A New Connected Coherence Tree Algorithm For Image Segmentation

Jingbo Zhou[1], Shangbing Gao[1,2] and Zhong Jin[1]
[1] The School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, China
[2] School of Computer Engineering, Huaiyin Institute of Technology, Huai'an, China
[e-mail: zhoujingbo2006@yahoo.com.cn, luxiaofen_2002@126.com]
*Corresponding author: Jingbo Zhou

Received January 18, 2011; revised March 19, 2012; accepted April 7, 2012; published April 25, 2012

Abstract

In this paper, we propose a new multi-scale connected coherence tree algorithm (MCCTA) by improving the connected coherence tree algorithm (CCTA). In contrast to many multi-scale image processing algorithms, MCCTA works on multiple scales space of an image and can adaptively change the parameters to capture the coarse and fine level details. Furthermore, we design a Multi-scale Connected Coherence Tree algorithm plus Spectral graph partitioning (MCCTSGP) by combining MCCTA and Spectral graph partitioning in to a new framework. Specifically, the graph nodes are the regions produced by CCTA and the image pixels, and the weights are the affinities between nodes. Then we run a spectral graph partitioning algorithm to partition on the graph which can consider the information both from pixels and regions to improve the quality of segments for providing image segmentation. The experimental results on Berkeley image database demonstrate the accuracy of our algorithm as compared to existing popular methods.

Keywords: CCTA, graph-based, image segmentation, multi-scale, spectral graph partitioning

This work is partly supported by NSFC of China (Grant No:90820306 and KT06015), Natural Science Research Project of Jiang Su Provincial Colleges and Universities (No. 11KJD520003).

http://dx.doi.org/10.3837/tiis.2012.04.014

1. Introduction

Image segmentation is an important process in computer vision and image processing [1][2][3], which divides an image into a number of disjoint regions such that the pixels have high similarity in each region and high contrast between regions. Although a human can delineate the object boundaries with much ease, segmenting images is not as easy for a computer. Generally speaking, there are several problems to make image segmentation difficult, such as finding the faint object boundaries and separating an object from a highly cluttered background. In fact, such situations often arise in natural images, as animals have often evolved to blend into their environment.

Numerous methods have already been proposed for image segmentation, e.g. clustering based methods [4], histogram based methods [5], region growing methods [4], and more recent ones such as adaptive thresholding methods [6] and graph based methods [7][8] among others. Clustering based methods, which generally define the segmentation problem as finding the labeling of all pixels in an image that minimizing a specific energy function, are unable to handle unbalanced elongated clusters. When one cluster has much more points than a neighboring cluster, it will erroneously split the larger cluster into artificial sub-clusters. Image thresholding methods are also popular due to their simplicity and efficiency. Traditional histogram-based thresholding algorithms, however, cannot separate those areas which have the same gray level but do not belong to the same part. In addition, they cannot process images whose histograms are nearly unimodal, especially when the target region is much smaller than the background area. Region growing algorithms deal with spatial repartition of the image feature information. Generally speaking, they perform better than the thresholding approaches for several sets of images. However, the typical region growing processes are inherently sequential. The produced regions depend both on the order in which pixels are scanned and on the value of pixels which are gathered to define each new segment. Graph based methods treat image segmentation as a graph partition problem which leads to another problem. In a graph-theoretic approach, the overall segmentation quality depends mainly on the graph affinities which affected by the graph connection radius seriously. It is a dilemma that how large the connection radius of a graph to be chosen. A large radius generally makes segmentation better but require a great computational cost. Smaller radius makes the segmentation falls to extract the global impressions of an image.

According to the aforementioned issues, we can see that image segmentation is important and difficult. Fortunately, several recent works indicate that integration of local grouping cues across long ranges of spatial connections between pairwise pixels would produce impressive segmentation results. However, using the full range connections would require a great computational cost. To avoid it, two conventional approaches that propagate local grouping cues across larger image areas become popular recently. The first one is the proposed multiscale image segmentation frameworks [9][10][11][23]. Their works give efficient approximation to incorporate long-range connections with low complexity. However, they typically fail to detect fine-level details along object boundaries due to the coarsening error. The second one is the segmentation that defined as the grouping of non-overlapping regions other than pixels in [12][13]. It is inspired by the hard constraint whereby pixels in a particular region should have the same label. As well as transferring local information to a larger image area with connections across regions, the region-based segmentation has the benefit of using more informative features extracted from the pixels inside the regions. However, there are

radical difficulties in obtaining the exact solutions if the regions are not consistent with the object boundaries. In fact, such situations often arise in natural images. Unlike these conventional approaches, our algorithm integrates multiscale image segmentation framework and region-based method into a new framework.

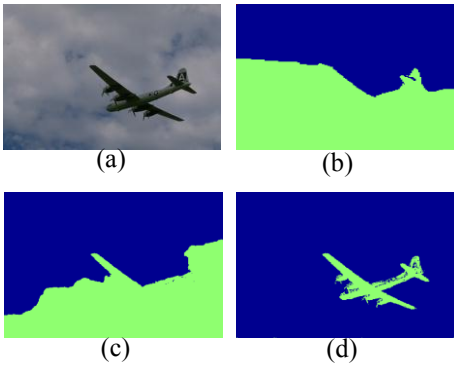


Fig. 1. Examples of segmentation algorithm, (a) Original image, (b) Result of NCut, (c) Result of MNCut, (d) Result of our algorithm, respectively.

In this paper, we propose a new algorithm named Multiscale Connected Coherence Tree algorithm plus Spectral graph partitioning (MCCTSGP), which combines Connected Coherence Tree algorithm [14] (CCTA for short) with spectral graph partitioning into a new image segmentation framework. The main contributions of our work are follows:

- We propose a new approach for multiscale image segmentation. It uses CCTA which can detect coarse and fine details by varying its parameter to generate different regions. Specifically, we firstly construct a graph in which vertexes contain the regions generated by CCTA and the pixels in original image. The weights of the graph are the relationship between each region and all pixels. The graph affinities built in this way consider the information not only from pixels but also from regions. Then partition the graph following the framework of spectral segmentation.
- CCTA is an algorithm which is sensitive to parameter seriously. The quality of segmentation and the number of clusters depend on the spatial scale and the intensity difference scale. Our algorithm decreases the dependence of parameters in CCTA and the class number can be set according to prior information.
- Our algorithm produces high-quality segmentation results. **Fig.1** shows that our proposed method has better segmentations with the object details comparing with Normalized Cut (NCut) [7] and Multiscale NCut (MNCut) [11] in nature images.

This paper is arranged as follows, we first review the CCTA and spectral graph partitioning algorithm in Section 2, and the Multi-scale Connected Coherence Tree algorithm plus Spectral graph partitioning (MCCTSGP) is proposed in Section 3, experiments are given in Section 4, conclusion are summarized in Section 5.

2. Related Work

In this section, we would like to introduce the connected coherence tree algorithm (CCTA) [14] and Spectral graph partitioning [15].

2.1 CCTA

An image I is a pair (χ,φ) consisting of a finite set χ of pixels, and a mapping φ that assigns to each pixel $p=(p_x, p_y)$ a pixel value $\varphi(p)$ in some arbitrary value space. For a fixed central pixel p, we define its neighborhood as

$$N_p=\{q\in\chi:|p_x-q_x|\le k,|p_y-q_y|\le k,k\in Z\} \tag{1}$$

Given an arbitrary threshold $\varepsilon\ge 0$, pixels in N_p can be divided into two different sets

$$\Omega_p=\{q\in N_p: d(\varphi(p),\varphi(q))\le\varepsilon\} \tag{2}$$

and

$$\Omega'_p=N_p-\Omega_p \tag{3}$$

where $d(\varphi(p),\varphi(q))=|\varphi(p)-\varphi(q)|$ is a pixel-value difference measure. Then we can define one neighborhood coherence factor (NCF) as follows:

$$NCF(p)=\frac{|\Omega_p|}{|\Omega'_p|+|\Omega_p|}=\frac{|\Omega_p|}{|N_p|} \tag{4}$$

Where $|\bullet|$ refers to the cardinality of a set, i.e., the number of elements in a set. $NCF(p)$ is defined as the ratio of the number of pixels which have the similar intensity with pixel p to the number of all pixels in the neighborhood. Obviously, this value is quite discrepant for different pixel. When $NCF(p)\ge 0.5, |\Omega_p|\ge|\Omega'_p|$, in this case, p is similar to most of its neighboring pixels and vice versa. **Fig.2** (b) shows the NCF of the image. We choose the pixels, whose NCF larger than 0.5, as the seeds to grow the regions.

$$SEED=\{p: NCF(p)\ge 0.5, p\in\chi\} \tag{5}$$

From the definition stated above, for any seed pixel p in a region, its ε-neighboring pixels in Ω_p are coherent with p which should be in the same region as p. If the pixels within the same region are similar to each other, it is likely that the ε-neighbors of any pixel in this region belong to the same part. More vigorously speaking, it defines a "transitive relationship". Specially, assume $p\in SEED$, $q\in SEED$ and $t\in SEED$, if t is one of the ε-neighbors of q while q is one of the ε-neighbors of p, t together with its all ε-neighbors is grouped into the same region as p. In this way, t together with its all ε-neighbors is obviously in the same region as p. The region starting from one of the seeds in it stops growing when all seeds existing in the region have added their ε-neighboring to this region. Then we can choose a new seed to start a new region. The algorithm terminates when all of the seeds in SEED have been scanned by one time, furthermore, added their ε-neighbors into accordingly region.

Obviously, there are two parameters involved in CCTA: the spatial scale k and the intensity difference scale ε. We can see that, for a fixed k, ε measures the degree of the similarity between p and one of its neighbor in a relative sense. For simplicity, we use Eq (7), the average of $mean(k)_p$, as a candidate for ε. In consequence, how to choose k is crucial for a successful segmentation. Usually, k is related to the size of the objects of interest in the image. It is difficult to choose an optimal k with respect to object dimensions since the size

information of objects is often not a known priori.

$$mean(k)_p = \frac{\sum_{q \in N_p} |I(p) - I(q)|}{|N_p|} \tag{6}$$

$$average(k) = \frac{\sum_{p \in I} (mean(k)_p)}{|I|} \tag{7}$$

2.2 Spectral Graph Partitioning

Spectral graph partitioning is a well studied area with many successful applications. Ng [15] proposed a popular multi-way spectral graph partitioning algorithm that can be simply described as follows.

Given a graph $G = (V, W)$, spectral graph partitioning first computes the degree matrix D, which is a diagonal matrix such that $D(i,i) = \sum_j W(i, j)$. Based on D, it then computes a normalized weight matrix $L = D^{-1}W$ and finds L's K eigenvectors corresponding to K largest eigenvalues $u_1, u_2, \cdots, u_K$ to form matrix $U = [u_1, u_2, \cdots, u_K]$. The rows of U are then normalized to have unit length. Treating the rows of U as K-dimensional embeddings of the vertices of the graph, spectral graph partitioning produces the final clustering solution by clustering the embedded points using K-means.

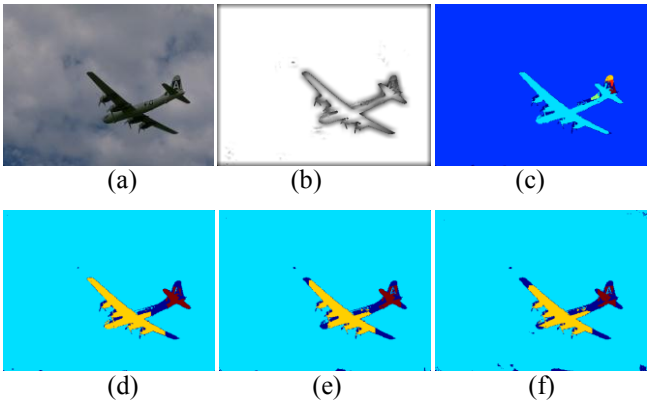


Fig. 2. Examples of CCTA with different scales. (a) Original image. (b) NCF. (c) k=20. (d) k=15. (e) k=10. (f) k=6.

3. An Improved CCTA Algorithm

In this section, we briefly describe the idea of multi-scale CCTA and then discuss how to combine the spectral graph partitioning with multi-scale CCTA into our new framework. We term our algorithm Multi-scale Connected Coherence Tree algorithm plus Spectral Graph Partitioning, MCCTSGP for short.

3.1 Multiscale CCTA

As mentioned above, the segmentation quality of CCTA is sensitive to the parameter k. In **Fig. 2-(c) , Fig. 2-(d) , Fig. 2-(e) , Fig. 2-(f)**, we give an example to demonstrate segmentation results by varying k from 20 to 6, where ε is calculated by Eq (7). When k=20, CCTA groups the pixels into a few clusters, mainly plane and sky, while the plane is separated into some classes. When k=6, pixels are grouped into many clusters, including some trivial segmentations.

Generally speaking, k is related to the size of the objects of interest in the image. If a smaller k is chosen, each region growing by seeds in them would be small. Accordingly, CCTA will be sensitive to noise and produce an over-segmentation. However, in this situation, the algorithm can detect the fine-level details along object boundaries. On the contrary, CCTA will suffer from the increasing computational complexity and yield under-segmentation. Such situation makes the CCTA retain the coarse-level details since the longer-range connections between pixels being built. The analysis for choosing the parameter k implies that we can change the spatial scales of CCTA to detect coarse- and fine-level details and then fuse them into our multiscale framework.

From the analysis stated above, we can see that the long-range connection between pixels generate the segments from the perspective of large scale. On the contrary, short-range means small scale. In our new algorithm, we consider a group parameter $S = \{S_1, S_2, \cdots, S_k\}$ instead of a single parameter k in CCTA. For each parameter S_i, different segment results $\{V_i\}$ can be obtained by running CCTA. The number of regions included in $\{V_i\}$ is decided by the parameter S_i. We do not know which parameter is the best one because there is no prior information about the content of the image. In natural images, different image need different parameter. However, we hope to merge these regions and produce a final segment result. The main contribution of the proposed algorithm is how to design a algorithm that compile the regions generated by CCTA into our new framework, which is the topic of the next subsection.

3.2 MCCTSGP

We design a graph $G^* = (V^*, E^*, W^*)$ inspired by the bipartite graph partitions referred to [16], where the nodes $V^* = (V \cup V_1 \cup V_2 \cup \cdots \cup V_k)$ include a set of pixels V and the sets of regions $\{V_i\}_{i=1,2,\cdots,k}$ which generated by running CCTA with different spatial scales. E^* is the graph edge which connected the region and pixels. The weight matrix W^* is defined as follows.

$$W^*(i,j) = \begin{cases} 1, & \textit{if pixel } i \textit{ belongs to region } j \\ 0, & \textit{otherwise} \end{cases} \tag{8}$$

$$W^*(j,i) = W^*(i,j) \tag{9}$$

It means that if the vertices i and j are both pixels or both regions, $W^*(i,j) = W^*(j,i) = 0$, otherwise if pixel i belongs to region j $W^*(i,j) = W^*(j,i) = 1$ and 0 otherwise.

Note that we can denote W^* as

$$W^* = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix} \tag{10}$$

where A is a connectivity matrix whose rows correspond to the pixels and columns correspond to the regions. $A(i, j)$ is an indicator that takes value 1 if pixel i belongs to the j-*th* region and 0 otherwise.

Fig. 3 shows an example. We use two different scales to segment the image and group the pixels into different regions. An edge between a region and a pixel indicates that the region contains the pixel. All the edges in the graph have weight one and edges with zero weights are omitted from the graph. In this graph, regions are only connected to pixel and vice versa.

We can run the spectral graph partitioning algorithm after the weight matrix being constructed. Note that, in this way, the regions and the pixels can be partitioned simultaneously. The proposed algorithm output the partition of the pixels as the final segmentation.

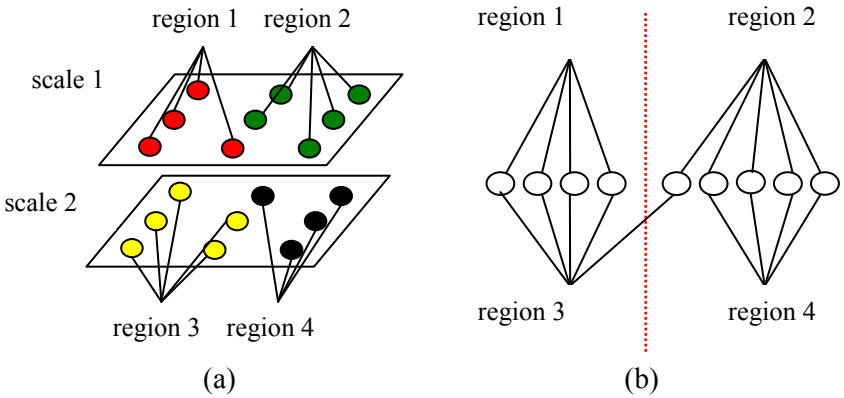


Fig. 3. An example of MCCTSGP formulation. (a) Regions generated by two scales of CCTA. (b) We can use red dashed line to cut one of edges to form two parts, region 1 and region 3, region 2 and region 4, according to spectral graph partitioning objective function, respectively.

According to this definition, we can see that the proposed algorithm has more advantages than others. First, W^* is a sparse matrix. As mentioned in [11], the more sparse the graph affinities W^* is, the easier it is to partition in spectral graph partitioning step. From the definition of W^*, the ratio of zero is more than $\frac{n^2+t^2}{(n+t)^2}, (t << n)$ except the extreme case that every region includes all pixels in an image, where n and t are the number of pixels and regions respectively. This property of graph affinities makes our algorithm calculate efficiently. Second, when much more regions generated, it is easy to add them to our framework. If the segment with a new scale is added, a new set of regions will be added to the graph and each of them will be connected to the instances that it contains. Last but not the least, new method exploits the information of pixels and regions produced by CCTA which make the final partitioning more accurately than those methods using the information only from pixels or regions.

The proposed algorithm can be summarized as follows.

Algorithm MCCTSGP

Given a image $\boldsymbol{I}$, and $S_1, S_2 \cdots, S_k$, K

$S_i = \#scales, i = 1,2,\cdots,k$;

Step 1: For each scale S_i,
Run CCTA to segment the image $\boldsymbol{I}$;
Return each region and the pixels in it.
Step 2: Compute W^* as in Eq (8) and (9);
Step 3: Construct $G^* = (V^*, W^*)$, using spectral graph partitioning to partition it into K clusters and output the groups of pixels.

3.3 Parameter Selection In MCCTSGP

In this subsection, we focus on the discussion to the parameter selection of MCCTSGP. Similar to CCTA, we also calculate the intensity difference scale ε by Eq (7) according to spatial scale k. Since regions generated by different spatial scales of CCTA are needed in the proposed framework, there are two kinds of parameters, the group scales and the number of segments, used in our new algorithm. Then, for group scales, we focus on two problems: (1) How to choose the spatial scale k? (2) How many scales used in MCCTSGP? As mentioned in [14], an optimal k would be in the range [5][15]. To obtain regions with large difference, we get k from the range [5][20] that covers the range of optimum. In addition, the larger the number of scales is, the more regions, which can make sure the accuracy of final segmentation, would be obtained. However, this process requires much more computational cost. Therefore, the trade-off between efficiency and accuracy of our algorithm is captured by the number of scale. We set this number as 4 in our experiments. For another parameter, the number of segments, we set it as the same to NCut and MNCut for a fair comparison.

3.4 Efficiency Analysis

For a clear qualitative analysis of the proposed algorithm, we will discuss its computational complexity. We show that the complexity of our algorithm is linear in the number of pixels. The computational complexity of MCCTSGP is $O(nMN)$, where n, M and N are the number of scales, pixels in neighborhood, and total pixels in an image, respectively. Since we must scan every pixel on each scale to build the graph, the complexity of this step is $O(nN)$. As mentioned in [15], the complexity of spectral graph partition is $O(eN)$, where e is the number of iterations required for k-means to converge. The total complexity of the proposed algorithm is $O(mN)$, $m = (n \times M + n + e)$. We find that $m << N$ since the number of pixels is very large in image segmentation. So the running time of our algorithm is dominated by the number of pixels. According to Haxhimusa [17], we can say that our method is an efficient image segmentation approach.

4. Experimental Results and Analysis

In this section, we demonstrate the results of new algorithm on natural images in the Berkeley segmentation database [21], which contains 300 images and their corresponding ground truth data (at least four human annotations per image).

To evaluate our proposed method, in this section, we, thus, experimentally carry out extensive comparisons with CCTA, NCut, MNCut and CTM [18]. NCut, based on spectral graph theory, provides a mechanism for going from pairwise pixel affinities to partitions that maximize the ratio of affinities within a group to that across groups, i.e., the normalized cut criterion. By embedding the normalized cut problem in the real value domain, this criterion is

formulated as a generalized eigenvalue problem. Then the eigenvectors are used to construct good partitions of the image. MNCut, which works on multiple scales of the image in parallel to capture both coarse and fine level details, is a new multi-scale method in image segmentation. CTM uses simple fixed-size windows to obtain texture features and effectively segments an image by minimizing the overall coding length of the feature vectors. It is a region-based image segmentation algorithm.

Besides some specific default parameters, NCut, MNCut and CTM depend critically upon several other parameters intrinsic to them. For NCut, there are two parameters, the number of regions and the radius of neighborhood. For MNCut, there are also two parameters, the number of regions and scales. For CTM, one parameter exists. Appropriate setting of these parameters is a prerequisite for a successful segmentation. To make a fair comparison, we tune them over a wide range of values and carefully select "optimum" such that each method presents the best results among the numerous different partitions for each image.

4.1 Visual Performance

We first visually verify the segmentation results on the Berkeley segmentation database. For better visual evaluation, we have partitioned the database into five different image categories, each of which consists of images that are more relevant, namely, Animals (**Fig.4**), Landscape (**Fig.5**), Sporting scene (**Fig.6**), Building (**Fig.7**), People (**Fig.8**). In our experiments, we experimentally choose an optimal k of CCTA in a wide range 5 to 20 for each image through trial and error. The results of CTM partly come from the webpage of the author. We set the radius of NCut $r = 10$ and the scales of MNCut $scales = \{1,4,9\}$, which suggest by the authors respectively.

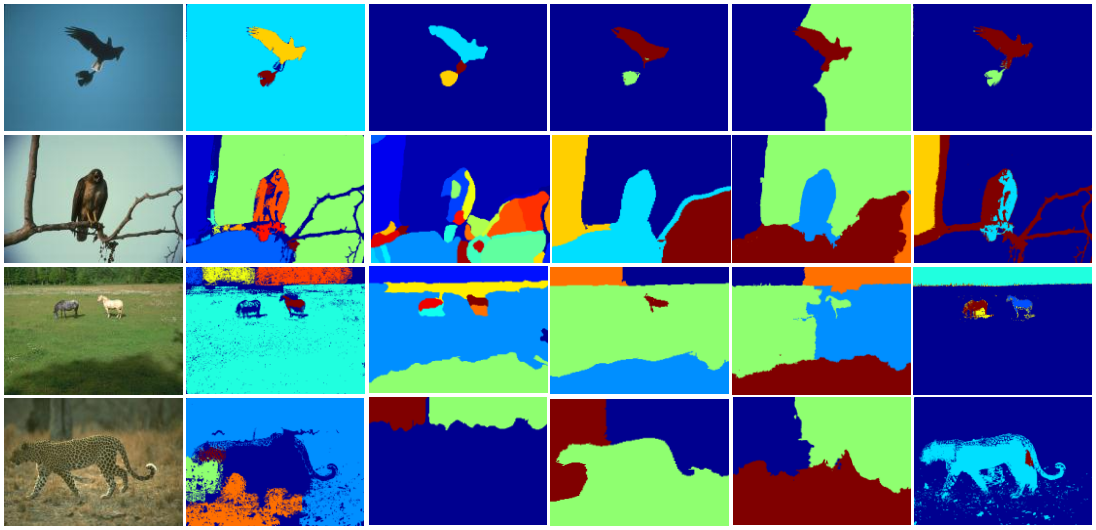


Fig. 4. Examples in category animals. Left to right are original, results of CCTA, CTM, NCut, MNCut, proposed algorithm, respectively.

In our method, we initially make the regions by CCTA, which needs two parameters (k, ε) for the spatial scale and the intensity difference scale respectively. In all experiments, we obtained the regions by four different scales with the parameters $k = \{6,10,14,20\}$ and the ε calculated by Eq (7) according to different k. Our proposed algorithm needs a parameter

K, the number of class to group the pixels. We obtained this parameter by the annotation of images or set it according to contents of the image.

From experiment we can see that CCTA groups the pixels into some classes according to the size of neighborhood which some times is difficult to be defined by user. On the other hand, CCTA produces trivial segments which affect the quality of the algorithm. MCCTSGP uses four different scales, each of which obtains segments with large difference, to generate exiting results in **Fig.4-8**. From these results, we can see that MCCTSGP ameliorate the problem that how to choose an appropriate parameter in CCTA to partition different images. Our algorithm, which can capture details in both coarse and fine level, perceptually produces higher-quality segmentations than NCut, the graph-based method, and MNCut, the multiscale method. To compare with CTM, on the one hand, the segmentation generated by MCCTSGP is accurate in finding fine level details in **Fig.4-8**; on the other hand, CTM is more robust to segment the texture images than MCCTSGP in **Fig.8**. It is the reason that for a fixed neighborhood, the pixels that similar to the center pixel are very few in texture image, which makes the seeds distributed scarcely in the texture part in an image. In this case, the region keeps growing until encountering the boundary of the texture. The segment results generated by the proposed algorithm are easy to produce the fragment in the texture part of the input image. To solve this problem, we can choose the smaller parameter which can increase the number of seeds in texture image to reduce the trivial segmentations. More segmented results on BSD database can be found in [26].

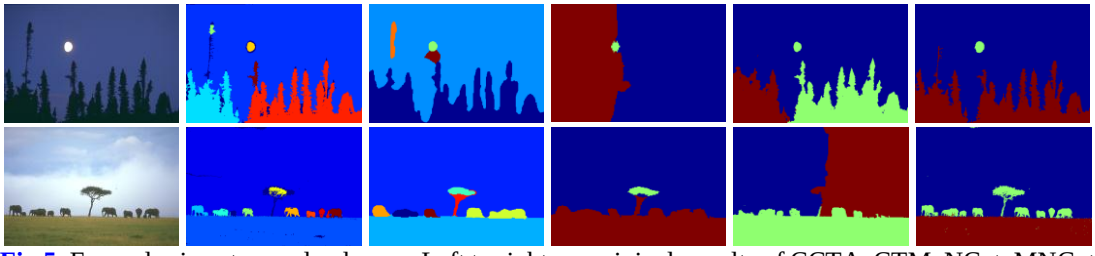


Fig.5. Examples in category landscape. Left to right are original, results of CCTA, CTM, NCut, MNCut, proposed algorithm, respectively.

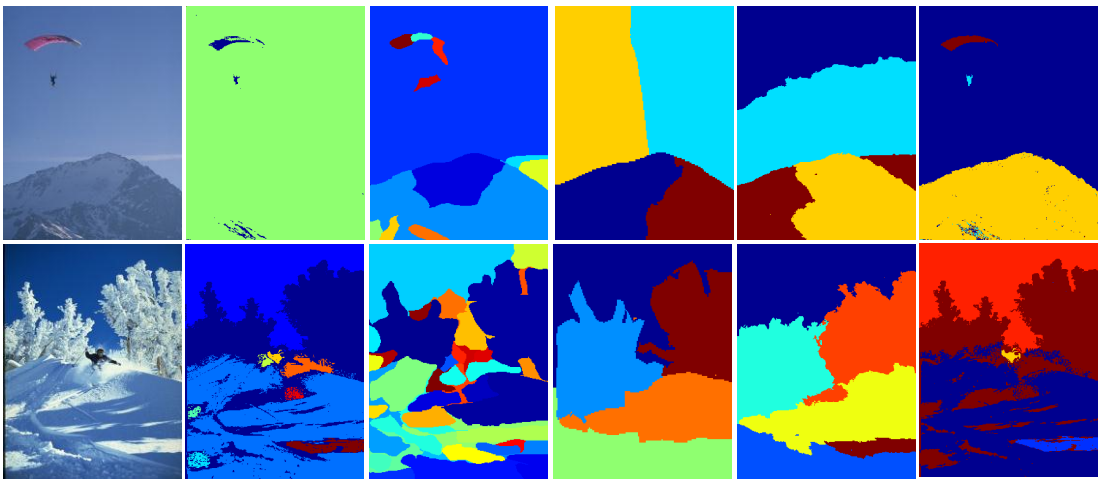


Fig. 6. Examples in category sporting scene. Left to right are original, results of CCTA, CTM, NCut, MNCut, proposed algorithm, respectively.

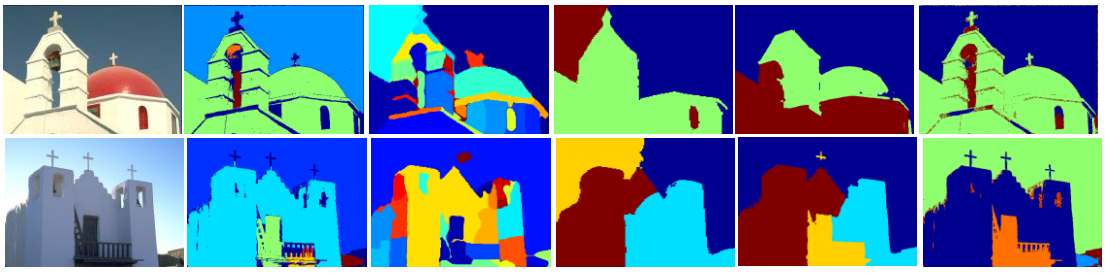


Fig. 7. Examples in category building. Left to right are original, results of CCTA, CTM, NCut, MNCut, proposed algorithm, respectively.

Fig. 8. Examples in category people. Left to right are original, results of CCTA, CTM, NCut, MNCut, proposed algorithm, respectively.

4.2 Quantitative Performance

We now quantitatively compare MCCTSGP against CCTA, CTM, NCut, and MNCut. The comparison is based on four quantitative performance measures:

- The Probabilistic Rand Index (PRI) [19] counts the fraction of pairs of pixels whose labels are consistent between the computed segmentation and the ground truth, averaging across multiple ground truth segmentations to account for scale variation in human perception. PRI ranges between [0, 1], higher is better.
- The Variation of Information (VoI) metric [20] 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 which cannot be explained by the other. VoI ranges between [0, 1), lower is better.
- The Global Consistency Error (GCE) [21] 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, since they could represent the same natural image segmented at different scales. GCE ranges between [0,1], lower is better
- The Boundary Displacement Error (BDE) [22] measures the average displacement error of boundary pixels between two segmented images. Particularly, it defines the error of

one boundary pixel as the distance between the pixel and the closest pixel in the other boundary image. BDE ranges between [0, 1) in the unit of pixel, lower is better.

Table 1 gives the quantitative comparison of MCCTSGP against the other four algorithms on the Berkeley segmentation benchmark. In the experiment, we set the number of segments equal to that of the human subjects. Due to memory issues with the NCut implementation in MATLAB, all images are normalized to have the longest side equal to 160 pixels. Other parameters of NCut and MNCut are set as the same to section 4.1. For CTM, we set the distortion $\gamma = 0.1$. To best evaluate and understand the algorithm proposed by us, two group scale values are tested for MCCTSGP, namely, {6,12,15,20} and {7,10,14,18}, which represented by "▲"and "★"respectively.

Table 1 shows that quantitatively, MCCTSGP outperforms CCTA, NCut, MNCut, and CTM in terms of most indices. MCCTSGP is better than NCut and MNCut in terms of all four indices according to two different group parameters present in last two line of **Table 1**. According to PRI and GCE indices, MCCTSGP has the overwhelming performance than other four methods. It is not supervising that CCTA specially has the best VoI index which penalize undersegmentation more heavily than oversegmentation. Our algorithm inherits this character of CCTA when comparing with other three algorithms. For BDE index, MCCTSGP obtains exciting performance. This comparison proves that our segmentations have less error in terms of boundary displacement with respect to the ground truth. When comparing to two performances both generated by our method, it is suggest that MCCTSGP can obtain similar results through different parameters. Therefore, the proposed algorithm can ameliorate the dependence of parameter k, the spatial scale, in CCTA.

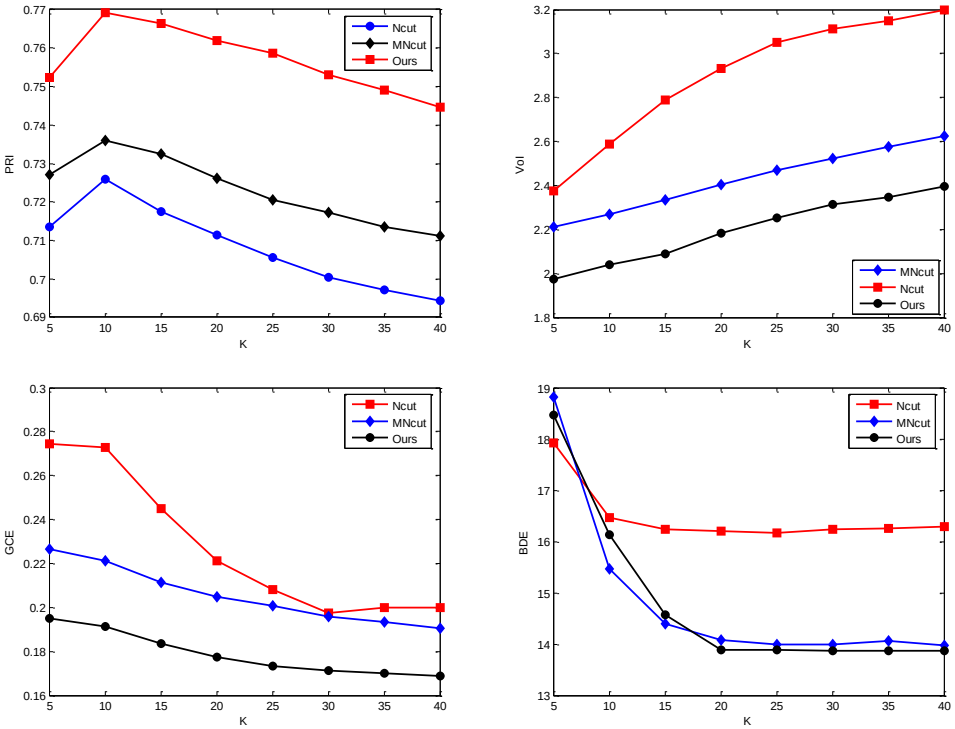


Fig. 9. Our statistics on Probabilistic Rand Index (PRI), Variation of Information (VoI), Global Consistency Error (GCE), and Boundary Displacement Error (BDE) over the Berkeley database with K = {5, 10, 15, ..., 40}, compared with NCut and MNCut.

Table 1. Average performance on Berkley segmentation database (bold indicates best of all the algorithms).

Method/Score	PRI	VoI	GCE	BDE
NCut[7]	0.7297	2.7085	0.2504	16.4774
CCTA[14]	0.6982	**1.7093**	0.1764	14.7073
MNCut[11]	0.7328	2.3678	0.2310	15.2435
CTM[18]	0.7666	2.9549	0.1868	**14.6081**
Our method▲	0.7845	1.8274	**0.1732**	14.6209
Our method★	**0.7928**	1.8149	0.1757	14.6531

To evaluate the effect of parameter K, the number of segments, on MCCTSGP, we run our algorithm by varying K from 5 to 40 (in steps of five) as compared to NCut and MNCut. **Fig.9** gives the average scores of three methods above. These comparisons present that our algorithm outperforms MNCut and NCut in all cases except quite small BDE score difference at K =5, 10, 15 in **Fig 9**.

To summarize quantitative comparisons, we notice that none of the algorithms with different parameters is a clear winner in terms of all four indices. However, for considering the information from both multiscale framework and regions generated by segment algorithm, MCCTSGP has more indices better than that of MNCut, which indirectly considers long-range connections in a multiscale framework, and CTM, the region-based method in image segmentation, or near the best index score of them.

Conclusion

In this paper, a new multiscale image segmentation method is proposed. The novel image segmentation algorithm integrates local and global groping cues by multiscale CCTA, and it combines the results of segment obtained by multiscale CCTA into a graph. Moreover, we use the spectral graph partitioning algorithm to cut the graph and segment the image pixels. From the experiments, we can see that our algorithm can produces high-quality segmentation results in natural images. Furthermore, we will combine MCCTSGP with the description of texture method [24][25] for texture images and estimate the number of segments K automatically. Meanwhile, we will plan to investigate the practicability of the proposed image segmentation algorithm to social image retrieval [27][28][29] and Video Annotation [30].

References

[1] Y. Zhang, "Advances in Image and Video Segmentation," *IRM Press*, 2006. Article (CrossRef Link)

[2] A. Sleit *et al*, "Image clustering using color, texture and shape features," *KSII Transaction on internet and information system*, vol.5, no.1, pp.211-227, 2011. Article (CrossRef Link)

[3] H.D. Cheng, X.H. Jiang, Y. Sun and J.L. Wang, "Color image segmentation: advances and prospects," *Pattern Recognition*, vol.34, pp.2259–2281, 2001. Article (CrossRef Link)

[4] D.A. Forsyth and J. Ponce, "Computer Vision: A Modern Approach," *Prentice-Hall*, Englewood Cliffs, 2002.

[5] N. Bonnet, J. Cutrona and M. Herbin, "A 'no-threshold' histogram-based image segmentation method," *Pattern Recognition*, vol.35, no.10, pp.2319–2322, 2002. Article (CrossRef Link)

[6] E. Navon, O. Miller and A. Averbuch, "Color image segmentation based on adaptive local thresholds," *Image and Vision Computing* ,vol.23 ,no.1, pp.69–85, 2005. Article (CrossRef Link)

[7] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on PAMI*, vol.22, no.8, pp.888–905, 2000. Article (CrossRef Link)

[8] Y.Y.Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," *International Journal of Computer Vision*, vol.70, no.2, pp.109–131, 2006. Article (CrossRef Link)

[9] C. Fowlkes, D. Martin and J. Malik, "Learning affinity funcions for image segmentation: combining patch-based and gradient-based approaches," In *Proc. CVPR 2003*, vol.2 , pp.54-61, 2003. Article (CrossRef Link)

[10] S. X. Yu, "Segmentation using multiscale cues," In *Proc. CVPR 2004*, pp.247-254, 2004. Article (CrossRef Link)

[11] T. Cour, F. Benezit and J. Shi, "Spectral segmentation with multiscale graph decomposition," In *Proc. CVPR 2005*, vol.2, pp.1124-1132, 2005. Article (CrossRef Link)

[12] X. S. Hua, C. Zhang and L. Quan, "Normalized tree partitioning for image segmentation," In *Proc. CVPR 2008*, pp.1-8, 2008. Article (CrossRef Link)

[13] S.R.Rao, H.Mobahi, A. Y. Yang, S. S. Sastry and Y. Ma, "Natural image segmentation with adaptive texture and boundary encoding," In *Proc. ACCV 2009*, pp.135-146, 2009. Article (CrossRef Link)

[14] Ding J, Ma R and Chen S, "A Scale-based coherence connected tree algorithm for image segmentation," *IEEE Trans. on Image Proc.*, vol.17, no.2, pp.204−216, 2008. Article (CrossRef Link)

[15] A. Ng, M. Jordan and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in Neural Information Processing Systems*, pp.849-856, 2002. Article (CrossRef Link)

[16] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," In *Proc.21th Int'l Conf. Machine Learning*, pp.281-288, 2004. Article (CrossRef Link)

[17] Y. Haxhimusa and W. Kropatsch, "Segmentation graph hierarchies," In *Proc. Structural, Syntactic, and Statistical Pattern Recognition*, vol.3138, pp.343–351, Aug.2004. Article (CrossRef Link)

[18] Allen Yang, John Wright, Yi Ma and Shankar Sastry, "unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol.110, pp.212-225, 2008. Article (CrossRef Link)

[19] C. Pantofaru and M. Hebert, "A comparison of image segmentation algorithms," *Tech. Rep.* 2005. Article (CrossRef Link)

[20] M. Meila, "Comparing clusterings: an axiomatic view", In *Proc.21th Int'l Conf. Machine Learning*, pp.577–584, 2005. Article (CrossRef Link)

[21] D. Martin, C. Fowlkes, D. Tal and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," In *Proc. of IEEE International Conference on Computer Vision*, pp.416–423, 2001. Article (CrossRef Link)

[22] J.Freixenet, X.Munoz, D.Raba, J. Marti and X. Cuff, "Yet another survey on image segmentation: region and boundary information integration", In *Proc. of European Conference on Computer Vision*, pp.408–422, 2002. Article (CrossRef Link)

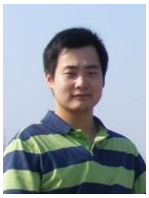
[23] Tae Hoon Kim, Kyoung Mu Lee and Sang Uk Lee, "Learning full pairwise affinities for spectral segmentation," In *Proc. of CVPR 2010*, pp.2101-2108, 2010. Article (CrossRef Link)

[24] Heekyung Yang and Kyungha Min, "Texture-based hatching for color image and video," *KSII Transaction on internet and information system*, vol.5, no.4, pp.763-781, 2011. Article (CrossRef Link)

[25] I. S. Ahmad, "Texture-based image indexing and retrieval using formal concept analysis," *KSII Transaction on internet and information system*, vol.2, no.3, pp.150-170, 2008. Article (CrossRef Link)

[26] Jingbo Zhou, Jun Yin and Zhong Jin, "A multiscale CCTA plus spectral graph partitioning for image segmentation," In *CCPR2010*, vol.1, pp.311-315, 2010. Article (CrossRef Link)

[27] Xiaobai Liu *et al*, "Label to region by bilayer sparsity priors," In *Proc. of the 17th ACM international conference on Multimedia*, 2009. Article (CrossRef Link)
[28] M.Wang, X.S.Hua, J.H.Tang and R.C.Hong, "Beyond distance measurement: constructing neighborhood similarity for video annotation," *IEEE Trans. on MM*, vol.11, no.3, pp.465-476, Apr.2009. Article (CrossRef Link)
[29] M.Wang *et al*, "Unified video annotation via multigraph learning," *IEEE Trans. on TCSVT*, vol.19, no.5, pp.733-746, May.2009. Article (CrossRef Link)
[30] K.Yang *et al*, "Tag tagging: towards more descriptive keywords of image content," *IEEE Trans. on MULTIMEDIA*, vol.13, no.4, pp.662-673, May.2011. Article (CrossRef Link)



Jingbo Zhou received the BS degree in school of computer science and technology from Hunan University of science and technology in 2006. He received the MS degree in computer engineering from Nanjing University of Science and Technology (NUST) in 2008. He is currently pursuing the Ph.D. degree with School of Computer Science and Technology, Nanjing University of Science and Technology (NUST). He is on the subject of pattern recognition and intelligence systems. His current research interests include pattern recognition and computer vision.

Shangbing Gao received the BS degree in mathematics from the Northwestern Polytechnical University in 2003. He received the MS degree in applied mathematics from the Nanjing University of Information and Science and Technology in 2006.He is currently pursuing the Ph.D. degree with School of Computer Science and Technology, Nanjing University of Science and Technology (NUST). He is on the subject of pattern recognition and intelligence systems. His current research interests include pattern recognition and computer vision.

Zhong Jin received his BS degree in mathematics, MS degree in applied mathematics and PhD degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology(NUST), China, in 1982, 1984 and 1999, respectively. He is a professor in the Department of Computer Science, NUST. He had a stay of 15 months as research assistant at the Department of Computer Science and Engineering, Chinese Universityof Hong Kong from 2000 to 2001. He visited the Laboratoire HEUDIASYC, Universite de Technologie de Compiegne, France, from October 2001 to July 2002. He was at the Centre de Visio per Computador, Universitat Autonoma de Barcelona, Spain, as the Ramon y Cajal Program Research Fellow from September 2005 to October 2005. His current interests are in the areas of pattern recognition, computer vision, face recognition, facial expression analysis and content-based image retrieval.