)) & u \text{ an internal node} \\ 0 & u \text{ a leaf node,} \end{cases} \tag{15}$$

where C_u represents the set of u's child nodes. By this recursion, the associations of all the subtrees can be evaluated in a bottom-up manner from the leaves to the root.

Size computation. Average cut needs to compute the size of each subtree, the number of the nodes for each possible partition. It can also be computed in a recursive way. Let a_v in this case be the size of the subtree $\mathcal{T}_v$. The recursive formulation is written as follows,

$$a_u = \begin{cases} 1 + \sum_{v \in C_u} a_v & u \text{ an internal node} \\ 1 & u \text{ a leaf node,} \end{cases} \tag{16}$$

If the size of the tree $\mathcal{T}_r$ $a_r = n$, with n being the number of nodes, and then $a_{r\setminus v} = n - a_v$.

In summary, the tree bipartitioning algorithm is outlined in Algorithm 1.

Algorithm 1 Tree bipartitioning

1. Calculate recursively association (size) a_v for each subtree T_v according to Equation (15) or Equation (16), and association (size) $a_{r/v}$ of subtree $T_{r/v}$.
2. Traverse all the edges to find the optimal bipartition.

B. K-Way Partitioning

We study the problem of segmenting images into more partitions with a focus on k-way partitioning as k can be easily set, while other parameters, such as the ideal regularized cut value, are not intuitively and easily set as it might depend on specific images; and fixing k can help know the complexity of the segmentation results and help some applications, e.g., image compression. To be convenient, we use normalized tree partitioning as an example for the discussion. Suppose the k partitions are denoted by $\mathcal{V}_1, \cdots, \mathcal{V}_k$ and the tree is denoted by $\mathcal{V}$, the objective function can be written as follows,

$$\text{NC} = \sum_{i=1}^{k} \frac{\text{Cut}(\mathcal{V}_i, \mathcal{V} - \mathcal{V}_i)}{\text{Assoc}(\mathcal{V}_i, \mathcal{V})}. \tag{17}$$

Unlike the case $k = 2$ in which we have a linear algorithm to get the optimal solution, it is difficult to find the global optimum. A naive algorithm may check all possible k-way partitions separately in a brute force manner, which will lead to $O(n^k)$ time complexity. To our knowledge, there does not exist an exact k-way partitioning algorithm similar to the above bipartitioning algorithm.

We propose a best-first recursive bipartitioning algorithm to approximately compute k-way partitioning. The recursive procedure is outlined in Algorithm 2. The algorithm starts from two partitions obtained by splitting the tree using the above bipartitioning algorithm. Then it finds one from the two partitions based on some criterion, and splits it into two parts using the bipartitioning algorithm, resulting three partitions. Next, our algorithm does the same job over the three partitions. Our algorithm continues the splitting procedure until getting k partitions.

We conducted an experiment to compare two approximate solutions of normalized cuts: our k-way partitioning algorithm and spectral clustering [33]. The comparison of the normalized cuts values is illustrated in Figure 6 and detailed in Section VI-A2. This shows that our algorithm achieves a better approximation though both algorithms cannot get the global optimum solutions. The time complexity analysis of the recursive procedure is given in the next section. Illustrative examples are presented in Figure 5 and detailed in Section V-C.

Algorithm 2 K-way tree partitioning

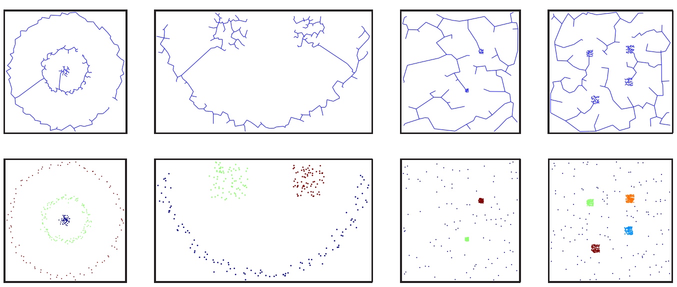
1. Bisect the input tree $\mathcal{T}$ into $\mathcal{A}_1$ and $\mathcal{A}_2$. Set the number of current partitions $p = 2$.
2. Bisect all the current subtrees $\{\mathcal{A}_i\}_{i=1}^{p}$.
3. Find the subtree $\mathcal{A}_t$ that produces the smallest normalized (average, maximum normalized, maximum average) cut value, denote its bisected subtrees $\mathcal{B}_1$ and $\mathcal{B}_2$, and let $\mathcal{A}_t = \mathcal{B}_1$ and $\mathcal{A}_{p+1} = \mathcal{B}_2$, increase p by 1.
4. Output the k-way partitions if $p = k$, otherwise go to step 2.

C. Time Complexity

In regularized tree partitioning, we evaluate the associations (sizes), using a recursive way costing $O(n)$ time in that each node is only involved once to calculate the association of its parent tree, and the optimal bipartition is found in $O(n)$ time. For k-way partitioning, the total time complexity is $O(kn)$. When applying it into general data clustering problems, we need to build a neighborhood graph for all the data points, which can be implemented in $O(n \log n)$ time using approximate nearest neighbor algorithms such as [5], [23] or direct construction [42], or build an image graph by

TABLE I
COMPARISON OF THE CRITERION AND TIME COMPLEXITY WITH OTHER METHODS. MTP = MINIMUM TREE PARTITIONING. NTP = NORMALIZED TREE PARTITIONING. ATP = AVERAGE TREE PARTITIONING. MAXNTP = MAXIMUM NORMALIZED TREE PARTITIONING. MAXATP = MAXIMUM AVERAGE TREE PARTITIONING. SC = SPECTRAL CLUSTERING [33]. GBIS = GRAPH-BASED IMAGE SEGMENTATION [14]. IGP = ISOPERIMETRIC GRAPH PARTITIONING [17]. MSNC = MULTISCALE NORMALIZED CUT [11]. NOTE: THE TIME COMPLEXITY OF SC IS FOR THE BASIC IMPLEMENTATION IN THE CASE OF A SPARSE GRAPH, AND THE COMPLEXITY OF MSNC IS FOR AN IMAGE.

	Cut criterion	Time complexity
MTP	minimum cut	$O(n(k + \log n))$
NTP	normalized cut	$O(n(k + \log n))$
ATP	average cut	$O(n(k + \log n))$
MaxNTP	maximum normalized cut	$O(n(k + \log n))$
MaxATP	maximum average cut	$O(n(k + \log n))$
SC	normalized cut	$O(n^2)$
GBIS	local criretion	$O(n \log n)$
IGP	isoperimetric cut	$O(n^2)$
MSNC	multiscale normalized cut	$O(n)$



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

Fig. 3. Illustration of normalized tree partitioning on 2D toy examples. The top row shows the MSTs, and the bottom shows the corresponding clustering results with the number of clusters prefixed as 3, 3, 3, and 5, where different colors represent different clusters.

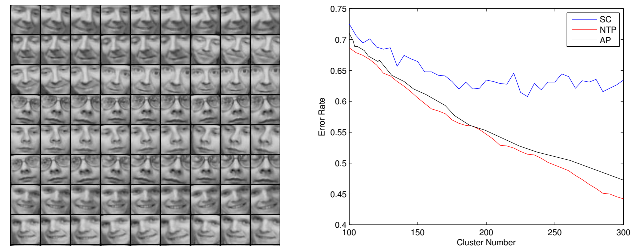
connecting spatially neighboring points, which costs $O(n)$. The maximum weight spanning tree can be achieved using Prim's or Kruskal's algorithms, which takes $O(n \log n)$ time in our sparse graph case. In summary, for general cases the time complexity is $O(n(k + \log n))$. The comparison of time complexity with representative existing graph based methods is presented in Table I.

V. ILLUSTRATIONS

In this section we use normalized tree partitioning over the MST as an example for illustration. First, we demonstrate the effectiveness of NTP on clustering complex data points and face examples. Then we present an illustration of k-way partitioning for image segmentation.

A. 2D Toy Example

We first illustrate tree partitioning for clustering performance on 2D toy examples, shown in Figure 3. We show the results of two simple examples shown in (a) and (b) and two challenging examples shown in (c) and (d), which are used in [50][1]. We construct a data graph by connecting 4 nearest neighbors. The top row in Figure 3 shows the MST, and the bottom row shows the clustering results of our approach. This example shows that our approach has the ability to cluster the complex data points as shown in Figure 3 (c) and (d).



(a) (b)

Fig. 4. (a) Sample face images. (b) Comparison with spectral clustering (SC) and affinity propagation (AP). It can be seen that the performance of NTP is better than SC and AP.

B. Face Example

We do a clustering experiment on the face dataset[2]. The data set contains 900 face images generated from the first 100 face images in the Olivetti database with simple editing. We build a sparse graph on the images by connecting each image and its 50 nearest neighbors. We vary the cluster number between 100 and 300, and compute the error rate against the ground truth (all the images, which are generated from the same original image, are considered to have the same label). The error rate is computed as the average of the error rates over all the clusters. For each cluster, we record the numbers of each of the 100 faces, $n_1, n_2, \cdots, n_{100}$, and the error rate is computed as $e = \frac{n - \max_{i \in [1,100]} n_i}{n}$, where $n = \sum_{i=1}^{100} n_i$. The comparison results with spectral clustering and affinity propagation [16] are presented in Figure 4. It can be seen that NTP performs better than other two methods. It is also worth pointing out that the error rate of NTP monotonically decreases, while that of spectral clustering oscillates, due to the instability of its discretization.

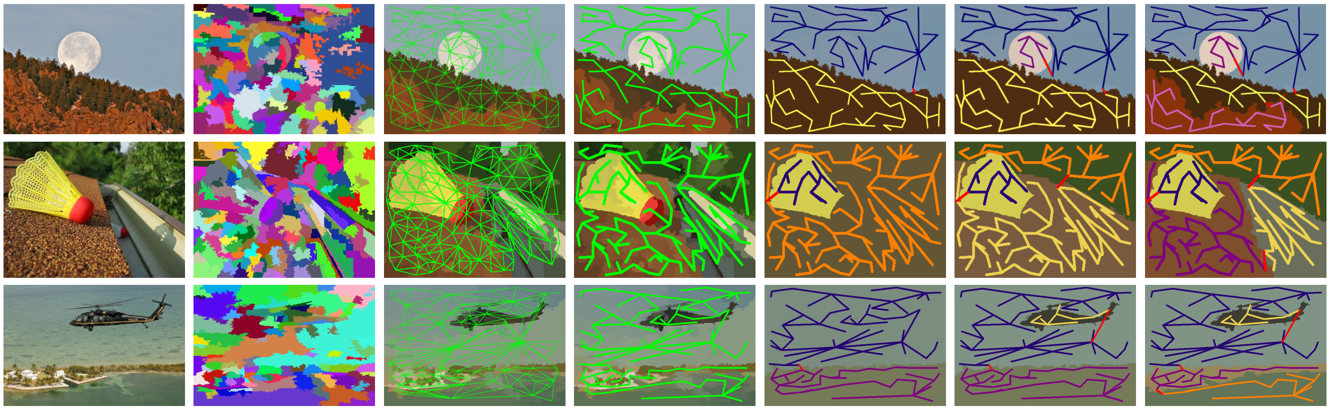
C. K-Way Partitioning

We illustrate the recursive bipartition scheme for realizing k-way partitioning for image segmentation, shown in Figure 5. The computation procedure of the weights over the edges is described in the next section. The superpixels are shown in (b), the graph on the superpixels is shown in (c). The maximum weight spanning tree approximating this graph is shown in (d). Next, the optimum bisegmentation by our method is shown (e), Last, the 3-way and 4-way partitions using the recursive scheme are shown in (f) and (g).

VI. EXPERIMENTS

We apply regularized tree partitioning approaches to image segmentation and present qualitative and quantitative comparisons on the Berkeley image segmentation data set [25]. The

[1]http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html
[2]http://www.psi.toronto.edu/affinitypropagation/Faces.JPG



(a) Image (b) Superpixels (c) Graph (d) Tree (e) Bipartition (f) 3-way partition (g) 4-way partition

Fig. 5. Illustration of recursive bipartition for k-way partitioning.

data set contains 200 training images and 100 test images with size of 481×321 or 321×481. Here we do unsupervised segmentation, and perform segmentation on all the 300 images.

The results based on tree partitioning are all obtained by segmenting the superpixels, which are generated by fragmenting the image using the watershed algorithm [38]. The graph is constructed by setting the superpixels as the nodes and connecting two superpixels iff they are spatial neighbors. The distance (d) between neighboring superpixels is evaluated as the χ^2 distance of the color histograms of two regions. An exponential function, $\exp\left(-\frac{d^2}{2\sigma^2}\right)$ with σ being the average of the distances in one image, is used to compute the weight.

By comparison, we also present the segmentation results of other widely-used grouping algorithms, including multi-class spectral clustering (SC) based on the normalized cut criterion [33], heuristic graph based image segmentation (GBIS) [14], isoperimetric graph partitioning for image segmentation (IGP) [17], mean shift image segmentation (MS) [9], multiscale normalized cut (MSNC) [11], and simple linear iterative clustering (SLIC) [1]. In addition, we also report the result of GPB-UCM [4], the state-of-the-art among contour based image segmentation approaches. For fairness, we perform the segmentation algorithms, SC and IGP, on the graph constructed from superpixels. We modify the implementation of spectral clustering[3] and isoperimetric graph partitioning[4] so that they are able to group superpixels, and get the results of SC and IGP. We run the implementation of multiscale normalized cut[5] to get the results for MSNC. We run the implementations of GBIS[6], mean shift[7], SLIC[8], and GPB-UCM[9]. The segment numbers of tree partitioning, spectral clustering are set to be the same. For GBIS, mean shift, and isoperimetric graph partitioning, we tune the parameters so that they have similar segment numbers on average.

[3] http://www.cis.upenn.edu/~jshi/software/

[4] http://www.cns.bu.edu/~lgrady/grady2006isoperimetric_code.zip

[5] http://www.seas.upenn.edu/~timothee/software/ncut_multiscale/ncut_multiscale.html

[6] http://people.cs.uchicago.edu/~pff/segment/

[7] http://coewww.rutgers.edu/riul/research/code/EDISON/index.html

[8] http://ivrg.epfl.ch/supplementary_material/RK_SLICSuperpixels/

[9] http://vision.caltech.edu/~mmaire/software/grouping.zip

A. Quantitative Comparison

The quantitative comparison is based on four criteria against the human annotations: probabilistic rand index (PRI) [35], variation of information (VoI) [26], global consistency error (GCE) [25], and boundary displacement error (BDE) [15]. The PRI score counts the number of pairs of pixels whose labels are consistent between the segmentation and the ground truth. The score is averaged over multiple ground truth segmentations to take scale variation into consideration in human perception. The VoI score defines the distance between two segmentations as the average conditional entropy of one segmentation given the other, and thus roughly measures the amount of randomness in one segmentation that cannot be explained by the other. The GCE score measures the extent to which one segmentation can be viewed as a refinement of the other. Segmentations which are related in this manner are considered to be consistent, because they could represent the same natural image segmented at different scales. The BDE score measures the average displacement error of boundary pixels between two segmented images.

The segmentation is viewed better if PRI is larger or the other three are smaller. It is reported in [47] that PRI is more correlated with human hand segmentations. It also should be pointed out that GCE favors over-segmentation and under-segmentation [31], which results in that the highest score is achieved when each pixel forms a segment or all pixels form a single segment.

1) Maximum and Random Spanning Trees: We conduct the image segmentation over the maximum spanning tree (MST) of the original graph using the proposed tree partitioning approaches. Here we provide empirical evidences to show that MST is a better choice than random spanning tree (RST). We present the performance comparison based on the MST and RST. The quantitative comparison is provided in Table II. We randomly generate 10 RSTs and report the average performance over them. As we can see, the performances over MST consistently outperform those over RST in terms of all the four criteria.

2) Normalized Tree Partitioning and Spectral Clustering: We conduct an experiment to show the superiority of nor-

TABLE II
QUANTITATIVE COMPARISON OF DIFFERENT MST AND RST. THE BEST SCORES ARE HIGHLIGHTED IN BOLD FONTS. NTP = NORMALIZED TREE PARTITIONING OVER THE MST. ATP = AVERAGE TREE PARTITIONING OVER THE MST. MAXNTP = MAXIMUM NORMALIZED TREE PARTITIONING OVER THE MST. MAXATP = MAXIMUM AVERAGE TREE PARTITIONING OVER THE MST. RSTNTP = NTP ON THE RST. RSTATP = ATP ON THE RST. RSTMAXNTP = MAXNTP ON THE RST. RSTMAXATP = MAXATP ON THE RST.

	NTP	ATP	MaxNTP	MaxATP	RSTNTP	RSTATP	RSTMaxNTP	RSTMaxATP
PRI	0.7984	**0.8039**	0.7963	0.8020	0.7806	0.7869	0.7794	0.7853
VoI	2.113	**2.021**	2.142	2.048	2.357	2.270	2.388	2.295
GCE	0.2171	0.2066	0.2194	**0.2039**	0.2621	0.2400	0.2667	0.2425
BDE	**13.58**	13.77	13.85	13.65	15.28	14.46	15.41	14.52

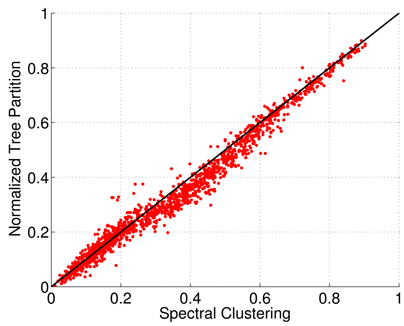


Fig. 6. Comparison of k-way normalized cut values of spectral clustering and normalized tree partitioning. Each point is formed by the normalized cut values computed from spectral clustering and normalized tree partitioning for each image in the Berkeley image data set. It can be seen that normalized tree partitioning performs better.

malized tree partitioning over spectral clustering in terms of the k-way normalized cut value. The value is computed by $\sum_{i=1}^{k} \frac{\text{Cut}(\mathcal{V}_i,\mathcal{V}-\mathcal{V}_i)}{\text{Assoc}(\mathcal{V}_i)}$. Here $\mathcal{V}$ is the whole set of superpixels, $\mathcal{V}_i$ corresponds to each partition. For each image, we consider all the numbers of segmentation that are provided in the ground-truth. The illustration is shown in Figure 6, where each point corresponds to the normalized cut values computed from spectral clustering and normalized tree partitioning for one image. We can see that most points lie in the bottom-left area, which indicates that for most images normalized tree partitioning gets smaller normalized cut values than spectral clustering. Specifically, there are 246 out of 300 (around 82%) images whose normalized cut values of our approach are smaller than spectral clustering. As a result, normalized tree partitioning get better solutions than spectral clustering.

3) Comparison with State-of-the-Art Approaches: The comparisons with state-of-the-art approaches are shown in Table III. We first discuss the results from grouping algorithms. SLIC gets the worst performance, which is reasonable as it is an approach to image oversegmentation. In terms of PRI, GCE and BDE, the proposed four tree partitioning approaches perform better than Spectral clustering (SC), isoperimetric graph partitioning (IGP), and multiscale normalized cut (MSNC). In terms of PRI, VoI, and BDE, NTP, ATP and MaxNTP outperform GBIS. In terms of PRI and VDE, NTP and MaxATP is better than MS. In terms of PRI and VoI, ATP is better than MS.

The superiority over SC, IGP, and MSNC is because our method can obtain a better solution by introducing the tree structure, while spectral relaxation adopted in SC, IGP, and MSNC suffers from the two approximation steps, relaxing the discrete values to continuous values and discretizing the continuous solutions to the discrete ones. The superiority over GBIS comes from the regularization considering both inter-similarities and self-similarities (NTP) or self-size (ATP) of clusters, while GBIS only utilizes the local similarity criterion and has no ability to measure the self-similarity of a cluster. The superiority over MS comes from the same reason. Experience shows that PRI seems to be more correlated with human segmentation in term of visual perception.

The running time of all the methods is reported in the last but one row of Table III. We can see that the running time (RT in seconds) of tree partitioning approaches is much less than SC, MS, IGP, and MSNC. It should be noted that the running time for tree partitioning is recorded for the whole segmentation process including both preprocessing, graph construction, MST computing and tree partitioning. SC, IGP, and MSNC contain a time-consuming eigenvalue decomposition step and MS is an iterative algorithm that essentially contains time-consuming matrix-vector multiplication operations. GBIS costs less time than tree partitioning approaches. It is reasonable because GBIS performs a greedy algorithm while tree partitioning approaches need to compute optimal solutions.

The memory cost of all the approaches for segmenting an image is reported in the last row of Table III. One can see that MSNC and GPB-UCM take the largest memory cost and others take small cost. GBIS and SLIC require the least memory because they only use the color feature (e.g., RGB or LUV) to represent the pixels. MS uses a sparse similarity matrix over pixels and it is an iterative algorithm, thus taking a little more memory cost. Other methods, MTP, NTP, ATP, MaxNTP, MaxATP, SC, IGP, need to store the regional features (i.e., color histogram) and hence take a little more memory cost, though the tree and graph structures over superpixels use very small cost. MSNC computes the normalized cut simultaneously across multiple scales and therefore needs more memory. GPB-UCM consumes the most memory because the local features in several channels used are large and the affinity matrix is also large.

We report the segmentation results from GPB-UCM [4], the best contour based segmentation approach. As shown in Table III, except VoI, our approach gets the similar (slightly worse) performance over other three criteria compared with GPB-UCM. This is understandable because our approach is a general clustering algorithm and GPB-UCM is specially designed for image segmentation. The comparable performance

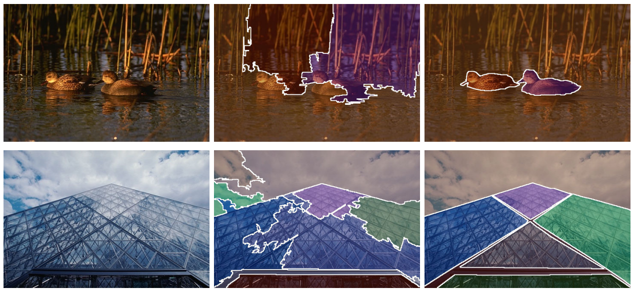


Fig. 8. Failure examples. (a) input image, (b) segmentation results of ATP, and (c) ground truth annotation.

obtained from our approach shows the power of our approach. Notably, our approach is much more efficient than GPB-UCM and around 170 times faster than GPB-UCM. Considering both segmentation quality and efficiency, our approach is quite competitive.

B. Qualitative Comparison

The qualitative results from clustering algorithms, including minimum tree partitioning, average tree partitioning and maximum average tree partitioning, spectral clustering, graph based image segmentation, mean shift, isoperimetric graph partitioning, and multiscale normalized cut are shown in Figure 7. Here, we do not show the results from (maximum) normalized tree partitioning as they are similar to and only a little worse than those of (Max)ATP.

In the challenging image in the second row of Figure 7 (a), the color of the leopard is very close to the background. In addition, the repeated patterns on the leopard's body also make the segmentation algorithm difficult to cut out its whole body. For this case ATP successfully cuts out the leopard while SC and MSNC both fail. Another example is the fourth row of Figure 7 (a), SC, GBIS, MS, IGP, and MSNC cannot cut out the body of the dog, while MaxATP can distinguish the dog from the background. This superiority of MaxATP comes from the tree structure in which the more accurate solution can be found. In the last but one row of Figure 7, the color of the wolf's body is very close to the background. GBIS and MS all fail to cut out the wolf. GBIS only cuts out the eye and nose of the wolf and MS only cuts out small regions on its body. ATP and MaxATP, however, which take the size of a cluster into consideration, successfully cut out the whole body of the wolf.

There are some cases in which regularized tree partitioning fails to achieve satisfactory segmentation results. Some failure results of the ATP approach are shown in Figure 8. The images are highly textured (e.g., the glass in the second row). It is difficult to generate appealing segmentation results to consider only the color information. By combining other cues, such as edge, texture, our approaches have potentials to get even better results in the failure cases.

VII. Conclusions and Future Work

In this paper, we propose regularized tree partitioning approaches and show the powerfulness in the application to image segmentation. We have presented normalized and average tree partitioning and two extensions, analyzed the property that results in an efficient algorithm. Moreover, we also have given the necessary condition on which our approaches over the MST can get the exact partition over the graph. Furthermore, we have derived our approaches from novel graph partitioning criteria and given the sufficient and necessary condition on which our approaches get the optimal bipartition. Experimental results of image segmentation over the Berkeley image data set demonstrate the effectiveness of the proposed approaches.

Discussions and future work. (1) The theoretical analysis presented in this paper is based on the assumptions: the maximum weight spanning tree is unique and there is only one edge in this tree corresponding to the smallest weight. Our experiments show that two assumptions often hold in the real-world images. But the analysis remains unclear if the two assumptions do not hold. It is worth investigating the theory without the two assumptions. (2) We present an approximate solution to k-way partitioning. As future works, we will study how the solution approximates the exact k-way partitioning and if there is any scheme automatically determining k.

Acknowledgements

This work was partially supported by 973 Program (2013CB329503), RGC-GRF 618711, RGC-GRF 618510, and RGC/NSFC N_HKUST607/11.

Appendix

A. Proof for Theorem 1

Proof. Suppose that $\mathcal{A}^*$ and $\mathcal{B}^*$ correspond to the optimal solution of Equation (4). Let $(u^*, v^*) = \arg\max_{u\in\mathcal{A}^*, v\in\mathcal{B}^*} w(u, v)$.

First, edge (u^*, v^*) must appear in the MST $\mathcal{T}$. Otherwise, we can build a tree $\widetilde{\mathcal{T}}$ with a larger weight than the weight of $\mathcal{T}$ in the below process. Here the weight of a tree is defined as the summation of the weights over all the edges in the tree. If adding (u^*, v^*) to the tree $\mathcal{T}$, it is obvious that there exists a loop that includes two edges across $\mathcal{A}^*$ and $\mathcal{B}^*$, with one being (u^*, v^*) and another edge e. By the definition, $w(u^*, v^*) > w(e)$, where $w(e)$ is the weight over edge e. Then we can remove edge e to break the loop to get another tree $\widetilde{\mathcal{T}}$ whose weight is larger than that of T.

Second, there exists only a single edge in the MST that is across the two subsets, $\mathcal{A}^*$ and $\mathcal{B}^*$. Otherwise, we can cut another edge in the MST to get a partition in the below process so that the objective value is smaller. Suppose there is another edge e in the MST across $\mathcal{A}^*$ and $\mathcal{B}^*$. By the definition, $w(u^*, v^*) > w(e)$. Then, we can cut edge e in the MST to get two subsets $\widetilde{\mathcal{A}}^*$ and $\widetilde{\mathcal{B}}^*$, and in this case, $w(e) = \max_{u\in\widetilde{\mathcal{A}}^*, v\in\widetilde{\mathcal{B}}^*} w(u, v)$ because e lies in the MST. Thus, $\max_{u\in\widetilde{\mathcal{A}}^*, v\in\widetilde{\mathcal{B}}^*} w(u, v) < \max_{u\in\mathcal{A}^*, v\in\mathcal{B}^*} w(u, v)$.

Finally, edge (u^*, v^*) has the smallest weight in the MST. Otherwise, we can find another partitioning with a smaller dominant cut. Suppose in the MST weight $w(e)$ over edge e is smaller than $w(u^*, v^*)$ and cutting edge e leads to two subsets $\widetilde{\mathcal{A}}^*$ and $\widetilde{\mathcal{B}}^*$. According to the

TABLE III
QUANTITATIVE COMPARISON WITH STATE-OF-THE-ART METHODS. THE BEST SCORES ARE EMPHASIZED IN BOLD FONTS. MTP = MINIMUM TREE PARTITIONING. NTP = NORMALIZED TREE PARTITIONING. ATP = AVERAGE TREE PARTITIONING. MAXATP = MAXIMUM AVERAGE TREE PARTITIONING. SC = SPECTRAL CLUSTERING [33]. IGP = ISOPERIMETRIC GRAPH PARTITIONING [17]. GBIS = GRAPH-BASED IMAGE SEGMENTATION [14]. MS = MEAN SHIFT [9]. MSNC = MULTISCALE NORMALIZED CUT [11]. SLIC = SIMPLE LINEAR ITERATIVE CLUSTERING [1]. GPB-UCM = CONTOUR BASED SEGEMENTATION [4].

	Grouping											Contour
	MTP	NTP	ATP	MaxNTP	MaxATP	SC	IGP	GBIS	MSNC	MS	SLIC	GPB-UCM
PRI	0.7442	0.7984	**0.8039**	0.7963	0.8020	0.7911	0.7896	0.7753	0.7614	0.7853	0.7287	**0.8183**
VoI	**1.840**	2.113	2.021	2.142	2.048	2.056	1.992	2.448	2.665	2.033	2.935	**1.547**
GCE	0.1953	0.2171	0.2066	0.2194	0.2039	0.2510	0.2343	0.2037	0.2610	**0.1817**	0.2738	**0.1911**
BDE	19.88	**13.58**	13.77	13.85	13.65	14.01	13.85	14.49	14.01	13.71	18.93	**13.04**
RT	**1.219**	1.252	1.230	1.246	1.227	3.460	21.74	**0.6551**	58.73	121.0	0.5223	207.6
Memory (MB)	6	6	6	6	6	7	6	**1**	181	6	**1**	707

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k)

Fig. 7. Visual comparisons of image segmentation on the Berkeley dataset. (a) the original image, (b) minimum tree partitioning, (c) average tree partitioning, (d) maximum average tree partitioning, (e) spectral clustering [33], (f) graph based image segmentation [14], (g) mean shift [9], (h) isoperimetric graph partitioning [17], (i) multiscale normalized cut [11], (j) GPB-UCM [4], and (k) simple linear iterative clustering (SLIC) [1].

definition of MST, $w(e) = \max_{u\in\widetilde{\mathcal{A}}^*, v\in\widetilde{\mathcal{B}}^*} w(u,v)$. Thus, $\max_{u\in\widetilde{\mathcal{A}}^*, v\in\widetilde{\mathcal{B}}^*} w(u,v) < \max_{u\in\mathcal{A}^*, v\in\mathcal{B}^*} w(u,v)$. This is in contradiction with that $\mathcal{A}^*$ and $\mathcal{B}^*$ is the optimal solution of Equation (4)

Consequently, the statement holds. □

B. Proof for Theorem 2

Proof. The theorem could be proved by contradiction. If in the optimal partition more than one edges, e.g., $\{e_1, \cdots, e_k\}$, are cut, the dominant cut would correspond to the edge with the greatest weight, which is assumed to be edge e_1 without loss of generality. Consider a new partition that is formed by cutting one edge among $\{e_2, \cdots, e_k\}$. It can be easily validated that the dominant cut over such a new partition is smaller than that in the optimal partition. This is in contradiction with the definition of the optimal partition.

The theorem regarding to the minimum cut criterion can be proved in a similar way. □

C. Proof for Theorem 3

Proof. We prove this theorem by contradiction. Suppose there exists an optimum, where m (> 1) edges, $\{(u_i, v_i)\}_{i=1}^m$, $u_i \in \mathcal{A}$, $v_i \in \mathcal{B}$, are removed, and this leads to $m+1$ connected subtrees $\{\mathcal{V}_j\}_{j=0}^m$. Denote $\beta_{\mathcal{V}_j} = \text{Assoc}(\mathcal{V}_j, \mathcal{V})$. Thus, $\text{Assoc}(\mathcal{A}, \mathcal{V}) = \sum_{\mathcal{V}_j\subset\mathcal{A}} \beta_{\mathcal{V}_j}$, $\text{Assoc}(\mathcal{B}, \mathcal{V}) = \sum_{\mathcal{V}_j\subset\mathcal{B}} \beta_{\mathcal{V}_j}$. Then the normalized cut value is written as

$$\begin{aligned}\text{NCut} &= \frac{\sum_{i=1}^m w(u_i, v_i)}{\sum_{\mathcal{V}_j\subset\mathcal{A}} \beta_{\mathcal{V}_j}} + \frac{\sum_{i=1}^m w(u_i, v_i)}{\sum_{\mathcal{V}_j\subset\mathcal{B}} \beta_{\mathcal{V}_j}} \\ &= \frac{w_{\mathcal{T}} \sum_{i=1}^m w(u_i, v_i)}{(\sum_{\mathcal{V}_j\subset\mathcal{A}} \beta_{\mathcal{V}_j})(\sum_{\mathcal{V}_j\subset\mathcal{B}} \beta_{\mathcal{V}_j})},\end{aligned} \tag{18}$$

where $w_{\mathcal{T}} \equiv \sum_{\mathcal{V}_j\subset\mathcal{A}} \beta_{\mathcal{V}_j} + \sum_{\mathcal{V}_j\subset\mathcal{B}} \beta_{\mathcal{V}_j} \equiv \text{Assoc}(\mathcal{A}\cup\mathcal{B}, \mathcal{A}\cup\mathcal{B}) \equiv \text{Assoc}(\mathcal{V}, \mathcal{V})$ by definition.

Let's consider the m partitions, $\{(\mathcal{A}_i, \mathcal{B}_i)\}_{i=1}^m$, where each partition $(\mathcal{A}_i, \mathcal{B}_i)$ is formed by splitting a single edge (u_i, v_i) in the original tree. The normalized cut value of a partition

$(\mathcal{A}_i, \mathcal{B}_i)$ is written as

$$\begin{aligned}\mathrm{NCut}_i &= \frac{w(u_i, v_i)}{\sum_{\mathcal{V}_j \subset \mathcal{A}_i} \beta_{\mathcal{V}_j}} + \frac{w(u_i, v_i)}{\sum_{\mathcal{V}_j \subset \mathcal{B}_i} \beta_{\mathcal{V}_j}} \\ &= \frac{w_{\mathcal{T}} w(u_i, v_i)}{(\sum_{\mathcal{V}_j \subset \mathcal{A}_i} \beta_{\mathcal{V}_j})(\sum_{\mathcal{V}_j \subset \mathcal{B}_i} \beta_{\mathcal{V}_j})}, \end{aligned} \quad (19)$$

where $w_{\mathcal{T}}$ is the same as the definition in Eqn. (18).

Subtrees $\{\mathcal{V}_j\}_{j=0}^m$ and edges $\{(u_i, v_i)\}_{i=1}^m$ can be viewed as a super tree $\widehat{\mathcal{T}}$ with subtrees as super nodes connected by edges $\{(u_i, v_i)\}_{i=1}^m$, which is shown in Figure 1. Then the following two statements hold: (1) $\mathcal{A}$ and $\mathcal{B}$ correspond to the subsets of super nodes with odd depths (blue nodes in Figure 1) and even depths (red nodes in Figure 1), respectively; (2) $(\mathcal{A}_i, \mathcal{B}_i)$ can be viewed as a partition of the super tree $\widehat{\mathcal{T}}$ by removing edge (u_i, v_i).

From the above two statements, the following inequality holds,

$$\sum_{i=1}^m (\sum_{\mathcal{V}_j \subset \mathcal{A}_i} \beta_{\mathcal{V}_j})(\sum_{\mathcal{V}_j \subset \mathcal{B}_i} \beta_{\mathcal{V}_j}) > (\sum_{\mathcal{V}_j \subset \mathcal{A}} \beta_{\mathcal{V}_j})(\sum_{\mathcal{V}_j \subset \mathcal{B}} \beta_{\mathcal{V}_j}). \quad (20)$$

This inequality can be justified because (1) the expansion of the right hand leads to the summation of all $\beta_{\mathcal{V}_o}$ and $\beta_{\mathcal{V}_e}$ with $\mathcal{V}_o$ and $\mathcal{V}_e$ being the super nodes of an odd depth and an even depth, and (2) any $\beta_{\mathcal{V}_o}$ or $\beta_{\mathcal{V}_e}$ must appear in the expansion of the left hand.

Then, we have the following inequality, by denoting $\overline{\mathrm{NCut}}_i = \frac{\mathrm{NCut}_i}{w_{\mathcal{T}}}$.

$$\begin{aligned} & \min\{\overline{\mathrm{NCut}}_i\}_{i=1}^m && (21) \\ \leqslant & \frac{\sum_{i=1}^m w(u_i, v_i)}{\sum_{i=1}^m (\sum_{\mathcal{V}_j \subset \mathcal{A}_i} \beta_{\mathcal{V}_j})(\sum_{\mathcal{V}_j \subset \mathcal{B}_i} \beta_{\mathcal{V}_j})} && (22) \\ < & \frac{\sum_{i=1}^m w(u_i, v_i)}{(\sum_{\mathcal{V}_j \subset \mathcal{A}} \beta_{\mathcal{V}_j})(\sum_{\mathcal{V}_j \subset \mathcal{B}} \beta_{\mathcal{V}_j})}. && (23) \end{aligned}$$

The inequality from Equation (21) to Equation (22) can be easily justified by the generalization of the fact: If $\frac{a_1}{b_1} \leqslant \frac{a_2}{b2}$ with a_1, b_1, a_2, and b_2 being positive, $\frac{a_1}{b_1} \leqslant \frac{a_1+a_2}{b_1+b_2}$ holds. The whole inequality means that at least one partition $(\mathcal{A}_i, \mathcal{B}_i)$ has a smaller normalized cut value than $(\mathcal{A}, \mathcal{B})$, and this is in contradiction with the assumption.

Consequently, the theorem holds. □

D. *Proof for Theorem 4*

Proof. First, it can be shown that the light edge $(u^*, v^*) = \arg\max_{u \in \mathcal{A}^*, v \in \mathcal{B}^*} w(u, v)$ must be contained in the MST according to the definition of MST. Second, NTP gets the partition $(\mathcal{A}^*, \mathcal{B}^*)$ and Theorem 3 indicates that NTP cuts only one edge in the tree. Third, if there is another edge e across $(\mathcal{A}^*, \mathcal{B}^*)$ contained in the MST, this means that cutting only one edge across $(\mathcal{A}^*, \mathcal{B}^*)$ cannot lead to the partition $(\mathcal{A}^*, \mathcal{B}^*)$. Hence, only the light edge across $(\mathcal{A}^*, \mathcal{B}^*)$ is contained in the MST.

Consequently, the theorem holds. □

E. *Proof for Theorem 5*

Proof. We use S1 to denote the statement that the solution to minimizing normalized cut over the MST also results in $\mathcal{A}^*$ and $\mathcal{B}^*$, and S2 to denote the statement among the edges across $(\mathcal{A}^*, \mathcal{B}^*)$, only the light edge is contained in the MST. Using the proof of Theorem 4, we can justify S1 $\rightarrow$ S2.

S2 $\rightarrow$ S1 can be justified by contradiction. On the one hand, if S2 holds, then we can split one edge in the MST to produce the partition $(\mathcal{A}^*, \mathcal{B}^*)$. On the other hand, assume that the solution to minimizing NCut over the MST leads to another partition $\widetilde{\mathcal{A}}^*$ and $\widetilde{\mathcal{B}}^*$, which leads to a smaller NCut value. It can be shown that $\mathrm{DCut}(\widetilde{\mathcal{A}}^*, \widetilde{\mathcal{B}}^*)$ over the graph is the weight over the edge in the MST that is cut to form the partition $(\widetilde{\mathcal{A}}^*, \widetilde{\mathcal{B}}^*)$. Then, the partition $(\widetilde{\mathcal{A}}^*, \widetilde{\mathcal{B}}^*)$ produces a smaller NDCut value over the graph. This is in contradiction with that $\mathcal{A}^*$ and $\mathcal{B}^*$ correspond to the optimal partition over the graph.

Consequently, the theorem holds. □

F. *Proof for Theorem 6*

Proof. It has be shown before that ratio cut is equivalent to average cut. Hence, we consider this theorem only in terms of ratio cut (RCut), and prove it by contradiction.

Suppose there exists an optimum, where m (> 1) edges, $\{(u_i, v_i)\}_{i=1}^m$, $u_i \in \mathcal{A}$, $v_i \in \mathcal{B}$, are removed, and this leads to $m+1$ connected subtrees $\{\mathcal{V}_j\}_{j=0}^m$. Then the ratio cut value can be written as

$$\mathrm{RCut} = \frac{\sum_{i=1}^m w(u_i, v_i)}{|\mathcal{A}||\mathcal{B}|}. \quad (24)$$

Consider the m possible partitions, $\{(\mathcal{A}_i, \mathcal{B}_i)\}_{i=1}^m$, corresponding to splitting one single edge from $\{(u_i, v_i)\}_{i=1}^m$. The ratio cut value of $(\mathcal{A}_i, \mathcal{B}_i)$ is written as

$$\mathrm{RCut}_i = \frac{w(u_i, v_i)}{|\mathcal{A}_i||\mathcal{B}_i|}. \quad (25)$$

The following inequality holds

$$\begin{aligned} & \min\{\mathrm{RCut}_i\}_{i=1}^m && (26) \\ \leqslant & \frac{\sum_{i=1}^m w(u_i, v_i)}{\sum_i |\mathcal{A}_i||\mathcal{B}_i|} && (27) \\ < & \frac{\sum_{i=1}^m w(u_i, v_i)}{|\mathcal{A}||\mathcal{B}|}. && (28) \end{aligned}$$

The inequality from Equation (26) to Equation (27) can easily be validated. The inequality from Equation (27) to Equation (28) can be proved similarly from Equation (22) to Equation (23).

Consequently, the theorem holds. □

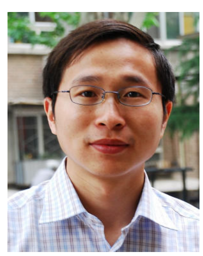
References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, 2012.

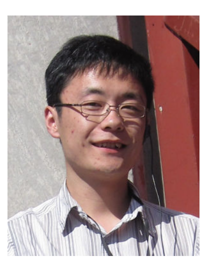
[2] R. Adams and L. Bischof. Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(6):641–647, 1994.

[3] C. Allène, J.-Y. Audibert, M. Couprie, and R. Keriven. Some links between extremum spanning forests, watersheds and min-cuts. *Image Vision Comput.*, 28(10):1460–1471, 2010.

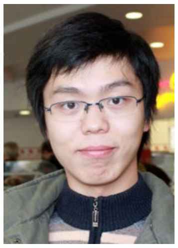
[4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011.
[5] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, 1998.
[6] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, pages 105–112, 2001.
[7] C. R. Brice and C. Fennema. Scene analysis using regions. *AI*, 1(3-4):205–226, 1970.
[8] F. R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, 1997.
[9] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.
[10] C. Couprie, L. J. Grady, L. Najman, and H. Talbot. Power watershed: A unifying graph-based optimization framework. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(7):1384–1399, 2011.
[11] T. Cour, F. Bénézit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR (2)*, pages 1124–1131, 2005.
[12] I. J. Cox, S. B. Rao, and Y. Zhong. Ratio regions: A technique for image segmentation. In *ICPR*, pages 557–564, 1996.
[13] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *ICDM*, pages 107–114, 2001.
[14] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.
[15] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí. Yet another survey on image segmentation: Region and boundary information integration. In *ECCV (3)*, pages 408–422, 2002.
[16] B. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.
[17] L. Grady and E. L. Schwartz. Isoperimetric graph partitioning for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):469–475, 2006.
[18] L. Guigues, J. P. Cocquerez, and H. L. Men. Scale-sets image analysis. *International Journal of Computer Vision*, 68(3):289–317, 2006.
[19] D. S. Hochbaum. Polynomial time algorithms for ratio regions and a variant of normalized cut. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(5):889–898, 2010.
[20] S. L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *JACM*, 23(2):368–388, April 1976.
[21] S. L. Horowitz and T. Pavlidis. A graph-theoretic approach to picture processing. 7(2):282–291, April 1978.
[22] I. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(10):1075–1088, 2001.
[23] Y. Jia, J. Wang, G. Zeng, H. Zha, and X.-S. Hua. Optimizing kd-trees for scalable visual descriptor indexing. In *CVPR*, pages 3392–3399, 2010.
[24] Y. Jia, J. Wang, C. Zhang, and X.-S. Hua. Augmented tree partitioning for interactive image segmentation. In *ICIP*, pages 2292–2295, 2008.
[25] D. R. 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. In *ICCV*, pages 416–425, 2001.
[26] M. Meila. Comparing clusterings: an axiomatic view. In *ICML*, pages 577–584, 2005.
[27] M. Meila and J. Shi. A random walks view of spectral segmentation. In *AISTATS*, 2001.
[28] O. Monga. An optimal region growing algorithm for image segmentation. *PRAI*, 1(4):351–375, December 1987.
[29] L. Najman. On the equivalence between hierarchical segmentations and ultrametric watersheds. *Journal of Mathematical Imaging and Vision*, 40(3):231–247, 2011.
[30] L. Najman and M. Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(12):1163–1173, 1996.
[31] M. Polak, H. Zhang, and M. H. Pi. An evaluation metric for image segmentation of multiple objects. *Image Vision Comput.*, 27(8):1223–1227, 2009.
[32] A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11(3):430–452, 1990.
[33] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.
[34] P. Soundararajan and S. Sarkar. Investigation of measures for grouping by graph partitioning. In *CVPR (1)*, pages 239–246, 2001.
[35] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):929–944, 2007.
[36] R. Urquhart. Graph theoretical clustering based on limited neighborhood sets. *Pattern Recognition*, 15(3):173–187, 1982.
[37] O. Veksler. Image segmentation by nested cuts. In *CVPR*, pages 339–344, 2000.
[38] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(6):583–598, June 1991.
[39] J. Wang. *Graph Based Image Segmentation: A Modern Approach*. VDM Verlag Dr. Müller, Dec 2008.
[40] J. Wang, Y. Jia, X.-S. Hua, C. Zhang, and L. Quan. Normalized tree partitioning for image segmentation. In *CVPR*, 2008.
[41] J. Wang, F. Wang, C. Zhang, H. C. Shen, and L. Quan. Linear neighborhood propagation and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(9):1600–1615, 2009.
[42] J. Wang, J. Wang, G. Zeng, Z. Tu, R. Gan, and S. Li. Scalable k-nn graph construction for visual descriptors. In *CVPR*, pages 1106–1113, 2012.
[43] S. Wang and J. M. Siskind. Image segmentation with ratio cut. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(6):675–690, 2003.
[44] Y.-C. A. Wei and C.-K. Cheng. Ratio cut partitioning for hierarchical designs. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 10(7):911–921, 1991.
[45] Z. Wu and R. M. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11):1101–1113, 1993.
[46] E. Xing and M. Jordan. On semidefinite relaxation for normalized k-cut and connections to spectral clustering. Technical report, 2003.
[47] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225, 2008.
[48] S. X. Yu and J. Shi. Multiclass spectral clustering. In *ICCV*, pages 313–319, 2003.
[49] C. Zahn. Graph theoretic methods for detecting and describing gestalt clusters. *IEEE Trans. Computers*, 20.
[50] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *NIPS*, 2004.
[51] X. Zhu. Semi-supervied Learning Literature Survey. *Computer Sciences Technical Report, 1530, University of Wisconsin-Madison*, 2006.



Jingdong Wang received the BSc and MSc degrees in Automation from Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and the PhD degree in Computer Science from the Hong Kong University of Science and Technology, Hong Kong, in 2007. He is currently a lead researcher at the Visual Computing Group, Microsoft Research Asia. His areas of interest include computer vision, machine learning, and multimedia search. At present, he is mainly working on the Big Media project, including large-scale indexing and clustering, and Web image search and mining. He is an editorial board member of Multimedia Tools and Applications.



Huaizu Jiang is currently working as a research assistant at Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. Before that, he received his BS and MS degrees from Xi'an Jiaotong University, China, in 2005 and 2009, respectively. He is interested in how to teach an intelligent machine to understand the visual scene like a human. Specifically, his research interests include saliency detection, (3D) scene understanding, and large-scale visual recognition.



Yangqing Jia received his BS and MS degrees in control science and engineering in Tsinghua University, China in 2006 and 2009, respectively, and the PhD degree in computer science from the University of California, Berkeley in 2014. He is currently a research scientist at Google Research. His research interests include object recognition, deep learning, parallel computation, and visual cognition.

Xian-Sheng Hua has been a senior researcher in Microsoft Research Redmond since 2013, working on Web-scale image and video understanding and search, as well as related applications.

Before that, he was a Principal Research and Development Lead in Multimedia Search for the Microsoft search engine, Bing, since 2011. He led a team that designed and delivered leading-edge media understanding, indexing and searching features.

He joined Microsoft Research Asia in 2001 as a researcher. Since then, his research interests have been in the areas of multimedia search, advertising, understanding, and mining, as well as pattern recognition and machine learning. He has authored or co-authored more than 200 research papers in these areas and has filed more than 90 patents.

Dr Hua received his BS in 1996 and PhD in applied mathematics in 2001 from Peking University, Beijing. He served or is now serving as an associate editor of IEEE Transactions on Multimedia, an associate editor of ACM Transactions on Intelligent Systems and Technology, an editorial board member of Advances in Multimedia and Multimedia Tools and Applications, and an editor of Scholarpedia (multimedia category). He was vice program chair; workshop organizer; senior TPC member and area chair; and demonstration, tutorial, and special session chairs and PC member of many more international conferences. He serves as a program co-chair for IEEE ICME 2013, ACM Multimedia 2012, and IEEE ICME 2012.

He was honored as one of the recipients of the prestigious 2008 MIT Technology Review TR35 Young Innovator Award for his outstanding contributions to video search. He won the Best Paper and Best Demonstration Awards at ACM Multimedia 2007, the Best Poster Award at IEEE International Workshop on Multimedia Signal Processing 2008, the Best Student Paper Award at ACM Conference on Information and Knowledge Management 2009, and the Best Paper Award at International Conference on Multimedia Modeling 2010. He was named one of Global Entrepreneur's "Business Elites of People under 40 to Watch" in 2009.

Changshui Zhang received the B.S. degree in mathematics from Peking University, Beijing, China, and the Ph.D. degree from Tsinghua University, Beijing, in 1986 and 1992, respectively. He joined the Department of Automation, Tsinghua University, in 1992, where he is currently a Professor. He has authored over 200 papers published in journals and conferences. His current research interests include pattern recognition and machine learning. Dr. Zhang is currently on the editorial board of Pattern Recognition.

Long Quan is a Professor of the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology (HKUST). He received his Ph.D. in 1989 in Computer Science from INPL, France. He entered as a permanent researcher into the Centre National de la Recherche Scientifique (CNRS) in 1990 and was appointed at the Institut National de Recherche en Informatique et Automatique (INRIA) in Grenoble, France. He joined the HKUST in 2001, and was the founding Director of the HKUST Center for Visual Computing and Image Science. He is a Fellow of the IEEE Computer Society.

He works on vision geometry, 3D reconstruction and image-based modeling. He supervised the first French Best Ph.D. Dissertation in Computer Science of the Year 1998 (le prix de thèse SPECIF 1998, now le prix de thèse Gilles Kahn), the Piero Zamperoni Best Student Paper Award of the ICPR 2000, and the Best Student Poster Paper of IEEE CVPR 2008. He co-authored one of the six highlight papers of the SIGGRAPH 2007. He was also elected as the HKUST Best Ten Lecturers in 2004 and 2009. He has served as an Associate Editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) and a Regional Editor of Image and Vision Computing Journal (IVC). He is on the editorial board of the International Journal of Computer Vision (IJCV), the Electronic Letters on Computer Vision and Image Analysis (ELCVIA), the Machine Vision and Applications (MVA), and the Foundations and Trends in Computer Graphics and Vision. He was a Program Chair of IAPR International Conference on Pattern Recognition (ICPR) 2006 Computer Vision and Image Analysis, is a Program Chair of ICPR 2012 Computer and Robot Vision, and is a General Chair of the IEEE International Conference on Computer Vision (ICCV) 2011.