

# Evaluating performance of image segmentation criteria and techniques

Dorit S. Hochbaum · Cheng Lyu · Erik Bertelli

Received: 16 April 2012 / Accepted: 17 October 2012 / Published online: 11 December 2012  
© Springer-Verlag Berlin Heidelberg and EURO - The Association of European Operational Research Societies 2012

**Abstract** The image segmentation problem is to delineate, or segment, a salient feature in an image. As such, this is a bipartition problem with the goal of separating the foreground from the background. An NP-hard optimization problem, the *Normalized Cut* problem, is often used as a model for image segmentation. The common approach for solving the normalized cut problem is the *spectral method* which generates heuristic solutions based upon finding the Fiedler eigenvector. Recently, Hochbaum (IEEE Trans Pattern Anal Mach Intell 32(5):889–898, 2010) presented a new relaxation of the normalized cut problem, called normalized cut' problem, which is solvable in polynomial time by a combinatorial algorithm. We compare this new algorithm with the spectral method and present experimental evidence that the combinatorial algorithm provides solutions which better approximate the optimal normalized cut solution. In addition, the subjective visual quality of the segmentations provided by the combinatorial algorithm greatly improves upon those provided by the spectral method. Our study establishes an interesting observation about the normalized cut criterion that the segmentation which provides the subjectively best visual bipartition rarely corresponds to the segmentation which minimizes the objective function value of the normalized cut problem. We conclude that modeling the image segmentation problem as normalized cut criterion might not be appropriate. Instead, normalized cut'

---

Research was supported in part by NSF award no. DMI-0620677, CMMI-1200592 and CBET-0736232.

---

D. S. Hochbaum (✉) · C. Lyu · E. Bertelli  
Department of Industrial Engineering and Operations Research,  
University of California, Berkeley, USA  
e-mail: hochbaum@ieor.berkeley.edu

C. Lyu  
e-mail: chenglu@berkeley.edu

E. Bertelli  
e-mail: erikbertelli@berkeley.edu

not only provides better visual segmentations but is also solvable in polynomial time. Therefore, normalized cut' should be the preferred segmentation criterion for both complexity and good segmentation quality reasons.

**Keywords** Image segmentation · Normalized cut · Network flow · Combinatorial algorithm · Spectral method

**Mathematics Subject Classification** 90-08 Computational methods · 90B10 Network models, deterministic · 90C27 Combinatorial optimization

## Introduction

Image segmentation is fundamental in computer vision (Shapiro and Stockman 2001). It is used in numerous applications, such as in medical imaging (Pham et al. 2000; Dhawan 2003; Hosseini et al. 2010; Roobottom et al. 2010), and is also of independent interest in clustering (Coleman and Andrews 1979; Pappas 1992; Wu and Leahy 1993; Shi and Malik 2000; Xing and Jordan 2003; Tolliver and Miller 2006). The image segmentation problem is to delineate, or segment, a salient feature in an image. As such, this is a bipartition problem with the goal of separating the foreground from the background. It is not obvious how to construct a quantitative measure for optimizing the quality of a segmentation. The common belief is that normalized cut (NC) criterion (Shi and Malik 2000) is a good model for achieving high-quality image segmentation and it is often used.

The normalized cut criterion uses similarity weights that quantify the similarity between pairs of pixels. These weights are typically set to be a function of the difference between the color intensities of the pixels. Such functions are increasing with the perceived similarity between the pixels. Even though the use of normalized cut is common, it is an NP-hard problem (Shi and Malik 2000) and heuristics and approximation algorithms have been employed (Shi and Malik 2000; Xing and Jordan 2003; Dhillon et al. 2004; Tolliver and Miller 2006; Dhillon et al. 2007). The most frequently used method for obtaining an approximate solution for the normalized cut problem is the spectral method that finds the Fiedler eigenvector (Shi and Malik 2000).

Hochbaum (2010) presented a new relaxation of the normalized cut problem, called the normalized cut' problem (NC'). The normalized cut' problem was shown in Hochbaum (2010) to be solved in polynomial time with a combinatorial (flow-based) algorithm. In addition, Hochbaum (2010, 2012) introduces a generalization of normalized cut, called the q-normalized cut problem (q-NC). For the q-normalized cut problem, there are, in addition to the similarity weights, also pixel weights. The pixel weights could be a function of some pixel's feature other than color intensity. The combinatorial algorithm that solves the normalized cut' problem was shown to generalize, with the same complexity, to a respective relaxation problem q-normalized cut' (q-NC') (Hochbaum 2010, 2012). It is also shown in Hochbaum (2012) that the spectral method heuristic for the normalized cut problem extends to a respective heuristic for q-normalized cut.

Unlike the combinatorial algorithm's solution, the spectral method's solution is a real eigenvector, rather than a discrete bipartition. In order to generate a bipartition, a

method, called the *threshold* technique, is commonly used. For a given *threshold* value, all pixels that correspond to entries of the eigenvector that exceed this threshold are set in one side of the bipartition, and the remaining pixels constitute the complement set. For further improvement, the *spectral sweep technique* selects, among all possible thresholds, the one that gives a smallest objective value for the respective normalized cut objective. A different technique, utilized by Yu and Shi (2003) and Cour et al. (2011), generates a bipartition from the Fiedler eigenvector which is claimed to give a superior approximation to the objective value of the respective normalized cut problem. This different method will be referred to as *Shi's code* in the remainder of the paper. Our experimental study implements both the spectral sweep technique and the Shi's code for the spectral method.

In this paper, we provide a detailed experimental study comparing the combinatorial algorithm to the spectral method, in terms of approximating the optimal value of both the normalized cut and the q-normalized cut criteria, quality of visual segmentation, and in terms of running times in practice.

To compare the approximation quality, we evaluate the objective functions of the normalized cut and q-normalized cut problems for the solutions resulting from solving the normalized cut' problem and the spectral method. These solutions are bipartitions, and hence feasible solutions for the normalized cut and q-normalized cut problems.

To evaluate visual quality, we view the feature(s) that are delineated by the bipartition solutions. The evaluation is inevitably subjective. The manner in which we evaluate the visual quality is explained in detail in "Visual segmentation quality evaluation".

For running time comparisons, we test the methods not only for the benchmark images given in  $160 \times 160$  resolution but also for higher image resolutions.

The main findings of the experimental study presented here are:

1. The combinatorial algorithm solution is a better approximation of the optimal objective value of the normalized cut problem than the solution provided by the spectral method. This dominance of the combinatorial algorithm holds for both the spectral sweep technique and the Shi's code's. This is discussed in "Quantitative evaluation for objective function values".
2. The discretizing technique used in Shi's code to generate a bipartition from the eigenvector is shown here to give results inferior to those of the spectral sweep technique, in terms of approximating the objective value of the respective normalized cut problem. This is displayed in "Comparing approximation quality of SWEEP and COMB".
3. The visual quality of the segmentation provided by the combinatorial algorithm is far superior to that of the spectral method solutions, as presented in "Visual segmentation quality evaluation".
4. Shi's code includes a variant that uses similarity weights derived with *intervening contour* (Leung and Malik 1998; Malik et al. 2001). The visual quality resulting from segmentation with the intervening contour code is much better than the other spectral segmentations. Yet, the combinatorial algorithm with standard similarity (exponential similarity) weights delivers better visual results than Shi's code with intervening contour ("Visual segmentation quality evaluation"). The

combinatorial algorithm does not work well with intervening contour similarity weights since these weights tend to be of uniform value. A detailed discussion of this phenomenon is provided in “Comparing instances with intervening contour similarity weights: comparing SHI-NC-IC with COMB-NC-IC and SHI-qNC-IC with COMB-qNC-IC”.

5. Our study compares the visual quality of segmentations resulting from the q-normalized cut' criterion with those resulting from the normalized cut' criterion in “Visual segmentation quality evaluation”. (We use *entropy* for pixel weights in the q-normalized cut' instances.) The results show that q-normalized cut' often provides better visual segmentation than normalized cut'. Therefore, for applications such as medical imaging, where each pixel is associated with multiple features, these features can be used to generate characteristic node weights, and q-normalized cut' would be a better criterion than normalized cut'.
6. Over the benchmark images of size  $160 \times 160$ , the combinatorial algorithm runs faster than the spectral method by an average speedup factor, for the normalized cut objective, of 84. Furthermore, the combinatorial algorithm scales much better than the spectral method: the speedup ratio provided by the combinatorial algorithm compared to the spectral method grows substantially with the size of the image, increasing from a factor of 84 for images of size  $160 \times 160$  to a factor of 5,628 for images of size  $660 \times 660$ . The details are discussed in “Running time comparison between the spectral method and the combinatorial algorithm”.
7. For normalized cut' we get a collection of nested bipartitions as a bi-product of the combinatorial algorithm (Hochbaum 2010, 2012). The best visual bipartition and the best normalized cut objective value bipartition are chosen among these nested bipartitions. Our study results show that in most cases the best visual bipartition does not coincide with the bipartition that gives the best objective value of the normalized cut (or q-normalized cut) problem. (The details are discussed in “Visual segmentation quality evaluation”) Therefore, normalized cut, in spite of its popularity, is not a good segmentation criterion. Normalized cut' improves on normalized cut not only in complexity (from NP-hard to polynomial time solvable problem) but also in segmentation quality delivered.

The paper is organized as follows: “Notations and problem definitions” presents the notations employed. The detailed settings of the experiment are discussed in “Experimental setting”. In “Assessing quality of seed selection methods of COMB”, we first evaluate the effect of different selection of seeds—an important component of the combinatorial algorithm—in approximating the optimal objective values of the normalized cut and q-normalized cut problems. Then in “Running time comparison between the spectral method and the combinatorial algorithm”, we evaluate and compare the running times of the spectral method and the combinatorial algorithm. In “Quantitative evaluation for objective function values”, we present the quantitative evaluation of the spectral method and the combinatorial algorithm, in terms of the quality of approximation to the optimal objective values of the normalized cut and q-normalized cut problems. “Visual segmentation quality evaluation” provides a comparison of the visual results for the spectral method versus the combinatorial algorithm followed by concluding remarks in “Conclusions”.

## Notations and problem definitions

In image segmentation an image is formalized as an undirected weighted graph  $G = (V, E)$ . Each pixel in the image is represented as a node in the graph. A pair of pixels is said to be *neighbors* if they are adjacent to each other. The common neighborhoods used in image segmentation are the *4-neighbor* and *8-neighbor* relations. In the 4-neighbor relation, a pixel is a neighbor of the two vertically adjacent pixels and two horizontally adjacent pixels. The 8-neighbor relation adds also the four diagonally adjacent pixels. Every pair of neighbors  $i, j \in V$  is associated with an edge  $[i, j] \in E$ . Each edge  $[i, j] \in E$  has a weight  $w_{ij} \geq 0$  representing the similarity between pixel node  $i$  and  $j$ . We adopt the common notation that  $n = |V|$  and  $m = |E|$ .

For two subsets  $V_1, V_2 \subseteq V$ , we define  $C(V_1, V_2) = \sum_{[i,j] \in E, i \in V_1, j \in V_2} w_{ij}$ . A bipartition of a graph is called a *cut*,  $(S, \bar{S}) = \{[i, j] \in E | i \in S, j \in \bar{S}\}$ , where  $\bar{S} = V \setminus S$  is the complement of set  $S$ . The *cut capacity* is  $C(S, \bar{S})$ . Each node has a weight  $d(i) = \sum_{[i,j] \in E} w_{ij}$  which is the sum of the weights of its incident edges. For a set of nodes  $S$ ,  $d(S) = \sum_{i \in S} d(i)$ . A node may have also an arbitrary nonnegative weight associated with it,  $q(i)$ . For a set of nodes  $S \subseteq V$ ,  $q(S) = \sum_{i \in S} q(i)$ .

Let  $\mathbf{D}$  be a diagonal  $n \times n$  matrix with  $\mathbf{D}_{ii} = d(i) = \sum_{[i,j] \in E} w_{ij}$ . Let  $\mathbf{W}$  be the weighted node–node adjacency matrix of the graph, where  $\mathbf{W}_{ij} = \mathbf{W}_{ji} = w_{ij}$ . The matrix  $\mathcal{L} = \mathbf{D} - \mathbf{W}$  is called the Laplacian of the graph.

The mathematical formulations of the normalized cut and q-normalized cut problems are:

**Normalized cut** (Shi and Malik 2000):

$$\text{NC}_G = \min_{\emptyset \subset S \subset V} \frac{C(S, \bar{S})}{d(S)} + \frac{C(S, \bar{S})}{d(\bar{S})}, \quad (1)$$

**q-Normalized cut** (Hochbaum 2012):

$$\text{q-NC}_G = \min_{\emptyset \subset S \subset V} \frac{C(S, \bar{S})}{q(S)} + \frac{C(S, \bar{S})}{q(\bar{S})}. \quad (2)$$

A relaxation of these problems, introduced by Hochbaum (2010, 2012), omits the second term in the objective value. We call these relaxations normalized cut' and q-normalized cut' problems, respectively:

**Normalized cut'** (Hochbaum 2010):

$$\text{NC}'_G = \min_{\emptyset \subset S \subset V} \frac{C(S, \bar{S})}{d(S)}, \quad (3)$$

**q-Normalized cut'** (Hochbaum 2010, 2012):

$$\text{q-NC}'_G = \min_{\emptyset \subset S \subset V} \frac{C(S, \bar{S})}{q(S)}. \quad (4)$$

It is shown in Hochbaum (2010, 2012) that

$$\min_{\emptyset \subset S \subset V} \frac{C(S, \bar{S})}{C(S, S)} \quad (5)$$

defined in Sharon et al. (2006) is equivalent to (3) in that both have the same optimal solution. Problem (5) is a criterion characterizing a good image bipartition by two goals. One requires the salient region segmented to be dissimilar from the rest of the image, or formally to have a small value for  $C(S, \bar{S})$ . The second goal is to have the pixels in the segmented region as similar to each other as possible. This second goal is to have a large value for  $C(S, S)$ .

The normalized cut and q-normalized cut problems are NP-hard (Shi and Malik 2000; Hochbaum 2012). The combinatorial algorithm presented in Hochbaum (2010) solves the normalized cut and q-normalized cut problems approximately by solving their relaxations, normalized cut' and q-normalized cut' problems, respectively. Both the normalized cut' and the q-normalized cut' problems are polynomial time solvable by the combinatorial algorithm.

A bound on the relation between the spectral method solution and  $NC_G$

The bounds on the Fiedler eigenvalue were developed for a problem closely associated with normalized cut. This problem, devised by Cheeger (1970) and called the *Cheeger constant* problem (e.g., Chung 1997), is a “half-version” of normalized cut. If the balance constraint  $d(S) \leq d(V)/2$  is added, the formulation of the Cheeger constant problem is

$$h_G = \min_{\emptyset \subset S \subset V, d(S) \leq d(V)/2} \frac{C(S, \bar{S})}{d(S)}, \quad (6)$$

where  $h_G$  is called the Cheeger constant of the graph  $G$ . Like the normalized cut problem, the Cheeger constant problem is also NP-hard (e.g., Chung 1997). For any (undirected) graph  $G$ , the Cheeger inequality states that its Cheeger constant  $h_G$  is bounded by the second smallest eigenvalue of the (normalized) Laplacian of  $G$ , the Fiedler eigenvalue  $\lambda_1$ :  $\frac{\lambda_1}{2} \leq h_G \leq \sqrt{2\lambda_1}$  (Cheeger 1970; Chung 1997). In addition, the solution resulting from applying the spectral sweep technique to the Fiedler eigenvector evaluated by the objective of the Cheeger constant problem is of value at most  $2\sqrt{h_G}$  (Chung 2007). On the other hand, it is easily shown that  $h_G$  and  $NC_G$  satisfy:  $\frac{1}{2}NC_G \leq h_G \leq NC_G$  (e.g., Hochbaum 2012). These (approximation) bounds for the Cheeger constant (with respect to  $\lambda_1$  and the sweep solution) can be easily transformed to corresponding bounds for the normalized cut objective. Therefore, the spectral method approximately solves the normalized cut (and the Cheeger constant and q-normalized cut) problem by finding the Fiedler eigenvector (Cheeger 1970; Donath and Hoffman 1973; Fiedler 1975; Alon and Milman 1985; Alon 1986; Shi and Malik 2000; Hochbaum 2012).

## Experimental setting

### Edge and node weights

#### *Similarity edge weights*

The benchmark images used here consist of grayscale images. A color intensity value is associated with every pixel, represented as an integer in  $[0, 255]$  in MATLAB. This is normalized and mapped to  $[0, 1]$ . The similarity weight between a pair of pixel nodes is a function of the difference of their color intensities. For  $p_i$  and  $p_j$  the color intensities of two neighboring pixel nodes  $i$  and  $j$ , the *exponential similarity weight* is defined as

$$w_{ij} = e^{-\alpha|p_i - p_j|}, \quad (7)$$

where  $\alpha$  can be viewed as amplifying the dissimilarity between two pixels based on the color intensity difference. If  $\alpha$  is too small, then the dissimilarity is not significant enough to reflect the color intensity difference. On the other hand, setting the value of  $\alpha$  to be too large results in all pairs of pixels very dissimilar and therefore color intensity differences are not sufficiently informative. We tested several settings for  $\alpha$  values and found  $\alpha = 100$  works well. In all experiments prepared here  $\alpha$  is set to 100.

Another similarity weight is *intervening contour* introduced in [Leung and Malik \(1998\)](#) and [Malik et al. \(2001\)](#). Intervening contour uses the contour information in an image to characterize the (local) similarity between two pixels that are not necessarily neighboring. If two pixels are on the two different sides of a boundary, their similarity should be small as they are more likely to belong to different segments. In the experiment, we use the intervening contour similarity weight generated by Shi's code. Since Shi's code with intervening contour is considered to generate good segmentation, we compare it to the combinatorial algorithm.

#### *Node weights*

For q-normalized cut, the *entropy* of a pixel is used as its weight. The entropy of an image is a measure of randomness in the image that can be used to characterize the texture of an image. In MATLAB, by default the local entropy value of a pixel is the entropy value of the 9-by-9 neighborhood around the pixel. In our experiment, the entropy of a pixel is computed directly via the MATLAB built-in function `entropyfilt`.

### Image database

We select 20 benchmark images from the Berkeley Segmentation Dataset and Benchmark (<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>). See Fig. 6 in Appendix. The 20 benchmark images are chosen to cover

**Table 1** Three seed selection rules

Method no.	One seed node
1	Manual
2	Max entropy
3	Max group luminance

various segmentation difficulties and have been resized to be  $160 \times 160$  for testing since it is the default size in Shi's code.

## Implementation of the combinatorial algorithm

### *Seed selection*

The combinatorial algorithm requires to designate a node as a seed in one set and a node as a seed in the other set to guarantee that both sets are nonempty (Hochbaum 2010). On the other hand, the delineation of foreground versus background depends on the interpretation of what is the main feature. This is not self evident and the purpose of the seeds is to have one seed indicating a pixel in the foreground and the other seed indicating a pixel in the background.

Theoretically, in order to obtain the optimal solutions to the normalized cut' and q-normalized cut' problems, all possible pairs of seed nodes should be considered. This increases the complexity of the combinatorial algorithm by a factor of  $O(n)$ . To avoid this added complexity we devise a test for automatically choosing the seed nodes.

Two automatic seed selection rules are introduced here. The first rule is to select the pixel with the maximum entropy as a seed node in one set. The second rule is to select the pixel with the maximum *group luminance* value as a seed node in one set. The group luminance value is defined for pixels not on the boundary. For every pixel  $i$ , the group luminance value of pixel  $i$  is the average of color intensities of the nine neighboring pixels in the  $3 \times 3$  region centering at  $i$ . Intuitively, if a pixel has a greater group luminance value, that pixel and its surrounding pixels are more likely to be in the same segment. The other seed node is any arbitrarily selected node in the complement region to the one occupied by the first seed node. We compare the two automatic seed selection methods with a manual selection of both seed nodes (Table 1).

For each pair of seed nodes, the combinatorial algorithm is run twice where in the second run the two seed nodes are exchanged between the two sets. Therefore, for each image the combinatorial algorithm is executed six times, for the three different seed selection rules.

### *Nested cuts*

Each run of the combinatorial algorithm for a pair of seed nodes, to either the normalized cut or the q-normalized cut problem, produces a series of nested cuts. This is because the combinatorial algorithm uses a parametric minimum cut solver as a

subroutine (Hochbaum 2010, 2012). The parametric minimum cut problem can be solved efficiently. Theoretically, it is shown in Gallo et al. (1989) and Hochbaum (2008) that the running time to solve the parametric minimum cut problem is only a small constant factor of the time to solve a single instance of the minimum cut problem. We implement Hochbaum's pseudoflow algorithm in Hochbaum (2008) as the parametric minimum cut solver. The implementation is described in Chandran and Hochbaum (2009) and the code is available online at Chandran and Hochbaum (2012).

The number of the nested cuts is typically 5–15. The combinatorial algorithm stores the visual segmentations by all the nested cuts, which enables to choose the one that is deemed (subjectively) best visually. The combinatorial algorithm also automatically selects the bipartition which gives the smallest objective values of the normalized cut or q-normalized cut problem among the nested cuts.

### Implementation of the spectral method

The eigenvector solver subroutine of Shi's code, named `eigs2`, is based on the MATLAB built-in eigenvector solver `eigs` with some modifications. Shi's code applies the following two operations to the Laplacian matrix  $\mathcal{L} = \mathbf{D} - \mathbf{W}$ :

1. Sparsifying operation: It rounds to 0 small values of  $w_{ij}$  where the "small" is determined by some threshold value. The default threshold value in Shi's code is  $10^{-6}$ .
2. Offset operation: It adds a constant (1 by default) to  $\mathbf{D}_{ii} = d(i)$  ( $i = 1, \dots, n$ ). It also adds a value to each diagonal entry of the  $\mathbf{W}$  matrix. This value for entry  $\mathbf{W}_{ii}$  is 0.5 plus a quantity that estimates the round-off error for row  $i$ ,  $e(i) = d(i) - \sum_{j=1}^n w_{ij}$ .

In our experiment, we exclude the above two operations in Shi's code to compare it with the combinatorial algorithm, which contains no any sparsifying or offset operations.

The spectral sweep technique uses the Fiedler eigenvector from Shi's code (without the two operations) and then chooses the best bipartition threshold as described in "Introduction".

### Algorithm, optimization criterion, and similarity classifications and nomenclatures

Each experimental set is characterized by the choice of algorithm, the choice of optimization objective, and the choice of similarity weight definition. For the algorithm, we choose among the combinatorial algorithm, *COMB*, Shi's code, *SHI* and the spectral sweep technique, *SWEEP*. For the optimization objective, we choose among normalized cut, *NC*, and q-normalized cut, *qNC*. For the similarity weight definition, we choose among the exponential similarity weights, *EXP*, and the intervening contour similarity weights, *IC*. The format of *Algorithm-Criterion-Similarity* is used to represent an experimental set.

For each choice of optimization objective and similarity weight definition, the combinatorial algorithm outputs a series of nested cuts for a pair of seeds (see “Nested cuts”), among which the cut that gives the smallest *objective value* of NC or q-NC is selected. The pairs of seeds are selected according to the automatic seed selection criterion, including both the maximum entropy criterion and the maximum group luminance criterion, described in “Seed selection”. The numerically best cut is selected among the four series of corresponding nested cuts. The segmentation of the selected cut is considered as the output of the combinatorial algorithm and the objective value of NC or q-NC of the selected cut is considered as the objective value output by the combinatorial algorithm.

We test the following experimental sets:

COMB-NC-EXP  
 COMB-qNC-EXP  
 COMB-NC-IC  
 COMB-qNC-IC  
 SHI-NC-EXP  
 SHI-qNC-EXP  
 SHI-NC-IC  
 SHI-qNC-IC  
 SWEEP-NC-EXP  
 SWEEP-qNC-EXP  
 SWEEP-NC-IC  
 SWEEP-qNC-IC.

### Assessing quality of seed selection methods of COMB

In this section, we evaluate the three seed selection methods introduced in “Seed selection” for the combinatorial algorithm in terms of approximating the objective values of NC and q-NC. We evaluate the objective values of NC and q-NC for each of the solutions provided by the three seed selection methods and count the number of images on which each seed selection method gives the smallest objective values of NC and q-NC. Table 2 gives the percentage of the number of images, out of the 20 images, in which each method gets the best values for NC and q-NC, respectively. The exponential similarity weight is used in both cases.

**Table 2** Portions of each seed selection method in yielding the smallest NC and q-NC objective values

	Method 1 (manual) (%)	Method 2 (max-entropy) (%)	Method 3 (max-group luminance) (%)
NC	33.33	37.04	29.63
q-NC	26.09	47.83	26.09

The results given in Table 2 show that method 2 (max entropy) is best for NC and q-NC. This indicates that the maximum entropy is a good seed selection method for image segmentation.

Since method 3 (max group luminance) is automatic, and also works well, we derive an automatic seed selection method which combines methods 2 and 3. This is done by running the combinatorial algorithm for the pairs of seeds generated by methods 2 and 3, and the output is the one corresponds to the best of these four values. The automatic seed selection method is best 66.67 % of the time for NC and 73.92 % of the time for q-NC. This improves a great deal on method 1, where the two seeds are selected manually.

As a result of the comparison, in the following comparisons the cut that gives the smallest objective values of NC or q-NC of COMB is selected from the four series of nested cuts corresponding to the four pairs of seeds selected according to the automatic seed selection method defined above.

### Running time comparison between the spectral method and the combinatorial algorithm

In order to provide a fair comparison of the running times of the two algorithms, we disregard the input and output processing parts of each algorithm. We run both algorithms on a 2010 Apple MacBook Pro Laptop (2.4GHz Intel Core 2 Duo processor and 4GB of 1067 MHz DDR3). The running times of the two algorithms over all the 20 benchmark images with size  $160 \times 160$  are reported in Table 3. The exponential similarity weights are used and both NC and q-NC objectives are applied.

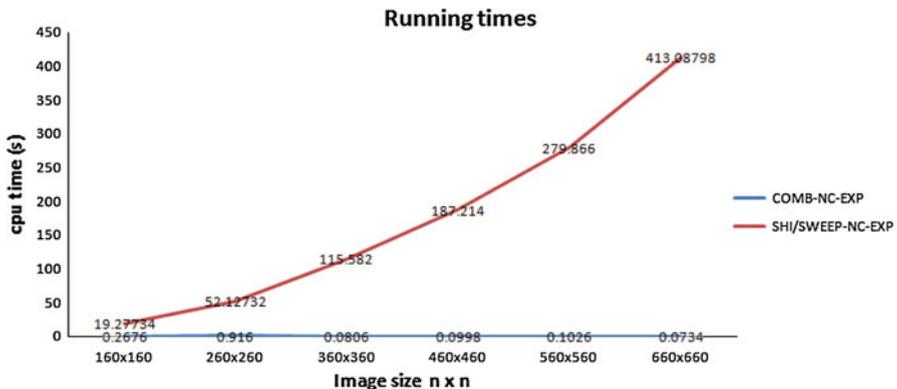
Table 3 shows that the combinatorial algorithm runs much faster than the spectral method for the NC objective by an average speedup factor of 84. For the q-NC objective, in most cases the combinatorial algorithm is still faster. The same comparison results also apply to the case of intervening contour similarity weights. It is not clear why the spectral method runs so much faster for q-NC than NC. We note, however, that the results delivered by the spectral method for q-NC are dramatically inferior to those provided by the combinatorial algorithm, both in terms of approximating the optimal objective value of q-NC (Figs. 2, 4), and in terms of visual quality (“Visual segmentation quality evaluation”).

We further evaluate the scalability of the two algorithms by creating input sequences of images each based on one image at increasing resolutions: we employ six different image sizes:  $160 \times 160$ ,  $260 \times 260$ ,  $360 \times 360$ ,  $460 \times 460$ ,  $560 \times 560$  and  $660 \times 660$ . For every image size we run the two algorithms on 5 of the 20 benchmark images (Images 4, 8, 12, 16 and 20) and average the running times over the five images. The running times of the spectral method and the combinatorial algorithm with these different image sizes are plotted in Fig. 1.

Figure 1 shows that as the input size increases, the running time of the spectral method grows significantly faster than that of the combinatorial algorithm, with an average speedup factor increasing from 84 for images of size  $160 \times 160$  to 5,628 for images of size  $660 \times 660$ . The running time of the combinatorial algorithm appears insensitive to changes in the input size. Interestingly, we observe that the running time

**Table 3** Running times of SHI/SWEEP-NC-EXP, COMB-NC-EXP, SHI/SWEEP-qNC-EXP and COMB-qNC-EXP

Time(s)	SHI/SWEEP-NC-EXP	COMB-NC-EXP	SHI/SWEEP-qNC-EXP	COMB-qNC-EXP
Image 1	19.5905	0.468	0.67045	0.595
Image 2	20.5193	0.029	0.56141	0.047
Image 3	19.9446	0.090	0.34875	0.208
Image 4	19.0059	0.186	0.79889	0.287
Image 5	20.4605	0.526	0.32494	0.810
Image 6	19.2924	0.138	0.95232	0.543
Image 7	20.4066	0.214	0.65428	0.505
Image 8	17.6540	0.390	0.82868	0.440
Image 9	17.5387	0.150	0.36554	0.241
Image 10	17.3702	0.119	0.37487	0.206
Image 11	19.6832	0.024	0.34565	0.162
Image 12	17.7123	0.226	0.39409	0.468
Image 13	17.3662	0.034	0.58339	0.046
Image 14	17.5793	0.456	0.54604	0.615
Image 15	18.9113	0.376	0.35412	0.617
Image 16	19.9957	0.125	0.78376	0.329
Image 17	19.6383	0.068	0.53126	0.073
Image 18	17.5165	0.263	0.56119	0.430
Image 19	20.3009	0.455	0.54215	0.575
Image 20	22.3611	0.227	0.54404	0.267

**Fig. 1** Running times of SHI/SWEEP-NC-EXP and COMB-NC-EXP for images with increasing resolutions

of the combinatorial algorithm does not increase with the size of the image. This is because for these images in higher resolutions, the number of breakpoints is smaller and therefore there are fewer updates required between consecutive breakpoints Hochbaum (2012).

## Quantitative evaluation for objective function values

In this section, we compare the performance of the spectral method and the combinatorial algorithm in terms of how well they approximate the optimal objective values of the normalized cut and q-normalized cut problems. In “Comparing approximation quality of SHI and COMB”, we compare SHI with COMB and in “Comparing approximation quality of SWEEP and COMB”, we compare SWEEP with COMB. Both exponential similarity weights and intervening contour weights are used in the comparisons.

In order to compare the performance of the spectral method, either SHI or SWEEP, with COMB in approximating the optimal objective value of NC or q-NC, we compute a ratio of the objective value of NC or q-NC generated by the spectral method to the corresponding objective value generated by COMB. If the ratio is greater than 1, it indicates that COMB performs better than the spectral method, while the ratio smaller than 1 is indicative of the spectral method having better performance. If the ratio is smaller than 1, its reciprocal characterizes the improvement of the spectral method on COMB in approximating the optimal objective value of NC or q-NC.

### Comparing approximation quality of SHI and COMB

*Comparing instances with exponential similarity weights: comparing SHI-NC-EXP with COMB-NC-EXP and SHI-qNC-EXP with COMB-qNC-EXP*

Tables 4 and 5 show the ratios over the 20 benchmark images using exponential similarity weights. Tables 4 and 5 present the ratios with respect to NC and q-NC, respectively, for SHI versus COMB.

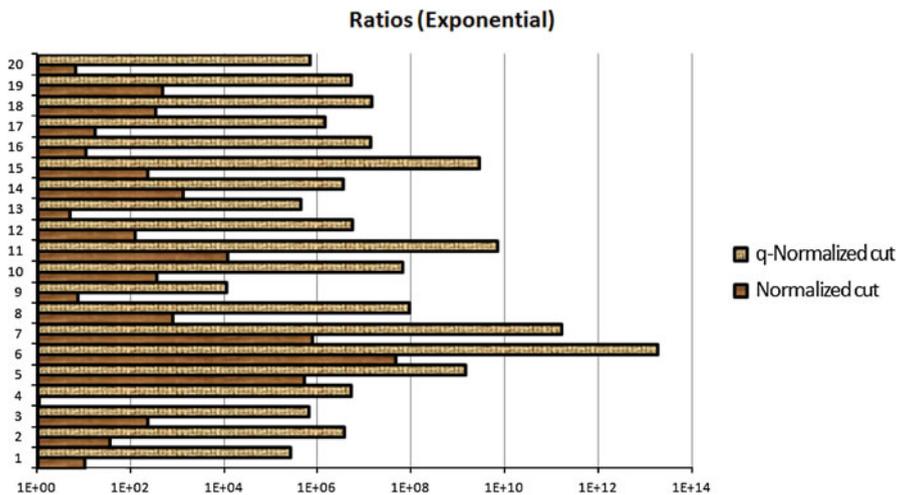
As seen in Table 4, for every case COMB yields smaller NC objective values than SHI, with the mean improvement factor exceeding  $2.3 \times 10^6$ . For q-NC the results in Table 5 show that COMB not only yields smaller objective values than SHI, but these improvements are dramatically larger than those for NC, with a mean improvement factor exceeding  $9.2 \times 10^{11}$ . We conclude that the relative performance of SHI is

**Table 4** The ratios of the NC objective values of SHI-NC-EXP to COMB-NC-EXP

Image 1	Image 2	Image 3	Image 4	Image 5
10.034375	34.945958	228.88261	1.0776067	522467.35
Image 6	Image 7	Image 8	Image 9	Image 10
45242414	757898.1	800.08425	7.1908952	357.12512
Image 11	Image 12	Image 13	Image 14	Image 15
11514.768	125.05640	4.8974465	1340.9285	233.03002
Image 16	Image 17	Image 18	Image 19	Image 20
11.050608	16.897142	345.39787	471.05938	6.5424435

**Table 5** The ratios of the q-NC objective values of SHI-qNC-EXP to COMB-qNC-EXP

Image 1	Image 2	Image 3	Image 4	Image 5
257689.45	3599904.9	654763.23	5261971.8	1418996100
Image 6	Image 7	Image 8	Image 9	Image 10
$1.8295880 \times 10^{13}$	$1.6418361 \times 10^{11}$	92852128	10836.986	63071852
Image 11	Image 12	Image 13	Image 14	Image 15
6835417700	5388176	431894.04	3397368.9	2755701700
Image 16	Image 17	Image 18	Image 19	Image 20
13524963	1440666.1	14317766	5203545.6	681058.90



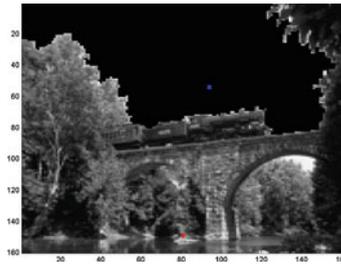
**Fig. 2** Bar chart for the ratios in Tables 4 and 5. The darker bars represent ratios for NC (Table 4) and the lighter bars represent ratios for q-NC (Table 5)

**Table 6** The mean and median values of the improvements of COMB-NC-EXP on SHI-NC-EXP and COMB-qNC-EXP on SHI-qNC-EXP

	Mean of improvements	Median of improvements
NC	2326914.4	230.95632
q-NC	$9.2356417 \times 10^{11}$	5325073.9

noncompetitive for NC and worse still for q-NC. Figure 2 is a bar chart for the ratios in Tables 4 and 5. (Note that the bar chart ratios are given in log scale, and same for the rest bar charts.) Table 6 displays the mean and median values of the ratios in Tables 4 and 5. They demonstrate the extent of the improvement of COMB over SHI.

**Table 7** Illustrating why COMB favors unbalanced cut with intervening contour similarity weights



(a) Image8-Cut  $(S8, \bar{S}8)$

	$C(S8, \bar{S}8)$	Number of edges in $(S8, \bar{S}8)$	Average edge weight of $(S8, \bar{S}8)$	$C(\{v\}, V \setminus \{v\})$
EXP	0.00095473570	520	0.0000018360302	3.9985
IC	886.11728	1315	0.67385345	8

(b) The capacities of the cut  $(S8, \bar{S}8)$  and  $(\{v\}, V \setminus \{v\})$ , and the average edge weight of edges in cut  $(S8, \bar{S}8)$  with two similarity weights.

*Comparing instances with intervening contour similarity weights: comparing SHI-NC-IC with COMB-NC-IC and SHI-qNC-IC with COMB-qNC-IC*

For the intervening contour similarity weights, we observe that the graph has nodes of (roughly) equal degrees. The edge weights are almost the same value close to 1. Hence, the cut separating a small set of nodes consists of fewer edges, and thus smaller capacity, than those cuts that separate sets of roughly equal sizes. Consequently, COMB favors unbalanced cuts with a small number of nodes on one side of the bipartition. This phenomenon is illustrated for Image 8 in Table 7.

The bipartition  $(S8, \bar{S}8)$  obtained by COMB-NC-EXP in Table 7a has the background pixels all black and the foreground pixels unmodified. In this particular bipartition the background is the sky. Thus, the similarity weights of edges in the cut  $(S8, \bar{S}8)$  should be small. We compute the capacity of cut  $(S8, \bar{S}8)$ ,  $C(S8, \bar{S}8)$ , with respect to exponential weights and intervening contour weights.

For the exponential and intervening contour similarity weights, the maximum similarity value is 1. Note, however, that the average intervening contour edge weight in cut  $(S8, \bar{S}8)$  is 0.67385345, which is quite close to 1. This is not the case for exponential similarity weights where the average exponential edge weight in cut  $(S8, \bar{S}8)$  is 0.0000018360302. This demonstrates that intervening contour similarity weights are almost uniform and close to 1 throughout the graph.

We now select a single pixel  $v$  (highlighted with the square) in the background (sky), which implies that it is highly similar to its neighbors, and considers the cut  $(\{v\}, V \setminus \{v\})$ . For exponential similarity weights the capacity of this cut,  $C(\{v\}, V \setminus \{v\})$ , is 3.9985 and therefore substantially higher than the capacity of the cut  $(S8, \bar{S}8)$ . For intervening contour similarity weights, however, the capacity of the cut  $(\{v\}, V \setminus \{v\})$  is 8, which is far smaller than the capacity of the cut  $(S8, \bar{S}8)$ .

Therefore, intervening contour similarity weights do not work well and produce unbalanced cuts with algorithms that consider the cut capacity such as COMB.

**Table 8** The ratios of the NC objective values of SHI-NC-IC to COMB-NC-IC

Image 1	Image 2	Image 3	Image 4	Image 5
0.0059191506	$3.013586 \times 10^{-5}$	$2.0904086 \times 10^{16}$	1	7.42769
Image 6	Image 7	Image 8	Image 9	Image 10
0.0026249997	0.00019634185	2.8444556	0.0047000578	0.014255762
Image 11	Image 12	Image 13	Image 14	Image 15
0.0053146160	0.0086205694	0.0071974529	$9.7893761 \times 10^{-6}$	1.5935882
Image 16	Image 17	Image 18	Image 19	Image 20
0.0015875799	0.0045286429	2.0502743	0.00040627891	0.033866313

**Table 9** The ratios of the q-NC objective values of SHI-qNC-IC to COMB-qNC-IC

Image 1	Image 2	Image 3	Image 4	Image 5
426.94051	0.49236168	51.416438	1291202.2	10270.311
Image 6	Image 7	Image 8	Image 9	Image 10
44.623615	0.64894629	110933.38	1.5971419	0.68595629
Image 11	Image 12	Image 13	Image 14	Image 15
0.61131118	1.1039278	1.1031171	1.3381326	3497.6627
Image 16	Image 17	Image 18	Image 19	Image 20
11.839168	0.50931088	34866.574	165.89208	307.61301

Tables 8 and 9 show the ratios over the 20 benchmark images using intervening contour similarity weights. Table 8 demonstrates the ratios with respect to NC objective values and Table 9 for q-NC objective values, for SHI versus COMB.

For the NC results shown in Table 8, there are 5 images where COMB gives better approximations while for the rest 15 images SHI performs better. In most of the cases among the 15 images, COMB just favors an unbalanced cut.

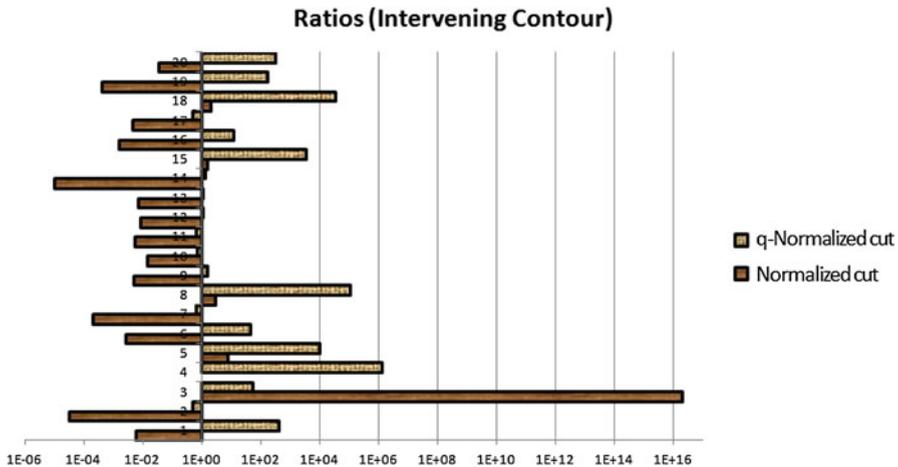
For q-NC, as can be seen from the ratios displayed in Table 9, COMB performs much better than SHI. There are 15 images where COMB gives better approximations to q-NC than SHI. Furthermore, for q-NC, the improvements of COMB on SHI are very significant. Table 10 shows the average improvements of COMB on SHI with respect to NC and q-NC with intervening contour similarity weights. They are the mean and median values of the ratios that are greater than 1 in Table 8, for NC, and Table 9, for q-NC, respectively. We also display the average improvements of SHI on COMB for NC and q-NC with intervening contour similarity weights in Table 11. For all the ratios in Tables 8 and 9 that are smaller to 1, the corresponding reciprocals characterize the improvements of SHI. We take the mean and median values of these reciprocals from Table 8, for NC, and Table 9, for q-NC, respectively, to produce Table 11. We display the ratios of Tables 8 and 9 in bar chart Fig. 3. The bars extending to the right (the ratios are greater than 1) indicate the improvements of COMB on SHI while the bars extending to the left (the ratios are smaller than 1) indicate the improvements of SHI on COMB.

**Table 10** The mean and median values of the improvements of COMB-NC-IC on SHI-NC-IC and COMB-qNC-IC on SHI-qNC-IC

	Mean of improvements	Median of improvements
NC	$4.1808173 \times 10^{15}$	2.8444556
q-NC	96785.570	165.89208

**Table 11** The mean and median values of the improvements of SHI-NC-IC on COMB-NC-IC and SHI-qNC-IC on COMB-qNC-IC

	Mean of improvements	Median of improvements
NC	9669.7517	212.76334
q-NC	1.7258142	1.6358281



**Fig. 3** Bar chart for the ratios in Tables 8 and 9. The darker bars represent ratios for NC (Table 8) and the lighter bars represent ratios for q-NC (Table 9)

Comparing approximation quality of SWEEP and COMB

In general, SWEEP should perform better than just taking the eigenvector with some predetermined procedure for bipartition. Let  $NC_{SWEEP}$  be the normalized cut objective value generated by SWEEP. By the analysis in “A bound on the relation between the spectral method solution and  $NC_G$ ”, it can be shown that  $NC_{SWEEP} \leq 4\sqrt{NC_G}$ . Therefore, one may expect that SWEEP will improve on SHI in approximating the optimal objective value of NC. We illustrate the potential improvement of SWEEP over SHI for Image 6, where the gap between the approximation of COMB-NC-EXP and the approximation of SHI-NC-EXP is largest among the 20 images, as shown in Table 4. From the original data, the approximate objective value of NC achieved by COMB is  $NC_{COMB} = 3.8129621 \times 10^{-11}$ . Therefore, the optimal NC objective of Image 6,  $NC_G$ , is less than or equal to the value of  $NC_{COMB}$ . If we use  $NC_{COMB}$  as

**Table 12** The ratios of the NC objective values of SWEEP-NC-EXP to COMB-NC-EXP

Image 1	Image 2	Image 3	Image 4	Image 5
0.0066413908	0.030815693	1.7318891	0.25438092	1.021992
Image 6	Image 7	Image 8	Image 9	Image 10
32.769017	151053.5	1.0124601	0.65635751	16.970223
Image 11	Image 12	Image 13	Image 14	Image 15
3229.7642	1.7051154	0.59025832	0.073415316	13.387584
Image 16	Image 17	Image 18	Image 19	Image 20
0.025636729	1.2725133	1.6439209	0.56137852	0.22995850

an estimation to  $NC_G$ , then we obtain an upper bound for  $NC_{\text{SWEEP}}$ :

$$\begin{aligned} NC_{\text{SWEEP}} &\leq 4\sqrt{NC_G} \leq 4\sqrt{NC_{\text{COMB}}} = 4\sqrt{3.8129621 \times 10^{-11}} \\ &= 2.4699675 \times 10^{-5}. \end{aligned} \quad (8)$$

However, our original data show that the objective value of NC achieved by SHI for Image 6 is  $1.7250761 \times 10^{-3}$ . Since  $NC_{\text{SWEEP}}$  can only be smaller than the upper bound  $2.4699675 \times 10^{-5}$ , the objective value of NC achieved by SWEEP improves by at least a factor of 70 on the objective value generated by SHI. Indeed, the experimental results match the theoretical prediction that SWEEP does better than SHI for NC. But still, COMB is better than SWEEP with exponential similarity weights.

In addition to the improvement of SWEEP over SHI or fixed threshold bipartition of the Fiedler eigenvector, SWEEP can improve on COMB for intervening contour similarity weights. As discussed in ‘‘Comparing instances with intervening contour similarity weights: comparing SHI-NC-IC with COMB-NC-IC and SHI-qNC-IC with COMB-qNC-IC’’, COMB tends to provide unbalanced bipartitions for intervening contour similarity weight matrices. For SWEEP this is not an issue, because each threshold bipartition is considered, and the best threshold will obviously correspond to a balanced bipartition. Therefore, we expect SWEEP to do better than COMB for intervening contour similarity weights.

In the following, we display the comparisons of approximation quality of SWEEP-NC-EXP with COMB-NC-EXP and SWEEP-qNC-EXP with COMB-qNC-EXP.

Tables 12 and 13 show the ratios over the 20 benchmark images using exponential similarity weights for SWEEP versus COMB. Tables 12 and 13 present the ratios with respect to NC and q-NC, respectively.

For the NC results shown in Table 12, there are 11 out of the 20 benchmark images where COMB gives a better approximation than SWEEP and the improvements of COMB over SWEEP are smaller than those over SHI. For the q-NC results displayed in Table 13, COMB dominates SWEEP and gives better approximations in every case. The results establish that while SWEEP delivers better results than SHI, COMB is still dominant, and gives better results in most cases. We display the mean and median values of the improvements of each method to the other in Tables 14 and 15, respectively. Table 14 is for the average improvement of COMB on SWEEP, while

**Table 13** The ratios of the q-NC objective values of SWEEP-qNC-EXP to COMB-qNC-EXP

Image 1	Image 2	Image 3	Image 4	Image 5
44415.346	295255	222060.29	4827.1580	26524576
Image 6	Image 7	Image 8	Image 9	Image 10
20416303000	16686194000	248921.64	5306.5587	23558163
Image 11	Image 12	Image 13	Image 14	Image 15
2097520100	1344474.1	80934.898	921433.37	25749626
Image 16	Image 17	Image 18	Image 19	Image 20
54403.132	216141.34	1191186.4	2289243	70300.882

**Table 14** The mean and median values of the improvements of COMB-NC-EXP on SWEEP-NC-EXP and COMB-qNC-EXP on SWEEP-qNC-EXP

	Mean of improvements	Median of improvements
NC	14032.252	1.7318891
q-NC	1964141900	608344.19

**Table 15** The mean and median values of the improvements of SWEEP-NC-EXP on COMB-NC-EXP

	Mean of improvements	Median of improvements
NC	27.658703	4.3486107

Notice that for q-NC, there is no improvement of SWEEP-qNC-EXP on COMB-qNC-EXP

Table 15 is for the average improvement of SWEEP on COMB. The mean and median values are obtained from the ratios in Tables 12 and 13 for NC and q-NC, respectively, using the same method introduced in “Comparing instances with intervening contour similarity weights: comparing SHI-NC-IC with COMB-NC-IC and SHI-qNC-IC with COMB-qNC-IC”. The ratios in Tables 12 and 13 are displayed as a bar chart in Fig. 4.

### Visual segmentation quality evaluation

In this section, we first evaluate the visual segmentation quality among the three methods: COMB, SHI and SWEEP. Then we compare the criteria NC' and q-NC' to NC and q-NC, respectively, to see which is a better criterion to give good visual segmentation results. Since visual quality is subjective, we provide a subjective assessment, which may not agree with the readers' judgement.

In some of the comparisons made in this section, we select for COMB the cut which gives the visually best segmentation among the four series of nested cuts corresponding to the four pairs of seeds selected according to the automatic seed selection criterion, as the output of COMB. This visually best cut is often not the numerically best cut that gives the smallest value for NC or q-NC objectives. When the visually best cut is chosen as the output of COMB, we use

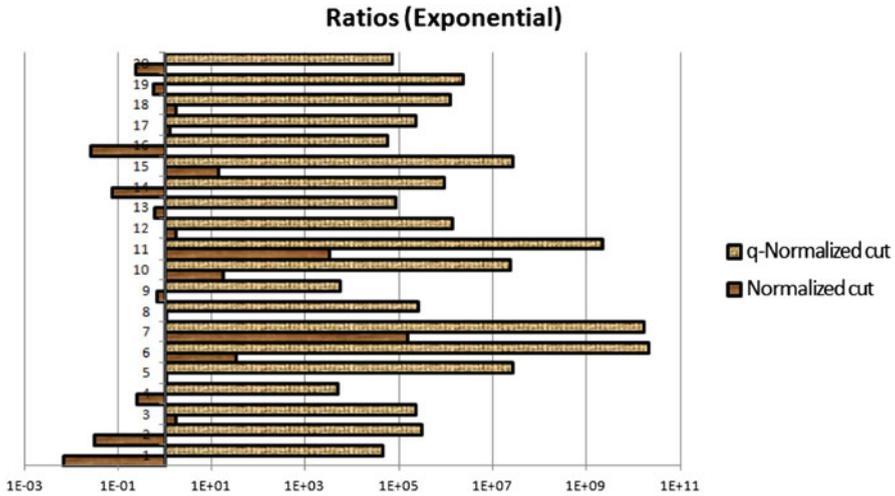


Fig. 4 Bar chart for the ratios in Tables 12 and 13. The darker bars represent ratios for NC (Table 12) and the lighter bars represent ratios for q-NC (Table 13)

the experimental set notation  $COMB(NC')$ -Similarity or  $COMB(qNC')$ -Similarity. Here, the  $(NC')$  or  $(qNC')$  is used to denote which optimization objective that COMB actually solves, and the -Similarity choice can be either exponential or intervening contour similarity weights. Notice that  $COMB(NC')$ -Similarity represents a different experimental set from  $COMB-NC$ -Similarity introduced in “Algorithm, optimization criterion, and similarity classifications and nomenclatures”, since the former experimental set uses the visually best cut while the latter experimental set uses the numerically best cut. So are  $COMB(qNC')$ -Similarity and  $COMB-qNC$ -Similarity.

For SHI and SWEEP, since each of them outputs a unique cut as the solution, there is no distinction between the numerically and visually best cuts. We still use the experimental set notations defined in “Algorithm, optimization criterion, and similarity classifications and nomenclatures” for experimental sets of SHI and SWEEP.

SHI uses a discretization method to generate a bipartition from the Fiedler eigenvector (Yu and Shi 2003) which is considered to give good visual segmentations. Hence, when comparing SHI with COMB, we use the visually best cut as the output of COMB. We conduct the following four comparisons between SHI and COMB:

- SHI-NC-EXP and  $COMB(NC')$ -EXP
- SHI-qNC-EXP and  $COMB(qNC')$ -EXP
- SHI-NC-IC and  $COMB(NC')$ -IC
- SHI-qNC-IC and  $COMB(qNC')$ -IC.

**Table 16** Visual comparison results

Experimental set 1	Experimental set 2	$1 \succ_v 2$	$2 \succ_v 1$	$1 \simeq_v 2$
SHI-NC-EXP	COMB(NC')-EXP	2	14	4
SHI-qNC-EXP	COMB(qNC')-EXP	0	20	0
SHI-NC-IC	COMB(NC')-IC	10	5	5
SHI-qNC-IC	COMB(qNC')-IC	0	5	15
SWEEP-NC-EXP	COMB-NC-EXP	2	8	10
SWEEP-qNC-EXP	COMB-qNC-EXP	3	8	9
COMB(NC')-EXP	COMB(qNC')-EXP	0	7	13
COMB-NC-EXP	COMB(NC')-EXP	0	14	6
COMB-qNC-EXP	COMB(qNC')-EXP	0	13	7

When comparing SWEEP with COMB, we use the numerically best cut as the output of COMB. This is because SWEEP outputs the cut that gives the smallest objective value of NC or q-NC among all potential threshold values. Hence, we conduct the following two comparisons between SWEEP and COMB:

SWEEP-NC-EXP and COMB-NC-EXP  
 SWEEP-qNC-EXP and COMB-qNC-EXP.

We assess the visual quality of segmentations generated by COMB to compare the performance of different optimization criteria in producing visually good segmentation results. We compare the visual segmentation quality of COMB(NC')-EXP with COMB(qNC')-EXP to determine which criterion, NC' or q-NC', works better visually. We then compare NC with NC' and q-NC with q-NC' by comparing the visual segmentation quality of the following two pairs of experimental sets:

COMB-NC-EXP and COMB(NC')-EXP  
 COMB-qNC-EXP and COMB(qNC')-EXP.

For each of the above comparisons of two experimental sets, namely *experimental set 1* and *experimental set 2*, we classify each of the 20 benchmark images into the following three categories:

1. Experimental set 1 gives a better visual segmentation result than experimental set 2. This is denoted as  $1 \succ_v 2$ , where the subscript “v” stands for “visual” and same for the rest.
2. Experimental set 2 gives a better visual segmentation result than experimental set 1. This is denoted as  $2 \succ_v 1$ .
3. Both experimental set 1 and experimental set 2 give segmentations of similar visual quality. It includes both cases where the segmentations generated by the two experimental sets are either both good or both bad. This is denoted as  $1 \simeq_v 2$ .

For each of the above visual comparisons, we count how many benchmark images belong to each category. The results are summarized in Table 16.



**Fig. 5** The visual segmentations of SHI-NC-IC (-SHI) and COMB(NC')-EXP (-COMB), and their respective original image (-Ori)

Based on the data in the first six rows of Table 16, we find that with exponential similarity weights, in general, the visual quality of segmentations generated by COMB is superior to both SHI and SWEEP. If the  $q$ -NC (or  $q$ -NC') optimization objective is applied, the visual superiority of COMB over SHI and SWEEP is dominant. Based on the data in the seventh row of Table 16, we find that  $q$ -NC' works better visually than NC'. According to the data in the last two rows of Table 16, we find that the criteria NC or  $q$ -NC are *not* good segmentation criteria. Since the visually best segmentations are obtained through solving NC' or  $q$ -NC', they should be preferred segmentation criteria, for good visual segmentation quality and their tractability.

We find from Table 16 that, in general, SHI-NC-IC delivers best visual segmentations among all the experimental sets using method SHI or SWEEP. That is, SHI works better with intervening contour similarity weights. We also find that COMB(NC')-EXP provides better visual results than COMB(NC')-IC, meaning that COMB works better with exponential similarity weights.

We provide here the images and their segmentations for the two leading methods, SHI-NC-IC and COMB(NC')-EXP. The segmentation is displayed by setting all the pixels in the background part to be black. For each of the 20 benchmark images shown, we give the segmentations generated by SHI-NC-IC and COMB(NC')-EXP in Fig. 5. Notice that for Images 1, 2, 10, 12, 19 and 20, the segmentations generated by SHI-NC-IC are almost entirely black. This is because the segmentations of these images by SHI-NC-IC have almost all pixels in the background parts.

Our judgment is that for Image 1, Image 2, Image 5, Image 10, Image 12, Image 13, Image 14, Image 15, Image 16, Image 17, Image 19 and Image 20, COMB(NC')-EXP gives visually better segmentations than SHI-NC-IC; for Image 3 and Image 6, SHI-NC-IC is visually better than COMB(NC')-EXP; for Image 4, Image 7, Image 8 and Image 18, both SHI-NC-IC and COMB(NC')-EXP generate visually good segmentations of similar quality; for the rest two images, Image 9 and Image 11, neither COMB(NC')-EXP nor SHI-NC-IC gives visually good segmentations.

## Conclusions

We report here on detailed experiments conducted on algorithms for the normalized cut and its generalization as quantity normalized cut applied to image segmentation problems. We find that, in general, the combinatorial flow algorithm of Hochbaum (2010, 2012) outperforms the spectral method both numerically and visually. In most cases, the combinatorial algorithm yields tighter objective function values of the two criteria we test. Furthermore, we find that the combinatorial algorithm almost always produces a visual segmentation which is at least as good as that of the spectral method, and often better.

Another important finding in our experiments is that, in contrary to prevalent belief, the normalized cut criterion is not a good model for image segmentation, since it does *not* provide good quality solutions, in terms of visual quality. Moreover, the normalized cut problem is NP-hard. We conclude that instead of modeling the image segmentation

problem as the normalized cut problem, it is more effective to model and solve the problem as the polynomial time solvable normalized cut' problem.

For future research, we plan on investigating other methods of solving image segmentation and other clustering problems, such as the k-means clustering method discussed in [Dhillon et al. \(2004, 2007\)](#).

**Acknowledgments** The authors wish to express their thanks to Arnaud CARUSO for his contribution to this project, and in particular, to the development of the automatic seed selection method.

## Appendix

### Benchmark images

Figure 6 contains all the 20 benchmark images we use in the experiment. We sequentially name them from Image 1 to Image 20.



**Fig. 6** The 20 benchmark images

## References

- Alon N, Milman VD (1985)  $\lambda_1$ , isoperimetric inequalities for graphs, and superconcentrators. *J Combin Theory Ser B* 38(1):73–88
- Alon N (1986) Eigenvalues and expanders. *Combinatorica* 6(2):83–96
- Chandran BG, Hochbaum DS (2012) Pseudoflow parametric maximum flow solver version 1.0. <http://riot.ieor.berkeley.edu/Applications/Pseudoflow/parametric.html>. Retrieved August 2012
- Chandran BG, Hochbaum DS (2009) A computational study of the pseudoflow and push-relabel algorithms for the maximum flow problem. *Oper Res* 57(2):358–376
- Cheeger J (1970) A lower bound for the smallest eigenvalue of the Laplacian. In: Gunning RC (ed) *Problems in analysis*. Princeton University Press, Princeton, pp 195–199
- Chung FRK (2007) Four proofs for the Cheeger inequality and graph partition algorithms. In: *Proceedings of the international congress of Chinese mathematicians*, vol 2
- Chung FRK (1997) *Spectral graph theory*. American Mathematical Society, Providence
- Coleman GB, Andrews HC (1979) Image segmentation by clustering. *Proc IEEE* 67(5):773–785
- Cour T, Yu S, Shi J (2011) MATLAB normalized cut image segmentation code. <http://www.cis.upenn.edu/~jshi/software/>. Retrieved July 2011
- Dhawan PA (2003) *Medical imaging analysis*. Wiley, Hoboken
- Dhillon IS, Guan YQ, Kulis B (2004) Kernel k-means: spectral clustering and normalized cuts. In: *Proceedings of international conference on knowledge discovery and data mining*
- Dhillon IS, Guan YQ, Kulis B (2007) Weighted graph cuts without eigenvectors: a multilevel approach. *IEEE Trans Pattern Anal Mach Intell* 29(11):1944–1957
- Donath WE, Hoffman AJ (1973) Lower bounds for the partitioning of graphs. *IBM J Res Dev* 17:420–425
- Fiedler M (1975) A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory. *Czech Math J* 25(100):619–633
- Gallo G, Grigoriadis MD, Tarjan RE (1989) A fast parametric maximum flow algorithm and applications. *SIAM J Comput* 18(1):30–55
- Hochbaum DS (2010) Polynomial time algorithms for ratio regions and a variant of normalized cut. *IEEE Trans Pattern Anal Mach Intell* 32(5):889–898
- Hochbaum DS (2012) A polynomial time algorithm for Rayleigh ratio on discrete variables: replacing spectral techniques for expander ratio, normalized cut and Cheeger constant. *Oper Res* (to appear, 2012) Early version in, Hochbaum DS (2010) Replacing spectral techniques for expander ratio, normalized cut and conductance by combinatorial flow algorithms. arXiv:1010.4535v1 [math.OC]
- Hochbaum DS (2008) The pseudoflow algorithm: a new algorithm for the maximum-flow problem. *Oper Res* 56(4):992–1009
- Hosseini MS, Araabi BN, Soltanian-Zadeh H (2010) Pigment melanin: pattern for Iris recognition. *IEEE Trans Instrum Meas* 59(4):792–804
- Leung T, Malik J (1998) Contour continuity in region based image segmentation. In: Burkhardt H, Neumann B (eds) *Proceedings of the fifth European conference on computer vision*, Freiburg, vol 1, pp 544–559
- Malik J, Belongie S, Leung T, Shi J (2001) Contour and texture analysis for image segmentation. *Int J Comput Vis* 43(1):7–27
- Pappas TN (1992) An adaptive clustering algorithm for image segmentation. *IEEE Trans Signal Process* 40(4):901–914
- Pham DL, Xu CY, Prince JL (2000) Current methods in medical image segmentation. *Annu Rev Biomed Eng* 2:315–337
- Roobottom CA, Mitchell G, Morgan-Hughes G (2010) Radiation-reduction strategies in cardiac computed tomographic angiography. *Clin Radiol* 65(11):859–867
- Shapiro LG, Stockman GC (2001) *Computer vision*. Prentice-Hall, New Jersey
- Sharon E, Galun M, Sharon D, Basri R, Brandt A (2006) Hierarchy and adaptivity in segmenting visual scenes. *Nature* 442:810–813
- Shi J, Malik J (2000) Normalized cuts and image segmentation. *IEEE Trans Pattern Anal Mach Intell* 22(8):888–905
- Tolliver DA, Miller GL (2006) Graph partitioning by spectral rounding: applications in image segmentation and clustering. *IEEE conference on computer vision and pattern recognition*, pp 1053–1060
- Wu Z, Leahy R (1993) 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

- 
- Xing EP, Jordan MI (2003) On semidefinite relaxations for normalized k-cut and connections to spectral clustering. Tech. Report No. UCB/CSD-3-1265, June
- Yu SX, Shi J (2003) Multiclass spectral clustering. In: Proceedings of international conference on computer vision, pp 313–319