Hierarchical Mutual Nearest Neighbour Image Segmentation

SM Abdullah[1], Peter Tischer[1], Sudanthi Wijewickrema[2] and Andrew Paplinski[1]
[1]Faculty of Information Technology, Monash University
[2]Department of Surgery (Otolaryngology), The University of Melbourne
[1]{sm.abdullah, peter.tischer, andrew.paplinski}@monash.edu
[2]sudanthi.wijewickrema@unimelb.edu.au

***Abstract*—This paper presents a hierarchical image segmentation algorithm based on the principle of mutual nearest neighbours. Image segmentation remains a great challenge in the computer vision community. To solve this problem, various algorithms have been proposed in the literature. However, most of these algorithms depend heavily on thresholds or parameter settings. Furthermore, the majority of them do not recognise the hierarchical nature of the problem. In particular, there might not be a single best segmentation for an image as the level of detail that should be present in a segmentation will depend on the purpose for which that segmentation will be used. Many algorithms might provide good results for a specific application, or in detecting a certain level of detail in an image, but may fail when they are applied to different types of images at a different level of detail. The method proposed in this paper generates a hierarchy of segmentations that retain different levels of detail. Thus, depending on the application, the segmentation that provides the required level of detail can be selected. Utilisation of only one meta-parameter in the process makes it applicable to any dataset as is. It can also be easily generalised to segment different types of images including 3D and multispectral images. Evaluation on the Berkeley BSD500 dataset and comparison with existing hierarchical segmentation algorithms provide superior results.**

I. Introduction

Since the time of the Gestalt movement in psychology [1], it has been known that perceptual grouping (or image segmentation) plays an important role in human visual perception. It is a preliminary process that happens prior to object detection and recognition. This was shown in Sinha [2] through a study conducted on people who had their sight restored after being blind since birth. After regaining their sight, they were shown images and asked to explain what they saw. Interestingly, though they had prior knowledge about objects in the images, what they identified as separate 'objects' were segments based on colour differences (see Figure 1). This aspect of human vision has important implications for computer vision that aims at mimicking it's biological counterpart. For instance, an appropriate region of support is required for intermediate-level vision problems such as stereo and motion estimation. In these applications, a segmentation technique can be useful to find non-uniform regions of support [3]. For image recognition and indexing, the results of segmentation techniques play a vital role in matching, figure-ground separation, and recognition by parts [4].

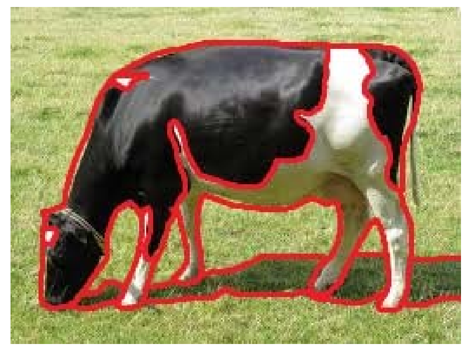


Fig. 1. A segmented cow image (recreated from [2])

The objective of any image segmentation algorithm is to partition an image I into *logical* units, commonly known as segments. However, no proper definition of logical segments can be found in the literature. Hence, image segmentation is an ill-posed problem due to a large number of possible partitioning for any single image. Mathematically, the segments of an image segmentation process can be represented as:

$$S = \{S_1 \cup S_2 \cup ... \cup S_M\} \text{ where, } \\ S_i, S_j \in S, S_i \cap S_j = \emptyset, i \neq j \quad (1)$$

Here S is the set of all segments equivalent to the input image I and there is no common pixel shared by any two segments.

Most of the existing image segmentation methods provide one level of segmentation. However, as an ill-posed problem, image segmentation has no unique solution. We believe that segmentation is a hierarchical problem and that different levels of detail should be provided at each level. Thus, depending on the application, the segmentation that provides the required level of detail can be selected. For example, consider the textured image shown in Figure 2. Depending on the requirements there are many possible 'logical' segmentations, depicting different levels of detail. See Figure 3, which presents three segmentations each providing a different level of detail.

Further, a segmentation algorithm should be generalised for different types of images and applications. However, most of the existing methods are heavily dependent on thresholds

978-1-5090-2896-2/16/$31.00 ©2016 Crown

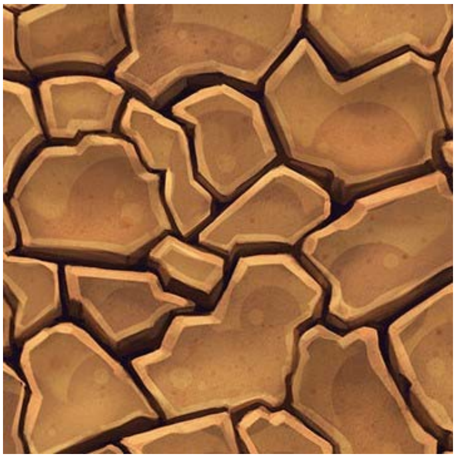
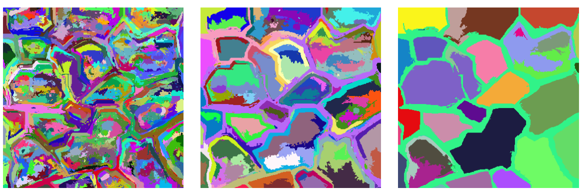


Fig. 2. A textured image



(a) (b) (c)

Fig. 3. Segmentation with different levels of detail. Segments are coloured randomly.

and parameters that control the segmentation process. These methods may not perform well when applied to different domains.

In this paper, we propose an image segmentation algorithm that exploits the concept of 'mutual nearest neighbour' and generates a hierarchy of segmentations at different levels of detail. This algorithm utilises only one parameter (used to control the level of 'inclusion' as explained later), and as such is applicable to different application domains.

The rest of the paper is organised as follows. Section II presents related literature and current state-of-the-art approaches of image segmentation. The proposed method is described in Section III followed by Section IV which discusses the experimental setup, result analysis and comparisons. Section V concludes the paper.

II. Related Work

In the last few decades, image segmentation has been well researched, and hence a vast among of literature is available on the subject. It is beyond the scope of this paper to even go through all the different categories of image segmentation. Instead, we briefly present the state-of-the-art methods that are related to the proposed method.

A. Clustering-based Approaches

Since the early days of image segmentation, clustering-based techniques have been extensively used. Although segmentation and clustering are two different problems in two different domains, there is an underlying similarity between them. Both techniques try to establish maximum intra segment or cluster similarity and maximum dissimilarity among the segments or clusters. As a consequence, many methods have been proposed in the literature, which is based on the clustering concept.

Among the many different clustering techniques, two techniques, namely, *fuzzy c-means* [5] and *k-means* [6], and their variations have gained popularity among researchers. The fuzzy c-means (FCM) algorithm is an iterative optimisation process that minimises a cost function based on the distances between pixels, cluster centres and their associated fuzzy memberships. One of the criticisms against the FCM algorithm is that it does not consider spatial relationships in the cost function. To deal with this problem, many improved FCM algorithms have been proposed that incorporate local spatial information into the original FCM cost function [7]–[9]. Ahmed et al. [7] proposed FCM_S, which modified the cost function of FCM by introducing the spatial neighbourhood term to tackle image noise. To reduce time to calculate the spatial neighbourhood term, Chen and Zhang [8] proposed another variant, which simplified the neighbourhood term of the cost function. Recently, Gong et al. [9] proposed another variation based on a trade-off weighted fuzzy factor and a kernel metric.

Another well-studied clustering approach is the K-means (KM) [6] algorithm. Unlike the FCM algorithm, it is an exclusive clustering algorithm, which guarantees that data belonging to one definite cluster would not be included in another cluster. It partitions a set of *n* objects into *k* clusters so that the resulting intra-cluster similarity is high whereas the inter-cluster similarity is low. Mashor [10] proposed a moving k-means (MKM) algorithm aiming to solve dead centres, centre redundancy and local minima problems of the traditional KM algorithm. Sulaiman and Isa [11] proposed an adaptive KM algorithm. They borrowed the idea of the fuzzy membership function from the FCM algorithm and introduced a 'belongingness' term in the traditional KM algorithm.

One of the drawbacks of clustering-based segmentation approaches is the need to know the number of clusters or segments at the start of the process.

B. Graph-based Approaches

Graph-based approaches are another popular and well-studied field in image segmentation. The basic idea of any graph-based approach is to formulate a weighted undirected graph $G = (V, E)$ out of the input image and subsequently solve the created graph for segmentation. Graph-cut and region merging are two different approaches, which have gained popularity for solving this problem.

In graph cut, a graph G can be partitioned into two disjoint sets, A, B, $A \cup B = V$, $A \cap B = \emptyset$, by simply removing edges connecting the two sets. One of the early contributions of graph cut techniques can be found in Wu and Leahy [12]. They proposed a method based on the minimum cut criterion.

However, their method favours small sets of isolated nodes in the graph. Shi and Malik [13] proposed the normalized cut (Ncut) method that eventually solved the problem of Wu and Leahy's method [12]. However, Ncut has high computational complexity.

A greedy graph-based region merging algorithm advocated by Felzenszwalb and Huttenlocher [4] attempts to partition image pixels into components such that the resulting segmentation is neither too coarse nor too fine. Given a graph in which pixels are nodes and edge weights measure the dissimilarity between nodes, a segmentation is produced by partitioning graph nodes into components. A pairwise region comparison predicate was used to decide whether or not to merge two components. A varying observation scale was used in the predicate calculation to facilitate small component merging. However, the use of observation scale was criticised by Guimarães et al. [14]. According to their analysis, the observation scale of Felzenszwalb and Huttenlocher [4] violated causality and location principles. Instead of a fixed value of the observation scale, they calculated it locally based on the internal and external distances between two components.

III. Proposed Method

The proposed segmentation method uses the concept of *k-mutual nearest neighbours*. To illustrate, consider Figure 4. Here, B is close to C in the sense that B is the nearest neighbour (k=1) of C but C is not close to A, since B is the nearest neighbour of A. A and B are mutual nearest neighbours. Instead of the nearest neighbour, we can consider k-nearest neighbours. In the first level of segmentation, this principle can be stated that 'a pixel belongs to a segment which contains its k-nearest neighbouring pixels'. Recursively, at each higher level segmentation, a segment belongs to a 'segment of segments' which contains its k-nearest segments and so on. This principle requires only one meta-parameter k, the number of nearest neighbours, which is an integer value. For an n-connected neighbourhood, k might have possible values in-between $[1, n]$. For example, for an eight-connected neighbourhood setup, as shown in Figure 5, k may take any value from one to eight.

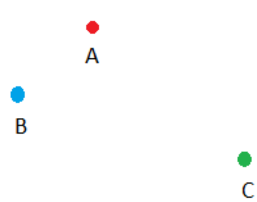


Fig. 4. Mutual nearest neighbour principle

A distance function is used to determine the k-nearest neighbours. Following consideration from Tkalcic et al. [15], we operate in the CIELAB space since for natural images the RGB colour space has a high correlation between its components and it is also perceptually non-uniform. Then, the perceptual colour distance function, CIEDE2000 [16], is used

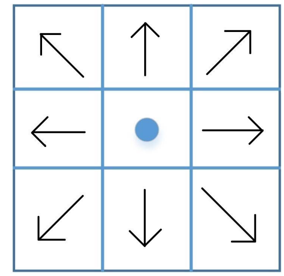


Fig. 5. Eight-connected neighbourhood of the central pixel, represented by a dot

to determine colour distances. At the pixel-level, individual L^*, a^*, b^* values of the CIELAB colour space are used to determine distance. At each higher level, the median L^*, a^*, b^* values of each segment is used. Any distance function can be used instead of the CIEDE2000 if required, as the proposed method is free from any adjustable parameter based on the distance function.

In the following two subsections, we discuss segment formation from pixels and hierarchical segment merging at higher levels of segmentation.

A. Natural Level Segmentation

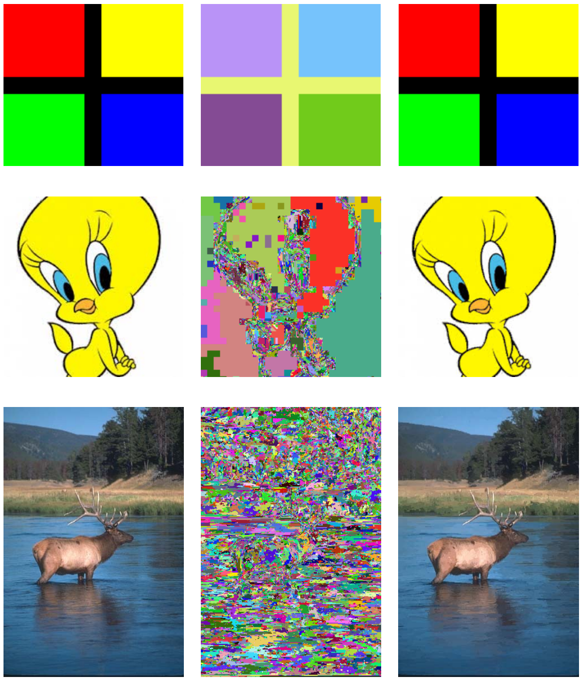
At the first level of the hierarchy, segments are formed by chaining mutually connected pixels. Initially, the set of natural segments is empty. The process starts by allocating a pixel into the first segment. This pixel then adds its k-mutual nearest neighbours to the current segment and continues until it has run out of new pixels that satisfy the mutual nearest neighbour criteria or there are no more pixels left in the image. After the formation of a segment, if there exist any unallocated pixels, the above procedure starts again to generate the next segment. Algorithm 1 presents the method of natural level segment formation and Figure 6 shows outputs resulting from this step.

Algorithm 1 Natural level segmentation

Input: Image (I), k-nearest-neighbour
Output: natural segment, $C = \{C_1, C_2, ..., C_M\}$
  for each unallocated pixel of I **do**
    $C_i[1] \leftarrow$ current_pixel
    for each pixel $\in C_i$ **do**
      KNN$_j \leftarrow$ k-nearest-neighbour$(C_i[j])$
      for each KNN$_j$ **do**
        if $C_i[j] \in$ k-nearest-neighbour$(KNN_j[k])$ **then**
          $C_i[j++] \leftarrow$ KNN$_j[k]$
        end if
      end for
    end for
  end for

B. Higher Level Segmentation

Similar to natural segment generation, at each level of the hierarchy, the k-mutual nearest neighbour principle is applied

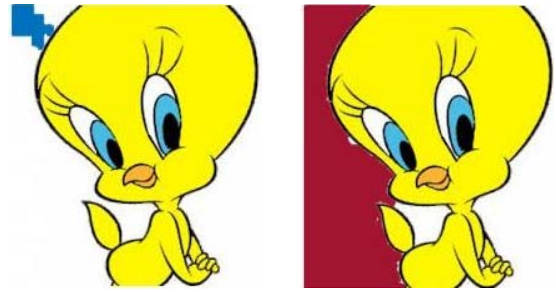


(a) (b) (c)

Fig. 6. Outputs of the natural level segmentation step. The first column (a) shows three input images. The last image of this column is from the Berkeley segmentation dataset [17]. The second column (b) shows the natural segments in random colours. The third column (c) shows the same segments with the mean colour value of each segment. All results were generated with k=3

over the segments of the previous level. At any level, a segment merges with its k-mutual nearest neighbours until no new segment can be found to merge as shown in Figure 7. Then, the current segment is marked off to allow not to grow further at this level. Algorithm 2 presents the method for higher level segment merging.

As a parameter-free method, this merging process continues at each level until all segments merge with each other at the last level. Figure 8 shows an example of segmentations formed by the proposed method at different levels. Figure 9 presents segmentation results for some images from the Berkeley segmentation dataset [17]. Note that only one level of segmentation is shown here.



(a) (b)

Fig. 7. (a) A natural segment at the first level of segmentation (shown in blue). (b) A higher level segment (shown in red) after the segment in (a) is merged with all of its k-mutual nearest neighbours

Algorithm 2 Segment merging at each higher level

Input: prev_level_segment, $P = \{P_1, P_2, ..., P_N\}$
Output: current_level_segment, $S = \{S_1, S_2, ..., S_M\}$
  $mergeable[P_1, P_2, ..., P_N] \leftarrow 1$
  $sz \leftarrow \text{sizeof}(P)$
  $i \leftarrow 1$
  $j \leftarrow 0$
  $k \leftarrow 0$
  while i $\leq$ sz **do**
    $\eta \leftarrow \text{findMKNN}(KNN, mergeable, i)$
    if isempty(η) **and** $j \neq 0$ **then**
      $mergeable[i] \leftarrow 0$
      $i{+}{+}$
      $j \leftarrow 0$
    else if isempty(η) **and** $j = 0$ **then**
      $i{+}{+}$
    else
      $S_k \leftarrow [P_i, \eta]$
      for each η **do**
        $mergeable[\eta_n] \leftarrow 0$
      end for
      $j{+}{+}$
      $k{+}{+}$
    end if
  end while

IV. Experimental Results

The proposed method was implemented using a commercial software package (MATLAB 8.6, The MathWorks Inc., Natick, MA, 2015). To make the process run faster, we used the parallel processing toolbox of the MATLAB software. Evaluating the results produced by a segmentation algorithm is challenging as it is difficult to find canonical test sets providing ground truth segmentations. This is partly because manual delineation of segments in everyday complex images can be laborious. Furthermore, people often tend to incorporate into their segmentations semantic considerations which are beyond the scope of data-driven segmentation algorithms. For this reason, many existing algorithms show only a few segmentation results. The following sub-sections discuss the choice of benchmark dataset and performance metrics, along with performance results of the proposed algorithm based on them.

A. Benchmark Dataset

The Berkeley segmentation dataset and benchmark [17] for natural images is an important attempt to produce an extensive evaluation database for segmentation. It is widely used for evaluating segmentation performance. The BSD500 dataset contains 200 natural images. This is the newest dataset of the benchmark.

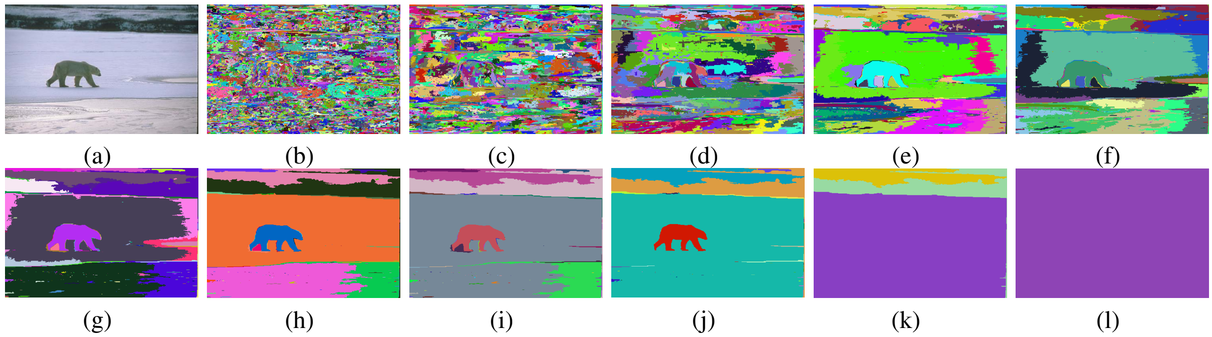


Fig. 8. (a) an image from the Berkeley segmentation dataset. (b)-(l) represents each level of segmentation by the proposed method. Segments are coloured randomly

This benchmark, has its own limitations, as can be noticed by the differences between subjects (manual segmentations by humans). In many cases, images are under-segmented and semantic considerations seem to dominate the annotated segmentations as shown in Figure 10.

Although this benchmark has shortcomings, it is widely used for segmentation performance evaluation. As such, we use it in evaluating the proposed algorithm. The images in this benchmarking database are either 321×481 or 481×321 in dimension.

B. Performance Metrics

To evaluate the performance of any segmentation method, performance metrics must be used. Different segmentation approaches have used different measures to evaluate the performance. To evaluate the performance of the proposed method, we relied on the Berkeley benchmark [17] where five metrics are defined: *variation of information*, *rand index*, *segmentation covering*, *precision*, and *recall*. The later two metrics are generally used for contour boundary evaluation. This section presents a brief overview of the metrics.

1) Variation of information: The variation of information (VI) metric was introduced in Meila [18] for cluster comparison. Based on the average conditional entropy, it measures the distance between two segments using the following equation:

$$VI(S, S_1) = H(S) + H(S_1) - 2I(S, S_1) \tag{2}$$

Here, H and I represent the entropies and mutual information between two clusterings of data S and S_1, respectively. In the Berkeley benchmark, these clusterings are test and ground truth segmentations, respectively. Lower VI means less variation when compared with the ground truth.

2) Rand index: Originally introduced in Rand [19], the rand index (RI) is also used for general clustering evaluation. It operates by comparing the compatibility of assignments between pairs of elements in the cluster. The RI between test and ground-truth segmentations S and G is given by the sum of the number of pairs of pixels that have the same label in S and G and those that have different labels inthe two segmentations, divided by the total number of pairs of pixels. Variants of the RI have been proposed [20], [21] for dealing with the case of multiple ground-truth segmentations. Given a set of ground-truth segmentations $\{G_k\}$, the Probabilistic Rand Index (PRI) is defined as:

$$PRI(S, \{G_k\}) = \frac{1}{T}\sum_{i<j}[c_{ij}p_{ij} + (1 - c_{ij})(1 - p_{ij})], \tag{3}$$

where c_{ij} is the event that pixels i and j have the same label and p_{ij} its probability, and T is the total number of pixel pairs. Using the sample mean to estimate p_{ij}, eqn (3) amounts to averaging the PRI among different ground-truth segmentations. Higher values of PRI reflect better segmentation.

3) Segmentation covering: The overlap between two regions R and R_1, defined as:

$$O(R, R_1) = \frac{|R \cap R_1|}{|R \cup R_1|}, \tag{4}$$

has been used for the evaluation of pixel-wise classification tasks in recognition [22]. The covering of a segmentation S by another segmentation S_1 is defined as:

$$Seg\_cover(S_1 \to S) = \frac{1}{N}\sum_{R \in S}|R| \cdot \max_{R_1 \in S_1} O(R, R_1), \tag{5}$$

where N denotes the total number of pixels in the image. Similarly, the covering of a machine segmentation S by a family of ground-truth segmentations $\{G_i\}$ is defined by first covering S separately with each human segmentation G_i and then averaging over the different humans. To achieve perfect covering, the machine segmentation must explain all of the human data.

4) Precision and recall: Precision is the probability that a machine-generated boundary pixel is a true boundary pixel. Recall is the probability that a true boundary pixel is detected. These metrics are also used in pattern recognition and information retrieval with binary classification. Although the precision-recall curve for an algorithm is a rich descriptor of its performance, it is still desirable to distill the performance of an algorithm into a single number. The harmonic mean of precision and recall is known as the F-measure, which provides a single description of the segmented boundary.

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

Fig. 9. Visual results on the Berkeley BSD500 segmentation dataset. From left to right: first column (a) shows the original images from the dataset. The second column (b) shows the boundary maps of the segmented images. The third column (c) shows the segmented images, presented by the mean colour of each segment and the last column (d) shows the segments with random colouring. Among all levels visually the best results are shown using k=3

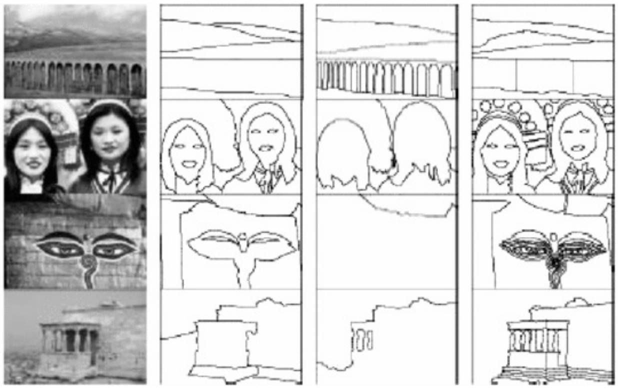


Fig. 10. An example of human annotations from the Berkeley segmentation data set (on the left are the original images). Note the large variations in human annotations and the difference in the underlying number of segments

C. Results on the Benchmark Dataset

The Berkeley segmentation benchmark uses a fixed threshold for all images in the dataset, calibrated to provide optimal performance on the training set. This scale is known as the optimal dataset scale (ODS). It also evaluates performance when the optimal threshold is selected by an oracle on a per-image basis, referred to as the optimal image scale (OIS). In our case, we do not train our algorithm on the training dataset to get the best segmentation level. Instead, we pass all levels of segmentation (except the last level that merges all segments into one segment) to the benchmark to decide required scales. As a consequence, slightly lower results are obtained for the ODS scale. Boundary and region results of the proposed method on the benchmark are reported in Tables I and II. Figure 11 presents the precision-recall curve.

TABLE I
BOUNDARY BENCHMARK RESULTS ON THE BSD500 DATASET: F-MEASURES ON TWO DIFFERENT SCALES (ODS AND OIS), AND AVERAGE PRECISION (AP)

ODS	OIS	AP
0.53	0.59	0.26

TABLE II
REGION BENCHMARK RESULTS ON THE BSD500 DATASET: SEGMENTATION COVERING (SEG_COVER), PROBABILISTIC RAND INDES (PRI) AND VARIATION OF INFORMATION (VI)

Seg_cover			PRI		VI	
ODS	OIS	Best	ODS	OIS	ODS	OIS
0.50	0.55	0.64	0.79	0.82	2.22	1.98

D. Comparative Results

As mentioned earlier, comparison of different segmentation algorithms is challenging as different algorithms put emphasis on different criteria while performing segmentation.

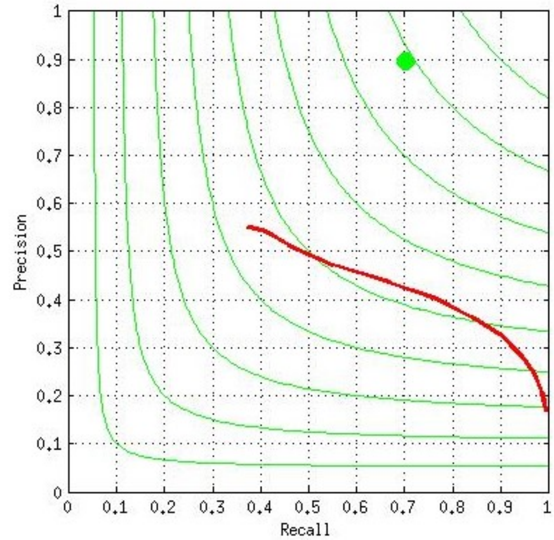


Fig. 11. Precision-recall curve on the BSD500 dataset

A comparison of some attributes of three relevant, existing algorithms and the proposed method is provided in Table III. In comparison to the proposed method, the existing techniques use additional measures to improve performance, such as, post-processing to merge small segments and optimising parameter values based on the dataset. As the proposed method is free of such processing, it is readily applicable to any dataset as is.

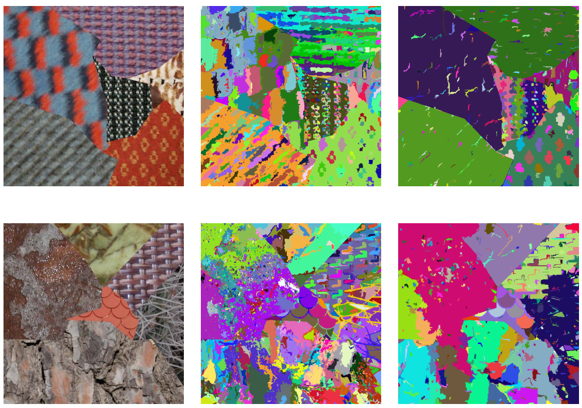
TABLE III
CHARACTERISTICS OF DIFFERENT SEGMENTATION ALGORITHMS

Method	Parameter-free	Hierarchical	Post-processing
Our method	Yes	Yes	No
[14] Guimarães et al.	No	Yes	Yes
[13] Ncut	No	Yes	-
[4] Felz-Hutt	No	No	Yes

Table IV presents comparison results with existing methods for the Berkeley BSD500 dataset. The proposed method provides superior results when compared to the other two hierarchical segmentation methods [13], [14]. Slightly lower results are seen while comparing with Felz-Hutt [4], which is a single level method. However, it utilises two thresholds and performs a post-merging to connect small segments. Moreover, this method may not generate comparable results when applied to other datasets. For example, it tends to loose image detail and produces overlapping segmentations when run on the Prague texture segmentation datagenerator and benchmark [23] as shown in Figure 12. In the first texture image it loses details of the image, while in the second image it overlaps multiple segments.

TABLE IV
REGION COMPARISON RESULTS ON THE BSD500 DATASET WITH EXISTING ALGORITHMS

Method	Seg_cover		PRI		VI	
	ODS	OIS	ODS	OIS	ODS	OIS
Proposed method	0.50	0.55	0.79	0.82	2.22	1.98
[14] Guimarães et al.	0.42	0.52	0.75	0.81	-	-
[13] Ncut	0.45	0.53	0.78	0.80	2.23	1.89
[4] Felz-Hutt	0.52	0.57	0.80	0.82	2.21	1.87



(a) (b) (c)

Fig. 12. Comparison on the Prague texture segmentation datagenerator and benchmark. From left to right: (a) images from the benchmark, (b) segmentation results of the proposed method, and (c) results of the Felz-Hutt method. The results of the proposed methods are level 5 of the segmentation hierarchy, with k=3. The default parameters for Felz-Hutt's method were used

V. CONCLUSION

A hierarchical image segmentation method was presented in this paper, which is free from adjustable parameters. It generates natural segments and subsequently merges them using the k-mutual nearest neighbour principle. The proposed method is simple and easy to understand, yet out-performs existing hierarchical segmentation algorithms when run on the Berkeley segmentation dataset. Initial segmentation results on a texture dataset are compelling. The current implementation only handles 2D images. However, this method can be easily extended to accommodate higher dimensional images. In future work, we will develop methods of assessing segmentation quality during the process so that segmentations that are clearly not viable (for example, lowest and highest levels of the segmentation hierarchy) can be discarded.

REFERENCES

[1] M. Wertheimer, "Laws of organization in perceptual forms." 1938.
[2] P. Sinha, "Once blind and now they see," *Scientific American*, vol. 309, no. 1, pp. 48–55, 2013.
[3] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 10–17.
[4] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.
[5] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," 1973.
[6] S. P. Lloyd, "Least squares quantization in PCM," *Information Theory, IEEE Trans on*, vol. 28, no. 2, pp. 129–137, 1982.
[7] M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag, and T. Moriarty, "A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data," *Medical Imaging, IEEE Trans on*, vol. 21, no. 3, pp. 193–199, 2002.
[8] S. Chen and D. Zhang, "Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Trans on*, vol. 34, no. 4, pp. 1907–1916, 2004.
[9] M. Gong, Y. Liang, J. Shi, W. Ma, and J. Ma, "Fuzzy c-means clustering with local information and kernel metric for image segmentation," *Image Processing, IEEE Trans on*, vol. 22, no. 2, pp. 573–584, 2013.
[10] M. Y. Mashor, "Hybrid training algorithm for RBF network," *International Journal of the computer, the Internet and Management*, vol. 8, no. 2, pp. 50–65, 2000.
[11] S. N. Sulaiman and N. A. M. Isa, "Adaptive fuzzy-k-means clustering algorithm for image segmentation," *Consumer Electronics, IEEE Trans on*, vol. 56, no. 4, pp. 2661–2668, 2010.
[12] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Trans on*, vol. 15, no. 11, pp. 1101–1113, 1993.
[13] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Trans on*, vol. 22, no. 8, pp. 888–905, 2000.
[14] S. J. F. Guimarães, J. Cousty, Y. Kenmochi, and L. Najman, "A hierarchical image segmentation algorithm based on an observation scale," in *Structural, Syntactic, and Statistical Pattern Recognition*. Springer, 2012, pp. 116–125.
[15] M. Tkalcic, J. F. Tasic *et al.*, "Colour spaces: perceptual, historical and applicational background," in *Eurocon*, 2003.
[16] G. Sharma, W. Wu, and E. N. Dalal, "The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *Color Research & Application*, vol. 30, no. 1, pp. 21–30, 2005.
[17] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.
[18] M. Meila, "Comparing clusterings: an axiomatic view," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 577–584.
[19] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971.
[20] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *Pattern Analysis and Machine Intelligence, IEEE Trans on*, vol. 29, no. 6, pp. 929–944, 2007.
[21] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.
[22] T. Malisiewicz and A. A. Efros, "Improving spatial support for objects via multiple segmentations," 2007.
[23] M. Haindl and S. Mikes, "Texture segmentation benchmark," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.