Pattern Recognition 43 (2010) 4069–4076

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Enabling scalable spectral clustering for image segmentation

Frederick Tung*, Alexander Wong, David A. Clausi

Vision and Image Processing Lab, Systems Design Engineering, University of Waterloo, 200 University Ave. West, Waterloo, Ontario, Canada N2L 3G1

ARTICLE INFO

Article history:
Received 1 February 2010
Received in revised form
25 May 2010
Accepted 20 June 2010

Keywords:
Spectral clustering
Image segmentation
Stochastic ensemble consensus

ABSTRACT

Spectral clustering has become an increasingly adopted tool and an active area of research in the machine learning community over the last decade. A common challenge with image segmentation methods based on spectral clustering is scalability, since the computation can become intractable for large images. Down-sizing the image, however, will cause a loss of finer details and can lead to less accurate segmentation results. A combination of blockwise processing and stochastic ensemble consensus are used to address this challenge. Experimental results indicate that this approach can preserve details with higher accuracy than comparable spectral clustering image segmentation methods and without significant computational demands.

1. Introduction

In the last decade, spectral clustering has become a very active area of research in the machine learning community, with many extensions and applications of the algorithm being developed. One of the more popular applications of spectral clustering algorithms is image segmentation. Image segmentation plays a fundamental role in computer vision as a requisite step in such tasks as object detection, classification, and tracking [1]. Application areas of image segmentation include content-based image retrieval, automated industrial inspection, medical image processing, and remote sensing [2,3].

One of the challenges of image segmentation algorithms based on spectral clustering is scalability. Spectral clustering involves the eigendecomposition of a pairwise similarity matrix, which is intractable for sufficiently large images. Down-sizing the image, however, will cause a loss of finer details and can lead to inaccurate segmentation results. The proposed method solves this problem by successfully applying spectral clustering to large images using a combination of blockwise processing and stochastic ensemble consensus.

Section 2 briefly outlines the basic spectral clustering algorithm. Section 3 discusses related work in spectral clustering image segmentation. Section 4 describes the proposed method for scalable and detail-preserving spectral clustering image segmentation. Experimental results are presented in Section 5, and conclusions and directions for future work are presented in Section 6.

2. Spectral clustering

Like other clustering algorithms, spectral clustering attempts to partition data points into groups such that the members of a group are similar to each other and dissimilar to data points outside of the group. Spectral clustering has a simple formulation and can be solved by standard linear algebra techniques, however, it typically produces better results than traditional clustering algorithms such as k-means and mixture models [4–7].

Spectral clustering requires the construction of a weighted graph that encodes the similarity (or affinity) between data points. Nodes in the graph correspond to data points; the weight of the edge between two nodes is a function of the similarity between the corresponding two data points. Given data points $x_1,\dots,x_n$, this weighted graph can be represented by a weighted adjacency matrix W, where w_{ij} is a measure of the similarity between x_i and x_j. In the case of image segmentation, n is the total number of pixels in the image.

The degree d_i of node i is the sum of all edge weights incident on x_i:

$$d_i = \sum_{j=1}^{n} w_{ij} \tag{1}$$

The degree matrix D is defined as the $n \times n$ diagonal matrix with $d_1,\dots,d_n$ on the diagonal. Finally, the graph Laplacian matrix can be defined as [5,8]

$$L = D - W \tag{2}$$

Note that this is just one possible definition of the graph Laplacian and variants do exist (e.g., as found in [4]).

* Corresponding author. Tel.: +1 519 888 4567x35342; fax: +1 519 746 4791.
E-mail addresses: ftung@uwaterloo.ca (F. Tung), a28wong@uwaterloo.ca (A. Wong), dclausi@uwaterloo.ca (D.A. Clausi).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.06.015

The basic spectral clustering algorithm can now be formulated as follows [5,8]:

Given: Data points $x_1,\ldots,x_n$, number of clusters k

1. Calculate a weighted similarity graph over the data points,
2. Calculate the Laplacian matrix $L = D-W$, where D is the degree matrix and W is the weighted adjacency matrix induced by the similarity graph, as described above,
3. Calculate the first k eigenvectors (the eigenvectors corresponding to the k smallest eigenvalues) $v_1,\ldots,v_k$ of the generalized eigenproblem $Lv=\lambda Dv$, where $1 \le k \le n$,
4. Construct a matrix V with eigenvectors $v_1,\ldots,v_k$ as columns,
5. Create a new set of points $y_1,\ldots,y_n$, such that $y_i \in \mathbf{R}^k$ is the ith row of V,
6. Cluster the new set of points $y_1,\ldots,y_n$ using k-means [9] to obtain clusters $C_1,\ldots,C_k$.

Return: Set of clusters $A_1,\ldots,A_k$, in which x_j is assigned to A_i if y_j is assigned to C_i.

In the case of image segmentation, L and D are $n \times n$ matrices. Solving the generalized eigenproblem can easily become intractable, for even small images. A larger 1000×1000 image would involve $10^6 \times 10^6$ matrices. The operation is commonly made tractable by considering only each pixel's local spatial neighbourhood in the construction of the weighted similarity graph; that is, a pixel has zero similarity to pixels outside its local neighbourhood. This produces sparse W, D, and L matrices, that allows an efficient algorithm for approximating only the first k eigenvalues [8]. However, for sufficiently large images and neighbourhood sizes, this computation can still be impractical in terms of memory requirements.

The interested reader can find a more complete treatment of the theory behind spectral clustering in [10]; a good introduction to spectral clustering can also be found in [5].

3. Related work

Possibly the first image segmentation algorithm based on spectral clustering was developed by Shi and Malik [8]. Shi and Malik formulated the problem from a graph theoretic perspective and introduced the normalized cut to segment the image. Malik et al. applied the normalized cut formulation to segment grayscale images based on a combination of contour and texture cues [11].

Zelnik-Manor and Perona [7] proposed a method for automatically determining an appropriate number of clusters (segments) and adaptively selecting an appropriate neighbourhood size or scale. The number of clusters is determined by minimizing the cost, for a range of possible numbers of clusters, associated with rotating the columns of V in alignment with a canonical coordinate system, such that every row of the rotated V contains at most one non-zero entry. A local neighbourhood scale is determined for each pixel based on the distance to its kth nearest neighbour.

Chang and Yeung [12] integrated robust statistics methods in their development of a path-based spectral clustering algorithm for image segmentation. M-estimation is applied to reduce the effect of noise and outliers on the pairwise similarity matrix.

Xiang and Gong [6] proposed a method for both estimating an appropriate number of clusters and dealing with noisy data. In their method, only those eigenvectors that are likely to help separate $y_1,\ldots,y_n$ are selected to be included in V. The relevance of each eigenvector is estimated based on how much of the data can be explained by a unimodal Gaussian distribution versus a multimodal Gaussian distribution, using the expectation maximization (EM) algorithm. Instead of the final k-means step, the authors propose fitting a Gaussian mixture model in which the number of mixture components determined by the Bayesian Information Criterion [13]. Applications in both image segmentation and behaviour clustering in video are demonstrated.

Fowlkes et al. [14] explicitly addressed the scalability issue by applying the Nyström approximation to the normalized cut framework [8]. Their method is able to efficiently find an approximate segmentation using the pairwise similarities between a small random subset of the image pixels and the entire image. Segmentation results are visually comparable to those obtained by normalized cut.

4. Proposed method

The proposed method addresses a common challenge faced by image segmentation algorithms based on spectral clustering: scalability. To address this challenge, the proposed method combines a blockwise segmentation strategy with stochastic ensemble consensus. The overarching idea is to perform an over-segmentation of the image at the pixel level using spectral clustering, and then merge the segments using a combination of stochastic ensemble consensus and a second round of spectral clustering at the segment level. The purpose of using stochastic ensemble consensus is to integrate both global and local image characteristics in determining the pixel classifications. The stochastic ensemble consensus step also removes blockwise processing artifacts. The rest of this section elaborates on the specifics of the algorithm.

4.1. Blockwise segmentation

In the first step of the algorithm, the image is partitioned into non-overlapping blocks of fixed size 32×32. An over-segmentation of the image is performed by segmenting each 32×32 block in sequence using spectral clustering. In essence, each 32×32 block is treated as a separate image. For each block, a $32^2 \times 32^2 = 1024 \times 1024$ similarity matrix is calculated over the pixels. The similarity between a pixel x_i and another pixel x_j in the neighbourhood of x_i is determined by an exponentially decaying function of the squared difference in their intensity values:

$$s_{ij} = \begin{cases} \exp(-\alpha(I(x_i)-I(x_j))^2) & x_j \text{ is in the neighbourhood of } x_i \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $I(x)$ denotes the intensity of pixel x and α is a scaling constant that controls a penalty based on intensity difference. The constant α is set to 30 in all experiments. Empirically, values in the range of 25–30 tend to be effective and produce similar results. As described in Section 2, pixels outside of the local neighbourhood are assigned similarity values of zero to allow for a sparse representation of the similarity matrix. A neighbourhood size of 9×9 is used in all experiments.

Segmentation of the block is now performed using basic spectral clustering on the calculated similarity matrix. The number of clusters k is set to a fixed constant. Normally, setting k this way would be non-robust and give poor segmentation results; however, two reasons make it a feasible option here. First, as will be discussed below, the subsequent merging step with stochastic ensemble consensus effectively handles any over-segmentation performed at this stage. Therefore, setting k higher than necessary poses no serious problems. Second, to handle the case of setting k lower than necessary, a splitting step is included to ensure that all pixels assigned to the same segment are actually connected in the similarity graph. This is done by constructing an adjacency matrix based on the non-zero similarity values and finding the connected components. Under-segmentation is detected when the pixels assigned to the same segment do not form a single connected component. Intuitively, two pixels in the same segment should be similar to each other, either directly or indirectly via other neighbours. If pixels are completely disconnected in the similarity graph, then they should be assigned to different segments. Thus, the proposed algorithm is not very sensitive to the value of k at this step, and we set k to 7 in all of our experiments. Using a fixed k in this over-segmentation step has the advantages of simplicity and computational efficiency: for example, estimating k using the eigenvector selection method [6] increases the computation time of this step by several-fold, and, as mentioned above, is unnecessary.

As previously mentioned, the blockwise over-segmentation step is followed by a merging stage. The aim of the merging stage is to integrate the results of the blockwise over-segmentation at the image level to obtain the final segmentation. The merging stage consists of stochastic ensemble consensus followed by segment-level spectral clustering.

4.2. Stochastic ensemble consensus [15]

The merging stage begins with an ensemble voting process based on the stochastic ensemble consensus (SEC) method [15]. In SEC, the segmentation of a pixel x_s is determined by a weighted vote of an ensemble of pixels $x_1,\ldots,x_N$ in the image. The ensemble is formed by stochastic sampling, in which the probability of a pixel being selected to be part of the ensemble is inversely proportional to its spatial distance from x_s. The weight of a pixel in the ensemble is proportional to its similarity to x_s. The intuition is that nearby pixels with similar intensity should have greater influence in the consensus clustering.

As a very simple example, suppose in an ensemble of size $N = 10$, five pixels are classified as segment A and five are classified as segment B. If the pixels classified as segment A have the same intensity value as s, whereas the pixels classified as segment B have intensities very different from s, then the consensus classification of s would be segment A.

Using the notation in [15], the influence of a pixel x_t on the consensus decision-making process of x_s is given by

$$\Psi_s(x_t) = \exp(-\beta(I(x_s)-I(x_t))^2) \tag{4}$$

where β is a regularization parameter that depends on the image noise. The ensemble of pixels is constructed by stochastic sampling. For a given pixel x_s to be classified, a set of random pixels $x_1,\ldots,x_N$ is selected to form the ensemble based on a spatially-adaptive probability distribution function p,

$$p(x_t|x_s) = \frac{1}{|x_t-x_s|} \tag{5}$$

where $|x_t-x_s|$ is the spatial distance between x_t and x_s. Identical to the original work [15], an ensemble size of $N=20$ is used. The ensemble size is independent on the image size.

This stochastic sampling scheme attempts to integrate global and local image information to determine the underlying classification of image pixels. The spatially adaptive probability distribution function helps preserve local spatial relationships while incorporating statistics over a broad area of the image. SEC re-classifies pixels in the image over several iterations until convergence. In the proposed method, the application of stochastic ensemble consensus has two key functions: (1) it removes blockwise processing artifacts (namely sharp corners and boundaries between segments) by re-classifying and smoothing segment boundaries, and (2) it simultaneously reduces the effects of noise and outliers by merging spurious segments.

4.3. Segment-level clustering

After the SEC step, a second round of spectral clustering is performed. The clustering is now performed at the segment level instead of at the pixel level. That is, the weighted adjacency matrix W compares pairs of segments instead of pixels. To preserve spatial locality relationships, the neighbourhood of a segment consists of only spatially adjacent segments. Non-adjacent segments are assigned similarity values of zero. The similarity between two adjacent segments is measured by the similarity of their intensity distributions, where the intensity distribution of a segment refers to the probability distribution of intensity values over its constituent pixels. A segment's intensity distribution is represented using a normalized histogram of intensity. Given two normalized intensity histograms $\mathbf{p}$ and $\mathbf{q}$, their similarity is given by ρ^γ, where γ is a scaling constant and ρ is a similarity metric based on the Bhattacharyya coefficient [16]:

$$\rho(\mathbf{p},\mathbf{q}) = \sum_{u=1}^{m} \sqrt{p_u q_u} \tag{6}$$

The term m is the number of bins in the histogram and is set to 16 as per [16], and γ is set to 8 in all experiments. Hence, the similarity between two segments is determined by constructing their normalized histograms of intensity and calculating the similarity metric ρ between the two histograms. Instead of using the segments produced by SEC directly, more accurate results were obtained by first conservatively merging adjacent segments as a pre-processing step; adjacent segments are merged only if their intensity distributions are extremely similar ($\rho > 0.98$). Noisy segments containing very few pixels are also excluded since histogram representation is unreliable with insufficient samples.

The number of clusters k' in the spectral clustering is a user-specified parameter and should be an estimate that is larger than the "true" number of segments in the image. The parameter k' is more favourable than k in traditional spectral clustering because k' can be over-estimated. The use of k' is difficult to avoid even in spectral clustering algorithms that automatically estimate the number of clusters: for example, it is comparable to the user-specified K_m in spectral clustering with eigenvector selection [6].

4.4. Post-processing

Finally, since k' is an over-estimate of the number of segments in the image, a post-processing step is performed to obtain the final segmentation. A greedy iterative strategy is taken to further merge the segments. At each iteration, the two adjacent segments with the highest pairwise similarity are merged, subject to the constraint that the boundary between the segments is not a strong edge. The boundary between two segments is considered a strong edge if a sufficient proportion of the boundary pixels are identified as edge pixels by an edge detector. The current implementation uses a simple Sobel edge detector [3], which generated effective segmentations. In future work, the applicability of more sophisticated edge detection algorithms could be assessed. Similarity is measured in the same way as in the spectral clustering step described above. The iterative merging continues until the highest pairwise similarity falls below a threshold, set to $\rho = 0.4$ in all experiments.

4.5. Space complexity

As outlined in Section 2, the basic spectral clustering algorithm has memory requirements that scale with the square of the number of data points, or in the case of image segmentation, the square of the

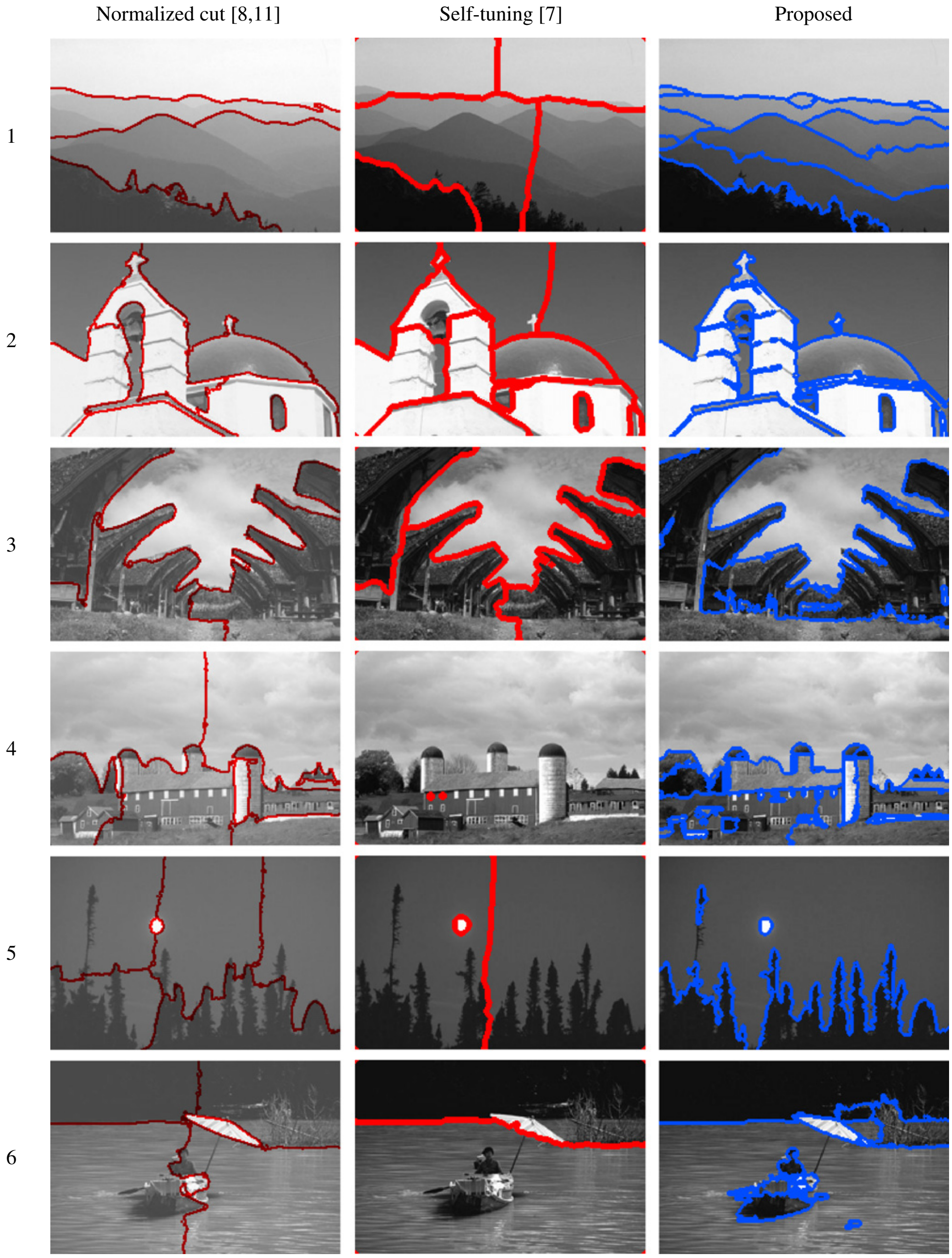

Fig. 1. Some segmentation results for natural scene images from the Berkeley segmentation database [17]. Left to right: normalized cut [8,11]; self-tuning spectral clustering [7]; proposed method. Detailed structures are better preserved, such as the distant mountains in image 1, the building outcroppings in image 2, and the tree line in image 5.

number of pixels. In contrast, since the blockwise over-segmentation is performed sequentially, the memory requirements in the first (pixel-level) round of spectral clustering do not change with the number of pixels. The memory requirements of the proposed method scale only with the square of the number of segments in the second (segment-level) round of spectral clustering.

5. Experimental results

This section presents experimental results on a set of natural scene images from the Berkeley segmentation database [17]. Segmentation results obtained by the normalized cut [8,11] and self-tuning [7] spectral clustering algorithms are shown for comparison.

Although the proposed method can be applied directly to the 320 × 480 images[1] from the Berkeley database, the MATLAB implementations of normalized cut and self-tuning spectral clustering encounter memory errors at this image resolution (running 32-bit MATLAB on a machine with 1014 MB RAM). Hence, the images are down-sized to 160 × 240 for the purposes of comparison as per Figs. 1 and 2, and full scale image results only for the proposed method are presented in Fig. 3.

Results for the Berkeley images are shown in Figs. 1 and 2. To obtain an approximate quantitative measure of classification performance, we also use the simple precision and recall

[1] The images are actually 321 × 481—in these experiments, one row and one column of pixels are trimmed to obtain uniform blocks.

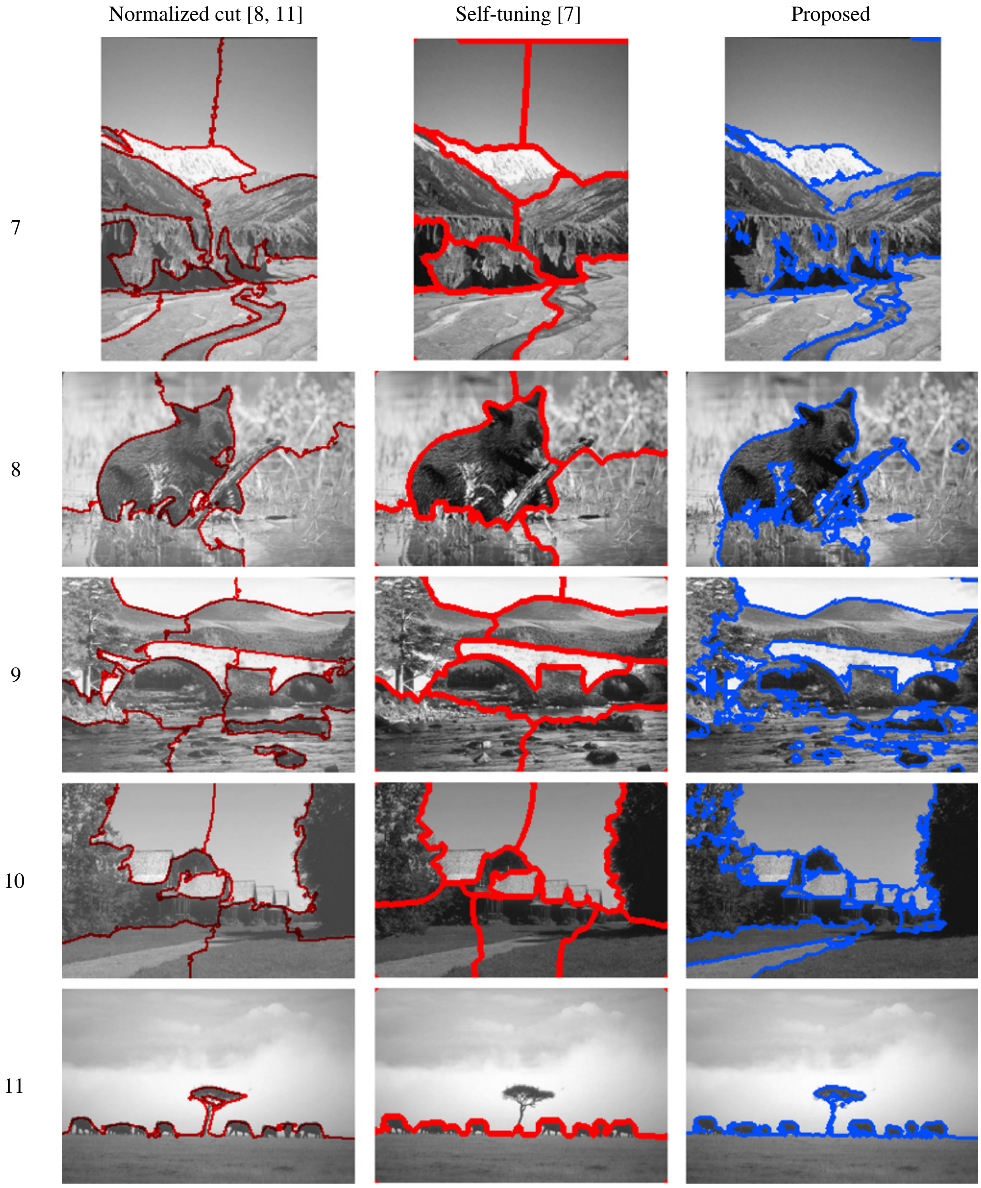

Fig. 2. Additional segmentation results for natural scene images from the Berkeley segmentation database [17]. Left to right: normalized cut [8,11]; self-tuning spectral clustering [7]; proposed method. Detailed structures are better preserved, such as the roofs and walking path in image 10, and the tree and animals in image 11.

methodology described by the authors of the Berkeley database in [18]. Precision is a measure of accuracy, and is defined as the proportion of detections that are true positives:

$$precision = \frac{TP}{TP+FP} \tag{7}$$

where TP is the number of true positives and FP is the number of false positives. Recall is a measure of completeness, and is defined as the proportion of positives that are detected:

$$recall = \frac{TP}{TP+FN} \tag{8}$$

where TP is the number of true positives and FN is the number of false negatives. Similar to [18], a distance tolerance of 2 pixels is used in the calculation of these measures. This means that in calculating recall, a ground-truth boundary pixel is considered to be successfully detected if the algorithm detects a boundary within 2 pixels, and in calculating precision, an algorithm-detected boundary pixel is considered to be a true positive if it is within 2 pixels of a ground-truth boundary.

The Berkeley segmentation database contains ground truth in the form of boundary maps that are marked by third-party human observers. Each image in Figs. 1 and 2 is marked by 4–7 humans. Following [19], detections that do not match any ground truth boundary are counted as false positives, and recall is averaged over all ground truths.

The precision and recall measures for the Berkeley images are summarized in Table 1. In Table 1, images 1 to 6 correspond to the images in Fig. 1 from top to bottom; images 7 to 11 likewise correspond to those in Fig. 2.

Overall, the proposed method achieves segmentation results that are comparable to or better than the other two methods. In particular, detailed structures are better preserved in the segmentation, as reflected in the higher recall values. Examples include the distant mountains in image 1 (overall recall of the segmentation: 0.82); the building outcroppings in image 2 (recall: 0.81); the tree

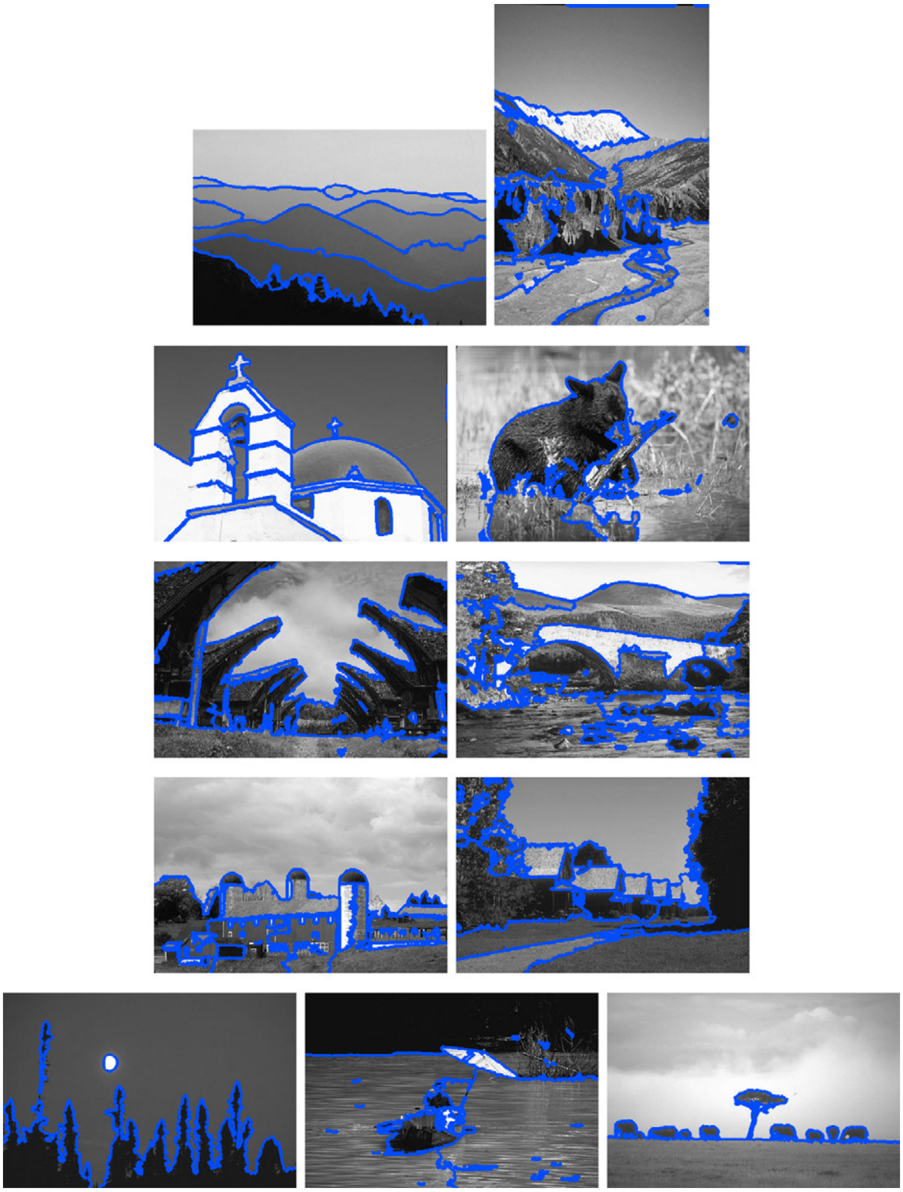

Fig. 3. Segmentation results for natural scene images from the Berkeley segmentation database [17] processed at their full resolution. Fine details such as the building outcroppings in image 2, the pillars in image 3, and the windows in image 4, are better preserved in the full resolution segmentation.

Table 1
Precision and recall measures for the segmentations in Figs. 1 and 2.

Image	Normalized cut		Self-tuning		Proposed	
	Precision	Recall	Precision	Recall	Precision	Recall
1	**0.95**	0.55	0.66	0.25	0.92	**0.82**
2	0.80	0.62	**0.89**	0.63	0.85	**0.81**
3	0.82	0.39	**0.88**	0.47	0.84	**0.63**
4	0.70	0.36	**1.00**	0.01	0.80	**0.66**
5	0.66	0.28	0.44	0.05	**1.00**	**0.80**
6	0.75	0.34	**0.90**	0.19	0.87	**0.61**
7	0.76	0.59	0.71	0.53	**0.83**	**0.67**
8	0.80	0.48	**0.82**	0.46	0.78	**0.66**
9	0.81	0.43	**0.86**	0.38	0.72	**0.58**
10	0.75	0.36	0.73	0.43	**0.76**	**0.71**
11	0.95	0.52	**1.00**	0.54	0.97	**0.86**

line in image 5 (recall: 0.80); the roofs and the walking path in image 10 (recall: 0.71); and the tree and animals in image 11 (recall: 0.86). However, the proposed method also sometimes over-segments highly textured areas. Examples include the trees in images 9 and 10 (overall precision of the segmentation: 0.72 and 0.76, respectively). This limitation is not surprising since the present implementation considers only intensity values.

Segmentation results obtained by the proposed method on the images at their full resolution are included in Fig. 3. The results are very similar to the results in Figs. 1 and 2, but fine details such as the building outcroppings in image 2, the pillars in image 3, and the windows in image 4, are better preserved in the full resolution segmentation. Fig. 4 shows magnified examples of better detail preservation at full resolution. As mentioned earlier, the normalized cut and self-tuning implementations cannot process the images at full resolution since they encounter memory errors.

6. Conclusion and future work

In conclusion, this paper has presented a novel spectral clustering image segmentation algorithm that addresses one of the main challenges faced by image segmentation methods based

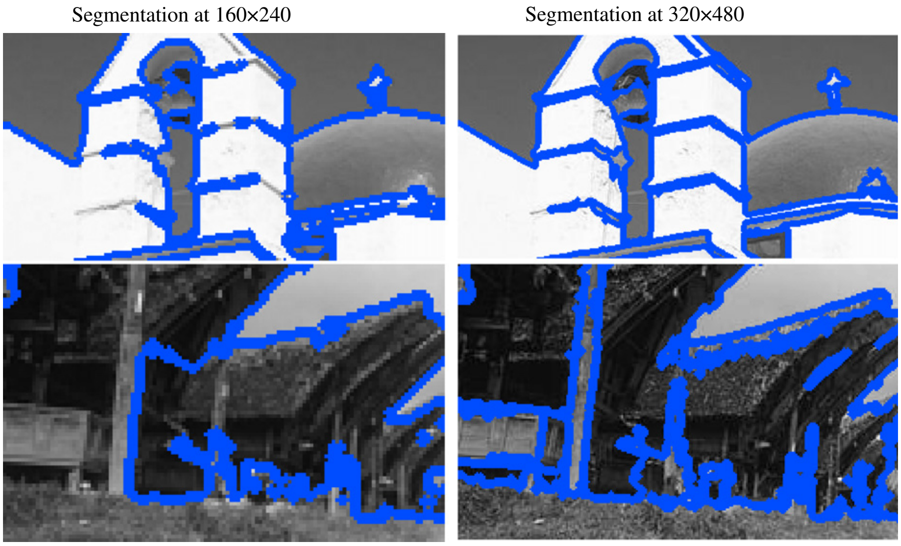

Fig. 4. Magnified examples of better detail preservation when segmenting images at full resolution. Fine details such as the building outcroppings in the top image and the pillars in the bottom image are better preserved in the full resolution segmentation.

on spectral clustering: scalability. Experimental results indicate that the proposed method is able to preserve details more accurately than comparable spectral clustering algorithms and without significant computational demands.

In the current implementation, only intensity is used to segment the images; an interesting direction of future work would be to incorporate texture information to improve the segmentation precision. We also plan to experiment with more advanced edge detection techniques for the post-processing step. Furthermore, we would like to investigate how k' might be estimated automatically based on image characteristics. Finally, we plan to explore the usefulness of the proposed method in applications where robust segmentation is needed for very large images, such as in remote sensing.

Acknowledgements

We would like to thank Lihi Zelnik-Manor and Pietro Perona for generously making a MATLAB implementation of self-tuning spectral clustering [7] available online, and also Jianbo Shi and Jitendra Malik for generously making a MATLAB implementation of normalized cut spectral clustering [8,11] available online.

This work is sponsored by the Natural Sciences and Engineering Research Council (NSERC) of Canada and by GEOIDE (Geomatics for Informed Decisions, a Network of Centres of Excellence).

References

[1] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Computing Surveys 38(4) (2006) 1–45.

[2] R. Datta, D. Joshi, J. Li, J.Z. Wang, Image retrieval: ideas, influences, and trends of the new age, ACM Computing Surveys 40(2) (2008) 1–60.

[3] R.C. Gonzalez, R.E. Woods, Digital Image Processing, third ed., Pearson Prentice-Hall, Upper Saddle River, NJ, 2008.

[4] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, Advances in Neural Information Processing Systems 2001.

[5] U. von Luxburg, A tutorial on spectral clustering, Technical Report, Max Planck Institute for Biological Cybernetics, 2006.

[6] T. Xiang, S. Gong, Spectral clustering with eigenvector selection, Pattern Recognition 41 (3) (2008) 1012–1029.

[7] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, Advances in Neural Information Processing Systems, vol. 17, 2004.

[8] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.

[9] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., John Wiley & Sons, New York, 2001.

[10] F.R.K. Chung, Spectral Graph Theory, American Mathematical Society, Providence, RI, 1997.

[11] J. Malik, S. Belongie, T. Leung, J. Shi, Contour and texture analysis for image segmentation, International Journal of Computer Vision 43 (1) (2001) 7–27.

[12] H. Chang, D.Y. Yeung, Robust path-based spectral clustering with application to image segmentation, in: Proceedings of IEEE International Conference on Computer Vision, vol. 1, Beijing, China, 2005, pp. 278–285.

[13] G. Schwarz, Estimating the dimension of a model, The Annals of Statistics 6 (2) (1978) 461–464.

[14] C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral grouping using the Nyström method, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2) (2004) 214–225.

[15] A. Wong, D.A. Clausi, P. Fieguth, SEC: stochastic ensemble consensus approach to unsupervised SAR sea-ice segmentation, in: Proceedings of Canadian Conference on Computer and Robot Vision, Kelowna, Canada, 2009, pp. 299–305.

[16] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 564–575.

[17] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, Proceedings of IEEE International Conference on Computer Vision, vol. 2, 2001, pp. 416–423.

[18] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural image boundaries using brightness and texture, Advances in Neural Information Processing Systems, vol. 15, 2002.

[19] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (5) (2004) 530–549.

Frederick Tung received the MASc degree in systems design engineering from the University of Waterloo in 2010. His research interests lie in computer vision, image processing, and pattern recognition; his current projects include image segmentation, motion estimation, and behaviour and scene analysis for intelligent surveillance.

Alexander Wong received the MASc degree in electrical and computer engineering in 2007 from the University of Waterloo. His current research interests include biomedical image processing and analysis, computer vision, and pattern recognition. He has been engaged with projects in image registration, image denoising, image super-resolution, image segmentation, biomedical tracking, biomedical image analysis, and image and video coding.

David A. Clausi is a professor in Systems Design Engineering at the University of Waterloo (Canada). As an active interdisciplinary researcher, he has an extensive publication record, publishing in diverse fields of remote sensing, computer vision, algorithm design, and biomechanics. He has received numerous scholarships, paper awards, and two Teaching Excellence Awards.