RBIR-oriented image segmentation combining Graph Cut and Minimum Spanning Tree

Anonymous Submission

ABSTRACT

Many researchers tried to exploit Region-based Image Retrieval (RBIR) in recent years, but most of them focused on matching strategies while few paid attentions to segmentation algorithms. However, as an important part of RBIR systems, segmentation should have great influence on the performance of the entire systems and thus is worth studying. Therefore, this paper analyzes how segmentation methods affect the performance of RBIR systems, and further proposes an RBIR-oriented image segmentation method named GC-MST, which is a combination of graph-cut-based and MST-based methods. In order to evaluate GC-MST, we compared it with several unsupervised segmentation algorithms in experiments, which were conducted by constructing RBIR systems with different segmentation algorithms and comparing their efficiency and retrieval performance. The results proved that GC-MST is capable of achieving the highest retrieval performance among all competing methods on four widely used datasets, while only consuming moderate amount of time both online and off-line.

CCS Concepts

•Computing methodologies → Visual content-based indexing and retrieval; Image segmentation;

Keywords

content-based image retrieval, region-based image retrieval, image segmentation, graph cut, minimum spanning tree

1. INTRODUCTION

Region-based Image Retrieval (RBIR) is a branch of Content-based Image Retrieval (CBIR), but it is different from traditional CBIR which is based on global features or local features. Instead of comparing images as wholes, RBIR chooses to integrate the similarities between homogenous (or semantically meaningful) regions into the similarities between images with certain matching strategies. In practice, RBIR usually works in a scale between global features like *Color and Edge Directivity Descriptor* (CEDD) [6] and key-point-based local features such as SIFT [21] and SURF [4].

The processes of RBIR systems [8, 11, 17, 20, 27] is usually as follows: **1)** segment images into regions with segmentation algorithms; **2)** describe each of the regions with visual features; **3)** obtain the similarities between images by applying a certain matching strategy on region sets. Hence, an RBIR system has three crucial parts, which are *segmentation algorithm*, *visual feature* and *matching strategy*.

There were many papers exploiting RBIR in the last a few years, most of which tried to propose different matching strategies such as *Unified Feature Matching* (UFM) [8], *Integrated Region Matching* (IRM) [20] and *Bag of Regions* (BoR) [27]. Few of the papers paid attentions to segmentation algorithms, i.e. most of the researchers chose existing segmentation algorithms without giving a clear reason. However, segmentation is an important part of RBIR systems, it should have great influence over the efficiency and retrieval performance of the entire systems.

[11] proposed a segmentation algorithm *Multiresolution Recursive Shortest Spanning Tree* (M-RSST) for RBIR application. By constructing graphs on multiresolution grids instead of pixel sets, M-RSST can reduce the number of vertices and in turn improve the efficiency of RSST algorithm. However, M-RSST was focused only on time cost and did not discuss the issue of retrieval performance.

Semantic Segmentation [3, 7] is a recently blooming territory of image segmentation study. The targets of semantic segmentation and RBIR are similar, because they both need to choose a group of regions from images according to given conditions, which in semantic segmentation is for recognition and in RBIR is for matching. However, the recognition (or classification) process of semantic segmentation is too costly and unnecessary for image retrieval tasks. Meanwhile, it is nearly impossible to train enough models for a vast amount of arbitrary images, on which RBIR is supposed to work. Therefore, these methods are not suitable to be used directly in RBIR systems.

Due to the lack of researches analyzing the influence of segmentation methods on the retrieval performance of RBIR systems, we focused on studying this problem. The work of this paper is based on one question: what could be the properties of a good RBIR-oriented segmentation algorithm? Intuitively, since segmentation is only a part of an RBIR system, it must be efficient enough not to slow down the entire system while achieving good retrieval performance with proper matching strategies, but this answer

DOI: 10.1145/1235

is too general.

To formally answer the question, this paper first analyzes the performance of RBIR systems both analytically and empirically, and gives several characteristics as requirements to RBIR-oriented segmentation algorithms, including *efficiency*, *generalization*, *stability* and *controlled region count*. Based on these analyses, we will then propose an image segmentation method. Meanwhile, validity of the proposed method also needs to be evaluated experimentally, so we conducted a series of experiments to compare its efficiency and retrieval performance with those of a few popular segmentation algorithms, as described in Sec.4.

The image segmentation method proposed here is graph-based, or more specifically a combination of graph-cut-based and MST-based methods, so we named it GC-MST meaning literally *Graph Cut Minimum Spanning Tree*. GC-MST was inspired by *Mean Cut segmentation* [5], *Recursive Shortest Spanning Tree* (RSST) [19] and *Gomory-Hu tree* [14].

Graph-cut-based and MST-based segmentation algorithms are widely used in existing systems. Cut-sets are inherently suitable for the description of region boundaries, but graph cut problems under arbitrary energy functions are N-P hard, though specific problems such as Mean Cut and NCut [25] have their own fast algorithms. In contrast, MST-based methods are guaranteed to be of polynomial time, but their ability of integrating image information is relatively low. Therefore, a combination of these two could provide a tradeoff between accuracy and efficiency, and this is where Gomory-Hu tree came into the picture.

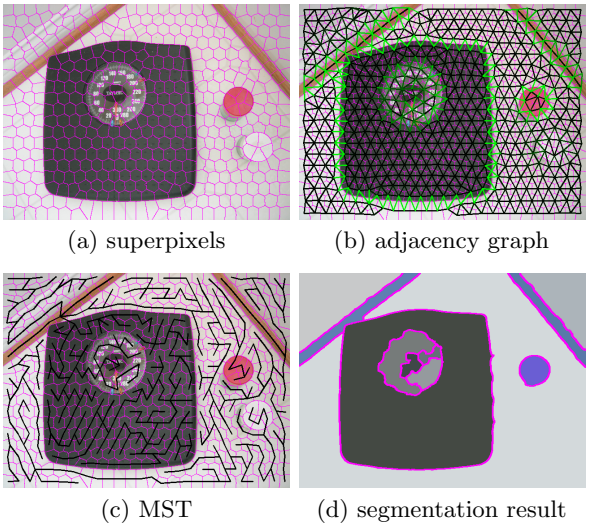
Gomory-Hu tree perfectly combined maximal flow (minimal cut) and a tree-based structure to solve multi-terminal network flow problem. The idea of associating edges in a tree with cut-sets in a graph is inspired, but the construction of Gomory-Hu trees is a relatively slow process, which needs to solve $|V|-1$ flow problems, assuming the graph is $G=(V,E)$. For most existing superpixel extractors, especially SLIC [1], each output superpixel has a limited number of neighbors, i.e. there are 2 constants $n_1, n_2 \in \mathbb{R}^+$ such that $n_1 |V| \leq |E| \leq n_2 |V|$, so $O(|E|) = O(|V|)$. Therefore, the time complexity is approximately $O\left[(|V|-1)\cdot|V|^2\right]$ to $O\left[(|V|-1)\cdot|V|^2\cdot\log|V|\right]$. It is a little slow, and some properties of Gomory-Hu trees are not necessary for arbitrary energy functions, so we propose to replace Gomory-Hu trees with simple MSTs in the segmentation process.

The process of GC-MST is similar to that of a top-down MST-based segmentation algorithm: **1)** generate graph and MST from the target image; **2)** score MST edges with a cut-set-based energy function; **3)** cut the edge with the highest score, and then update edge scores; **4)** repeat step 3 until stopping criteria are met. Fig.1 gives an illustration.

For evaluation, we compared GC-MST with several unsupervised segmentation algorithms experimentally. The retrieval performance of RBIR systems with different segmentation algorithms, visual features and matching strategies were compared on four public datasets, and the results will be presented and discussed in Sec.4. Meanwhile, Sec.4.3 will briefly discuss how the efficiency of an RBIR system is affected by the segmentation algorithm it use, and argue that GC-MST is a better tradeoff between retrieval performance and efficiency than the other competing methods.

The main *contribution of this paper* lies in two parts:

- analyzing the influence of segmentation methods over the retrieval performance of RBIR systems;
- proposing a novel image segmentation algorithm specially designed for region-based image retrieval, named *Graph Cut Minimum Spanning Tree* (GC-MST).



(a) superpixels (b) adjacency graph

(c) MST (d) segmentation result

Figure 1: Image segmentation process of GC-MST. In the first three figures, thin pink lines mark the boundaries of superpixels, and thick green (or black) lines show edges in the graph or MST, among which brighter color means higher difference.

The rest of this paper is structured as follows. Before proposing GC-MST, a few analyses of the influence of segmentation over RBIR performance will be given in Sec.2. Then, Sec.3 will present a fully detailed description of GC-MST algorithm. In order to evaluate the performance of GC-MST, several experiments have been conducted, and their results will be presented and discussed in Sec.4. Finally, Sec.5 concludes our work.

2. THE INFLUENCE OF SEGMENTATION OVER RBIR PERFORMANCE

RBIR methods are designed to be improvements over CBIR methods based on global features or *Bag of Visual Features* (BoVF) descriptors. As presented in Sec.1, an RBIR system normally works in a process consisting of tree steps, which are *segmentation*, *feature extraction* and *matching*. The main difference between RBIR and CBIR is that by introducing segmentation into image representation and matching of image retrieval, RBIR methods are capable of gathering more information from images to provide finer similarity calculations. However, finer matching approaches do not necessarily mean better performances in retrieval tasks, because while being able to identify exact matches of a query image more easily, they usually lose the ability of generalization. Therefore, the best choice usually is a tradeoff between matching precision and generalization.

Before presenting further discussion of this problem, we would like to define *linear features* as below:

DEFINITION 1. *If a visual feature F is linear, when a set of regions $\{R_i | i = 1, 2, \cdots, n\}$ forms a segmentation of an image I, the descriptor of I is a linear combination of the descriptors of the regions, i.e.*

$$\vec{F}(I) = \sum_{i=1}^{n} w_i \vec{F}(R_i), I = \bigcup_{i=1}^{n} R_i, \emptyset = \bigcup_{j \neq k} R_j \cap R_k \qquad (1)$$

Normally, for features based on histograms (e.g. CEDD) or statistics (e.g. Tamura textural features [26]), $\{w_i\}$ *is a partition of* 1*, meaning* $w_i \geq 0$ *and* $\sum_{i=1}^{n} w_i = 1$.

Linear feature is an important concept. Since the weighted sum of all region descriptors is always the same (equal to the image descriptor), linear features are able to maintain some basic information of the image regardless of the segmentation, while nonlinear features can not. This fact leads to a few differences between RBIR systems using linear features and those using nonlinear features, as makes it necessary to analyse them separately.

In this section, we will discuss the question of how segmentation methods affect the performance of RBIR systems. First, RBIR systems using nonlinear features or *Bag of Regions* (BoR) [27] matching will be briefly discussed, and the importance of segmentation stability to them will be explained. Then, systems using linear features with pair-wise Euclidean distance will be singled out and analyzed in detail both analytically and experimentally. Lastly, by summarizing the conclusions of the above discussions, we give a number of characteristics that are likely owned by RBIR-oriented image segmentation methods.

Besides being an interesting problem, the analysis of segmentation influence over RBIR performance can also give heuristic terms to algorithm designing. The segmentation algorithm proposed in Sec.3 is partly designed according to these analyses, i.e. a few parts of GC-MST are meant to satisfy the requirements of RBIR-oriented image segmentation which are proposed by the following discussions.

2.1 Nonlinear features and BoR matching

With nonlinear features or BoR matching, region descriptors are no longer parts of the image descriptor, because they contain information that is particular to them. In this case, an RBIR method is not just an enhanced version of an existing CBIR method but an individual CBIR method that works in its own logic.

Meanwhile, these RBIR methods are more sensitive to segmentation errors than those using linear features. As stated before, when working with linear features, some basic information of the image remains invariant regardless of the segmentation. In contrast, different segmentation results may lead to completely different representations of the image under nonlinear features or BoR matching, so slight changes in segmentation could have major impact on the matching process, e.g. if a trained object classifier is used to categorize regions during a BoR matching process, without precise segmentation of object areas, the classification results are unlikely to be reliable and so are the calculated similarities.

Fortunately, most region descriptors are not so delicate. What *they need from segmentation to achieve high matching accuracy are generalization (region-wise) and stability (image-wise)*, i.e. differences between similar images such as change in illumination, moving of the view ports and slight deformation of objects, should not lead to major changes to the segmentation results.

2.2 Linear features with pair-wise Euclidean distance

After describing images with vectorized features, *Euclidean distance* can always be used to measure the similarity between each pair of them. This seems overly simple, and there are indeed many researches proposing to use other distance measures such as *Manhattan distance*, *cosine distance*, *Jaccard distance* and so on. However, the Euclidean distance is the easiest to understand and analyze, and it works well in many scenarios and systems. Therefore, discussions in this section will be based on Euclidean distance.

With linear features and Euclidean distance, the difference between two images can be defined as the following equation, in which feature vectors of the two images are $\vec{F}^{(1)} = \sum_{i=0}^{n} w_i \vec{F}_i^{(1)}$ and $\vec{F}^{(2)} = \sum_{i=0}^{n} w_i \vec{F}_i^{(2)}$ respectively.

$$d_t \triangleq \left\| \vec{F}^{(1)} - \vec{F}^{(2)} \right\| = \left\| \sum_{i=0}^{n} w_i \cdot (\vec{F}_i^{(1)} - \vec{F}_i^{(2)}) \right\| \qquad (2)$$

Pair-wise Euclidean distance matching makes use of the differences between regions to obtain a single difference between two images. Matching strategies of this category usually work in the following process: **1)** construct region pairs $(R_j^{(1)}, R_k^{(2)})$ by selecting two regions from the two images $I^{(1)}$ and $I^{(2)}$ respectively, i.e. $R_j^{(1)} \in I^{(1)}$ and $R_k^{(2)} \in I^{(2)}$; **2)** obtain the difference between each pair of regions by calculating the Euclidean distance of their descriptors; **3)** integrate differences of region pairs into the difference between images.

For simplicity of analyses, we could assume that each pair find in the images is the perfect match, which means each image has the same number of regions and each region in a pair has the same weight. However in practice, this is not likely to be true even for two copies of one same image, because of the instability of segmentation methods. To address this problem, different integration strategies have been proposed for step 3, but the one that solved this problem completely is *Integrated Region Matching* (IRM) [20].

Simply speaking, IRM generates two equally sized region sets, each associated to one of the images, by dividing the regions into even smaller subregions, and IRM guarantees that each subregion in one image has a corresponding equally weighted subregion in the other image. More specifically, each region $R_j^{(1)}$ in one of the original region sets is converted into a region set $A_j^{(1)}$ such that $\bigcup_{e \in A_j^{(1)}} e = R_j^{(1)}$ and $\sum_{e \in A_j^{(1)}} w(e) = w(R_j^{(1)})$, and then a new subregion set is generated such that $S^{(1)} = \bigcup_j A_j^{(1)}$. IRM process guarantees that $|S^{(1)}| = |S^{(2)}|$, and there is a bijection $M : S^{(1)} \to S^{(2)}$ such that $\forall e \in S^{(1)}$, $w(e) = w[M(e)]$.

When using IRM-based strategies, pair-wise Euclidean distance can be defined as follows:

$$d_r \triangleq \sum_{i=0}^{n} w_i \cdot \left\| \vec{F}_i^{(1)} - \vec{F}_i^{(2)} \right\| \qquad (3)$$

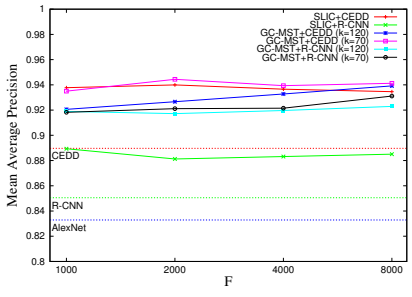
According to the *generalized triangle inequality*, $d_t \leq d_r$, which proves that by introducing segmentation into the matching process, slight differences between images are enhanced. Meanwhile, the enhancement can be easily calculated:

$$\vec{\delta}_i \triangleq \vec{F}_i^{(1)} - \vec{F}_i^{(2)}, i = 0, 1, \cdots, n$$

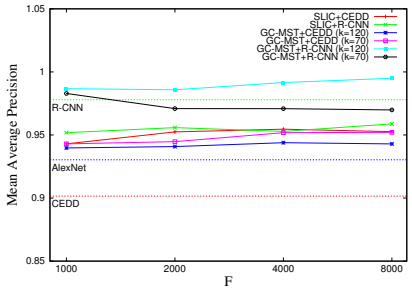
$$\begin{aligned}\Delta^2 &\triangleq d_r^2 - d_t^2 \\ &= \sum_{i\neq k}\left[\left\|\vec{\delta}_i\right\|\cdot\left\|\vec{\delta}_k\right\| - \vec{\delta}_i\cdot\vec{\delta}_k\right] \\ &= \sum_{i\neq k}\left\|\vec{\delta}_i\right\|\cdot\left\|\vec{\delta}_k\right\|\cdot\left(1-\cos<\vec{\delta}_i,\vec{\delta}_k>\right) \qquad (4)\end{aligned}$$

In order to improve the retrieval performance of CBIR, we want Δ^2 to be as big as possible when comparing two images from different categories, and Δ^2 is supposed to approach zero when comparing similar images. As shown in (4), when the difference between two images comes from environmental factors such as light and view change, all regions are likely to be under similar affections, which in turn causes $<\vec{\delta}_i,\vec{\delta}_k>\doteq 0$ and $\Delta^2 \doteq 0$. Meanwhile, when two images from different categories are compared to each other, we have $d_t \leq d_r$, which means the difference is enhanced. These facts proved the effectiveness of IRM-based strategies.

However, in most cases, analytical analysis of the performance improvement is difficult, so we conducted an experiment to evaluate it in practical situations. First, we tested the performances of CBIR methods with global features (CEDD, AlexNet [18] and R-CNN [13]), and then the performances of SLIC [1] (superpixel extractor) and GC-MST were evaluated. Datasets used in this experiment are generated from public datasets ZuBuD [24] and ukbench [22] by selecting around 100 pictures in approximate 25 classes. The reason of using two derived datasets instead of the original ones lies in the fact that the superpixels in each image are too many, which slows down the retrieval process greatly.



(a) ZuBuD100



(b) ukbench100

Figure 2: Experimental results for the analyses of segmentation influence on RBIR performance. Solid lines and points show retrieval precisions of different RBIR combinations, and dotted lines give three baselines which are retrieval precisions of CBIR methods using different global features. The x axes are logarithmic.

Fig.2 gives the results of this experiment. RBIR outperformed CBIR significantly with both CEDD (highly linear) and R-CNN (with relatively low linearity), which proved the validity of pair-wise Euclidean distance matching. Meanwhile, the performances of SLIC and GC-MST, which is based on SLIC, are quite similar, and when using R-CNN as visual feature, GC-MST worked even better than SLIC, which output tens of times more regions than GC-MST did. This suggests that additional merging over superpixels are necessary for improving both efficiency and retrieval performance of an RBIR system, and *it is possible to achieve high retrieval precision with a small number of output regions.*

Parameter F controls the number of superpixels generated by SLIC from one image, and k is the stopping criterion parameter which controls the number of regions generated by GC-MST per image. According to Fig.2, the influence of F is relatively small for both SLIC and GC-MST, which also suggests that finer superpixel extraction cannot improve retrieval performance of RBIR above a certain degree, while the increasing number of superpixels will surely lead to much higher time cost. The influence of k is more complicated and will be discussed in detail in Sec.4.4.

2.3 RBIR-oriented segmentation

Based on the discussions in the previous part of this section, we present a few characteristics possibly owned by an RBIR-oriented segmentation algorithm:

1. **efficiency**: lower time consumption of segmentation for off-line efficiency of RBIR systems;
2. **generalization and stability**: toleration of intra-category variations of regions and images for coping with nonlinearity of features, as discussed in Sec.2.1;
3. **controlled region count**: outputting as few regions as possible for efficiency reason while avoiding losing retrieval performance, as discussed in Sec.2.2.

In order to achieve *generalization and stability*, segmentation algorithms should most likely focus on clear boundaries between different (semantical) types of regions and ignore most textures within regions, which in turn will probably yield fewer regions. In this case, graph-cut-based segmentation is suitable for solving this problem, and when combining with MST-based methods, it could also make the time cost under control for an arbitrary energy function. A segmentation framework combining graph cut and MST will be presented in Sec.3.1 and Sec.3.2, and its corresponding boundary-focused energy function will be given in Sec.3.3.

The efficiency issue of GC-MST, including *efficiency* and *controlled region count*, will be discussed according to experimental results in Sec.4.3. And, a few empirical methods of choosing model parameters of GC-MST to achieve good tradeoff between efficiency and retrieval performance will be presented in Sec.4.4.

3. GRAPH CUT MINIMUM SPANNING TREE

As shown in Fig.1, the segmentation process of GC-MST is similar to that of a top-down MST-based algorithm:

1. target image is divided into superpixels by using SLIC [1] algorithm, and then an adjacency graph $G = (V, E)$ is constructed by regarding superpixels as vertices and connecting adjacent superpixels with undirected edges;

2. each edge in the graph is weighted by the Euclidean distance between the descriptors of the two superpixels connected by it, then a Minimum Spanning Tree $T = (V_t, E_t)$ is generated from G, i.e. $V_t = V$ and $E_t \subset E$;

3. each edge in E_t is scored by integrating the information of its corresponding cut-set with an energy function;

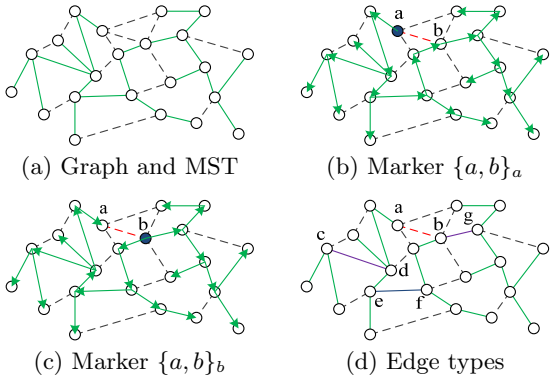
4. the edge with the highest score is cut to split the original tree into two subtrees, and then scores of surviving edges are updated;

5. step 4 is repeated until the stopping criteria are met.

The main difference between GC-MST and traditional MST-based methods lies in step 3, which GC-MST implements by making use of the idea of graph-cut-based methods, and the most important contribution of GC-MST is proposing a novel marker propagation method for identifying corresponding cut-sets of MST edges, which will be introduced in detail in Sec.3.1. Step 4 is highly related to the marker propagation process, so it will be presented after that in Sec.3.2. The energy function used in step 3 and 4 is inspired by Mean Cut but with a few changes to fit the use of superpixels, so we will briefly introduce it in Sec.3.3.

3.1 Marker propagation

Assuming there are a graph $G = (V, E)$ and its corresponding MST $T = (V_t, E_t)$, the marker propagation is done for each edge e such that $e \in E$ and $e \notin E_t$. Each edge has 2 markers each associated to one of its ends, e.g. $\{a, b\}$ has $\{a, b\}_a$ and $\{a, b\}_b$, and each of the two markers propagates individually as shown in Fig.3(b) and (c). To put it simply, if we see node a as root of the MST, the propagation of marker $\{a, b\}_a$ is in the exact same way of passing a token down the tree, from the root to each leaf.



(a) Graph and MST (b) Marker $\{a, b\}_a$

(c) Marker $\{a, b\}_b$ (d) Edge types

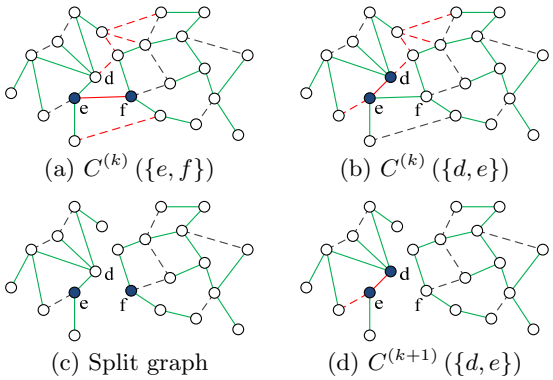
Figure 3: Marker propagation and cut-set identification in GC-MST. Solid lines represent MST edges and dashed lines are edges within the graph but outside MST.

After the propagation of all the markers, we can get two marker sets for each edge in the MST. For an edge $\{e, f\}$, the two marker sets are denoted by $M_{e\to f}$ and $M_{f\to e}$, where $M_{e\to f}$ contains all the markers passed from e to f and the same goes for $M_{f\to e}$. In this way, by recording the propagation direction of each marker, the cut-set of an MST edge can be easily identified, as shown in Fig.3(d). $\{a, b\}_a$ and $\{a, b\}_b$ passed $\{e, f\}$ in different directions, so $\{a, b\} \in C(\{e, f\})$, where $C(u)$ means the cut-set corresponding to edge u; $\{a, b\}_a$ and $\{a, b\}_b$ passed $\{c, d\}$ or $\{b, g\}$ in the same direction, so $\{a, b\} \notin C(\{c, d\})$ and $\{a, b\} \notin C(\{b, g\})$.

According to Fig.3, the time complexity of propagation of one marker is $O(|E_t|)$, so the time complexity of propagation of all markers is $O(|E| \cdot |E_t|) = O(|V|^2)$.

3.2 Cut-set update

After the marker propagation, the cut-sets are identified, and then it should be easy to calculate the scores of edges by using a cut-set-based energy function. However, after the first cut a problem arises: edge scores calculated on the original tree cannot be used directly in the rest of the process, because in the following iteration cycles cutting an edge will be splitting one of the subtrees instead of the original tree. Therefore, the scores or cut-sets should be updated.



(a) $C^{(k)}(\{e, f\})$ (b) $C^{(k)}(\{d, e\})$

(c) Split graph (d) $C^{(k+1)}(\{d, e\})$

Figure 4: The effect of cutting edge $\{e, f\}$. Red (solid and dashed) lines represent edges in the cut-sets.

As shown in Fig.4(a) and (c), cutting an edge $\{e, f\}$ in the MST equals removing its entire cut-set $C(\{e, f\})$ from the graph. Meanwhile, after cutting $\{e, f\}$, the cut-set of $\{d, e\}$ shrank, as shown in Fig.4(b) and (d).

There are two ways to update the cut-sets:

1. remove $C^{(k)}(\{e, f\})$ from the cut-set of each edge, i.e. $C^{(k+1)}(u) \leftarrow C^{(k)}(u) \setminus C^{(k)}(\{e, f\})$ for $\forall u \in E_t$;

2. since the main consequence of cutting $\{e, f\}$ is preventing markers from passing though it, we can remove elements of $M^{(k)}_{e\to f}$ from all marker sets within the subtree attached to node f and elements of $M^{(k)}_{f\to e}$ from those within the subtree attached to node e, i.e. assuming the subtrees attached to e and f are $T(e) = (V_t(e), E_t(e))$ and $T(f) = (V_t(f), E_t(f))$ respectively, $M^{(k+1)}_{v\to u} \leftarrow M^{(k)}_{v\to u} \setminus M^{(k)}_{e\to f}$ for all $v, u \in V_t(e)$ such that $\{v, u\} \in E_t(e)$, and the same goes for $T(f)$.

These two approaches are equivalent in effect, and their time complexities are also both $O(|E^{(k)}_t| \cdot |E^{(k)}|) = O(|V^{(k)}|^2)$, where $G^{(k)} = (V^{(k)}, E^{(k)})$ and $T^{(k)} = (V^{(k)}_t, E^{(k)}_t)$ are the subgraph and subtree split in the kth iteration cycle. The choice between the two depends mainly on the implementation. More specifically, if after the marker propagation cut-sets are stored, the first way is more suitable, and if marker sets are stored, the second way is to choose. However, there is a small difference: to use the first approach, additional

processes or flags are needed in order to determine whether an edge belongs to a specific subtree, while in the second approach, marker propagation can be used for this purpose.

3.3 Energy function

The energy function used in GC-MST is named Superpixel Mean Cut or S-MeanCut. It is inspired by Mean Cut [5] segmentation, which considers only edges on the boundaries while ignoring textures within regions completely. Since detailed textures can be handled by visual features, by adopting the idea of focusing on boundaries, GC-MST is potentially stable and capable of providing high generalization to RBIR systems. However, the energy function of Mean Cut is designed for pixel-based applications, so it needs to be modified for superpixel-based segmentation.

$$SMeanCut = \frac{\sum_{i \in C(e)} l(i) \cdot w(i)}{\left(\sum_{i \in C(e)} l(i)\right)^q}, e \in E_t, q \in \mathbb{R}^+ \qquad (5)$$

In (5), $l(i)$ is the length of the common boundary corresponding to edge i, e.g. $l(\{e, f\})$ is the number of pixels which have neighbors belonging to both superpixel e and superpixel f. q is a parameter, which indirectly controls the preference of region size. Generally speaking, a model with small q will try to cut longer boundaries first and in turn yield more big regions, and vice versa.

4. EVALUATION

In order to evaluate GC-MST, it was compared to a few widely used color image segmentation algorithms experimentally. In these experiments, retrieval performance of RBIR systems with different segmentation algorithms, visual features and matching strategies were compared on four widely used public datasets.

As stated in Sec.1, due to the complexity and high time cost, semantic segmentation methods are not suitable to be used directly in RBIR systems. In contrast, traditional unsupervised segmentation methods [10, 25, 12, 2] are much lighter and have been proven effective by many existing systems. Therefore, instead of semantic segmentation algorithms, we chose a few popular unsupervised segmentation algorithms to be the competing methods in our experiments, including *Local Variation segmentation* (LV) [12], *Normalized Cut* (NCut) [25], JSEG [10], and *Color Watershed Adjacency Graph Merge* (CWAGM)[1] [2]. These methods, though did not see much progress recently, are much lighter and general than class-specific supervised semantic segmentation methods, and their effectiveness has been proved in many existing systems.

Visual features used here are in two types: *Color and Edge Directivity Descriptor* (CEDD) [6] represents traditional global features, while AlexNet [18] and R-CNN [13] represents newly proposed CNN-based features. CEDD is one of the most popular global features for CBIR. By integrating color and textural information of images into fixed-sized vectors, CEDD is not only highly efficient but also capable of reflecting different image characteristics. *Convolutional Neural Network* (CNN) has drawn much attention in the areas of image retrieval and recognition recently, and CNN-based CBIR methods achieved amazingly high performances in a few contests and benchmarks. However, compared to traditional manually designed features, the training process of CNN is rather slow and complicated, which makes CNN a little difficult to use in many practical systems. Therefore, a few researchers proposed to use pre-trained models as substitutes of traditional visual features, and their results are encouraging. We adopted this idea and used pre-trained AlexNet and R-CNN models[2] as visual features.

We tested two famous matching strategies *Integrated Region Matching* (IRM) [20] and *Bag of Regions* (BoR) [27]. The retrieval performance of CEDD-based combinations is evaluated with IRM or BoR, while CNN-based combinations are only tested with IRM since the meaning of building codebooks for them is not convincing enough.

The rest of the experiment configuration is as follows:

- **Datasets**: datasets used in the following experiments are INRIA Holiday [15] (1491 pictures), ZuBuD [24] (1005 pictures), UCID [23] (1338 pictures with 200 queries) and ukbench [22] (10200 pictures);
- **Evaluation Measures**: the retrieval performance of different segmentation algorithms is compared in *Mean Average Precision* (MAP), and meanwhile in order to analyze the efficiency of different combinations, we also recorded the *Average Region Count* (ARC), which is the average number of regions generated for one image.

4.1 Working with CEDD

As shown in Table 1, when working with IRM+CEDD, the retrieval performance of GC-MST is the best among all the competing methods, which proved the effectiveness of GC-MST as a part of RBIR systems. Meanwhile, GC-MST and CWAGM outperformed the other 3 methods significantly. This is probably due to the fact that these two methods are both highly dependent on the clarity of region boundaries while mostly ignoring the textures within regions.

BoR-based methods showed little improvement over pure CEDD. We tried two different weighting strategies, with area proportions and with a saliency measure proposed in [9]. When regions are weighted with area proportion, no competing method achieved better results than CEDD on any of the four datasets, and weighting regions with saliency only make CWAGM and GC-MST beat pure CEDD on two datasets (ZuBud and UCID).

4.2 Working with CNN-based features

Table 2 shows the results of experiments with IRM+R-CNN combinations. The combinations of IRM+AlexNet are also tested in our experiments, but the results are very close to and slight worse than those of IRM+R-CNN, so only IRM+R-CNN combinations is discussed here.

GC-MST outperformed all the competing methods, and GC-MST+IRM+R-CNN also achieved higher precision than R-CNN and AlexNet on all the datasets. This proved that even when applied to effective CNN-based features, RBIR techniques can still improve their already high performance.

4.3 Efficiency Issue

The online time cost of IRM is almost solely determined by the number of regions, more specifically its time complexity

[1]CWAGM are implemented by LTI-LIB project.

[2]Networks of AlexNet and R-CNN are implemented and trained by Caffe project [16].

Table 1: Retrieval performance of IRM+CEDD in MAP (%). ($F = 2000$, $q = 0.6$, $k = 120$)

		ZuBuD		UCID		ukbench		Holiday	
		MAP	ARC	MAP	ARC	MAP	ARC	MAP	ARC
RBIR	LV [12]	85.83	241	71.93	136	72.91	149	73.96	255
	JSEG [10]	62.99	68	36.46	72	52.06	54	55.37	186
	Ncut [25]	80.62	128	72.61	128	73.07	128	71.91	128
	CWAGM [2]	**88.57**	536	74.54	390	76.94	372	73.45	4854
	GC-MST	87.60	120	**75.27**	120	**78.80**	120	**74.97**	120
CBIR	CEDD [6]	79.12	1	67.41	1	70.26	1	69.82	1

Table 2: Retrieval performance of IRM+R-CNN in MAP (%). ($F = 2000$, $q = 0.6$, $k = 70$)

		ZuBuD		UCID		ukbench		Holiday	
		MAP	ARC	MAP	ARC	MAP	ARC	MAP	ARC
RBIR	LV [12]	74.52	210	80.69	120	82.67	152	70.45	241
	JSEG [10]	46.72	29	31.92	51	6.01	19	48.98	125
	Ncut [25]	78.52	64	81.57	64	84.29	64	76.91	64
	CWAGM [2]	79.43	249	80.51	204	84.42	216	58.75	1697
	GC-MST	**88.39**	70	**84.07**	70	**86.35**	70	**79.64**	70
CBIR	R-CNN [13]	83.01	1	80.15	1	83.53	1	76.60	1
	AlexNet [18]	83.38	1	82.88	1	84.95	1	77.43	1

is $O(|R^{(1)}| \cdot |R^{(2)}|)$ for each pair of images. The off-line time cost mainly consists of two parts: segmentation and feature extraction. Most of the competing methods have similar time consumptions except JSEG, which is slower than others by around 3 orders of magnitude, and time cost of feature extraction is also governed by region count. Overall, the efficiency of an RBIR system is determined by ARC.

According to Table 1 and Table 2, GC-MST have moderate ARCs in both experiments while achieving the highest retrieval precision among all competing methods. This proved that GC-MST is capable of finding an optimal tradeoff between efficiency and retrieval performance.

4.4 Tuning the parameters

There are mainly three model parameters which need tuning for GC-MST, which are F, k and q. As discussed in Sec.2 and Sec.3, F controls the number of superpixels on which the segmentation process works, k controls the number of output regions of GC-MST, and q indirectly controls the preference of region size. Since different parameter choices can affect the output of GC-MST, they could also affect the retrieval performance of RBIR systems. Therefore, we conducted a series of experiments on the dataset UCID in order to have an intuitive understanding of these influences. Fig.5 shows the experimental results of combinations GC-MST+IRM+CEDD and GC-MST+IRM+R-CNN under different parameter configurations.

For F, both lines are almost horizontal, which suggests that the influence of parameter F over MAP is relatively small. The experimental results proved that as long as grouping criteria are stable, coarse elements (bigger superpixels) and fine elements (small superpixels) are not much different for RBIR application. However, bigger F means larger $|V|$, and thus higher time cost for GC-MST, so F should be set as small as possible in practice.

Combination with CEDD and R-CNN behaved differently while k changes. For R-CNN, MAP reached its maximum at $k \doteq 50$ and then dropped quickly as k grew. For CEDD, it appears that MAP was increasing along with k, but the increments quickly reduced to almost nothing when k continued to grew after $k = 100$. Although bigger k may lead to higher MAP, we have argued in Sec.4.3 that ARC affects both the online and off-line efficiency of RBIR systems, so it is not suitable to choose too big a k.

It seems that q does not have much influence over combinations with CEDD when compared to those with R-CNN. For R-CNN, there is a big difference between $q \leq 1$ and $q > 1$, and smaller q or bigger regions are preferred. This is probably because bigger regions are more stable since they are mostly major parts of objects or scenes rather than textural details.

5. CONCLUSION

In order to find the characteristics of RBIR-oriented segmentation algorithms, this paper had analyzed and discussed in detail the influence of segmentation methods over the retrieval performance of RBIR systems. Then based on these analyses, we proposed an RBIR-oriented image segmentation method named GC-MST, which combined the advantages of graph-cut-based and MST-based methods. In the experiments conducted to evaluate GC-MST, it is compared with a few traditional unsupervised segmentation algorithms on four popular public datasets, and the results proved that GC-MST is capable of achieving the highest retrieval precision and finding an optimal tradeoff between efficiency and retrieval performance.

6. 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 transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] J. P. Alvarado Moya. *Segmentation of color images for interactive 3d object retrieval*. PhD thesis, Bibliothek der RWTH Aachen, 2004.

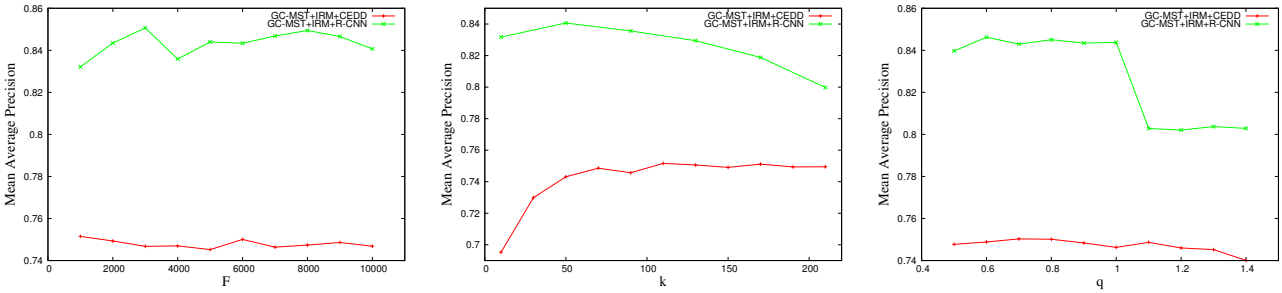


Figure 5: The influence of different parameters over retrieval performance of GC-MST. The base configuration is $(F = 2000, q = 0.6)$, and for GC-MST+IRM+CEDD $k = 120$, for GC-MST+IRM+R-CNN $k = 70$.

[3] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, pages 3378–3385. IEEE, 2012.

[4] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417, 2006.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[6] S. A. Chatzichristofis and Y. S. Boutalis. Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In *Computer vision systems*, pages 312–322. Springer, 2008.

[7] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2014.

[8] Y. Chen and J. Z. Wang. A region-based fuzzy feature matching approach to content-based image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 24(9):1252–1267, 2002.

[9] M. M. Cheng, J. Warrell, W. Y. Lin, S. Zheng, V. Vineet, and N. Crook. Efficient salient region detection with soft image abstraction. In *IEEE ICCV*, pages 1529–1536, 2013.

[10] Y. Deng, B. S. Manjunath, and H. Shin. Color image segmentation. In *CVPR*, volume 2. IEEE, 1999.

[11] A. Doulamis, N. Doulamis, and T. Varvarigou. Efficient content-based image retrieval using fuzzy organization and optimal relevance feedback. *International Journal of Image and Graphics*, 3(01):171–208, 2003.

[12] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[14] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

[15] H. Jégou, M. Douze, and C. Schmid. Hamming Embedding and Weak Geometry Consistency for Large Scale Image Search - extended version. Research Report 6709, Oct. 2008.

[16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[17] F. Jing, M. Li, H.-J. Zhang, and B. Zhang. An efficient and effective region-based image retrieval framework. *IEEE Transactions on Image Processing*, 13(5):699–709, 2004.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] S. H. Kwok and A. G. Constantinides. A fast recursive shortest spanning tree for image segmentation and edge detection. *IEEE Transactions on Image Processing*, 6(2):328–332, 1997.

[20] J. Li, J. Z. Wang, and G. Wiederhold. Irm: integrated region matching for image retrieval. In *ACM MM*, pages 147–156. ACM, 2000.

[21] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, volume 2, pages 1150–1157 vol.2, 1999.

[22] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, volume 2, pages 2161–2168. IEEE, 2006.

[23] G. Schaefer and M. Stich. Ucid: an uncompressed color image database. In *Electronic Imaging 2004*, pages 472–480. International Society for Optics and Photonics, 2003.

[24] H. Shao, T. Svoboda, and L. Van Gool. Zubud-zurich buildings database for image based recognition. *Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland, Tech. Rep*, 260, 2003.

[25] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[26] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6):460–473, 1978.

[27] R. Vieux, J. Benois-Pineau, and J.-P. Domenger. *Advances in Multimedia Modeling: 18th International Conference, MMM 2012*, chapter Content Based Image Retrieval Using Bag-Of-Regions, pages 507–517. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.