Volume 2, Issue 6, June 2017 ISSN (online): 2456-0006
International Journal of Science Technology Management and Research
Available online at: www.ijstmr.com

# Image Segmentation Using Hierarchical Merge Tree and Contour shape

**Asst. Prof. Nayana R.Kakad**
NDMVPS's KBTCOE, Nashik
.

**Asst. Prof. Shailesh M. Yawalkar**
NDMVPS's KBTCOE, Nashik

***Abstract:*** Current trends in image segmentation are to compute a hierarchy of image segmentations from fine to coarse. A classical approach to obtain a single meaningful image partition from a given hierarchy is to cut it in an optimal way, following the seminal approach of the scale-set theory. While intersecting in many cases, the resulting segmentation, being a non-horizontal cut, is limited by the structure of the hierarchy. In this paper, we formulate the tree structure as a constrained conditional model to associate region merging with likelihoods predicted using an ensemble boundary classifier. Final segmentations can then be inferred by finding globally optimal solutions to the model efficiently. We had also detected segments based on contours which provide us with boundary of object. We also present an iterative training and testing algorithm that generates various tree structures and combines them to emphasize accurate boundaries by segmentation accumulation.

## I. INTRODUCTION

IMAGE segmentation is one of the oldest and most challenging problems in image processing. As shown in [1], even for a human observer, it is hard to determine a unique meaningful segmentation of a given image. As promoted by Guigues et al. [2], a low level segmentation tool should remain scale uncommitted, because the structures which can be useful to high level task can have arbitrary size. In other words, segmentation should output a multi-scale description of the image. A usual approach to overcome the difficulty of finding a unique meaningful partition, and to satisfy the multi-scale property, is to compute a hierarchy of segmentations, which encodes a set of segmentations from fine to coarse. Another representation is the saliency map, originally introduced in [3], and independently rediscovered by Guigues et al. under the name of contour disappearance map in [2]. Recently, associated to a specific learning based algorithm, this representation has been popularized under the name of ultra-metric contour map [1]. Each threshold of the saliency map gives a segmentation result, and conversely, by stacking a set of segmentations satisfying a hierarchical property, one obtains a saliency map. A theoretical study and several characterizations of saliency maps can be found in [4], efficient algorithms to compute saliency maps are given in [5].

Our method falls into the object-independent hierarchical segmentation category. An early version of our process model with the merge tree model and a greedy inference algorithm discussed in [6] and [7] and was only applied to segmenting electron microscopy images, apart from which the contributions of this paper include:

- The hierarchical merge tree regeneration as a constrained conditional model with globally optimal solutions stated and an efficient inference algorithm developed, instead of the greedy tree model in [6] and [7].

• An iterative approach to diversify merge tree generation and improve results via segmentation accumulation.

• Experiments which extensively compare different variants and settings of the hierarchical merge tree model and show the robustness of the proposed system approach against image noise at testing time.

• Experimental results with state-of-the-art results on six public data sets for general image segmentation. Compared with current competitive hierarchical segmentation methods, ISCRA [8] and GALA [9], which consist a threshold-based greedy region merging strategy, our hierarchical merge tree model has two major advantages. First, the tree structure enables the incorporation of higher order image information into segmentation. The merge/split decisions taken in combination in a globally optimal manner ratherth by looking only at local region pairs. Second, our method does not require the threshold parameter to determine when to stop merging as in ISCRA and GALA, which may be so important to the results that need carefully tuning. Furthermore, our method is almost parameter-free given the initial super pixel over-segmentation. The only parameter is the number of iterations, which can be fixed as shown in the experiments on all the data sets.

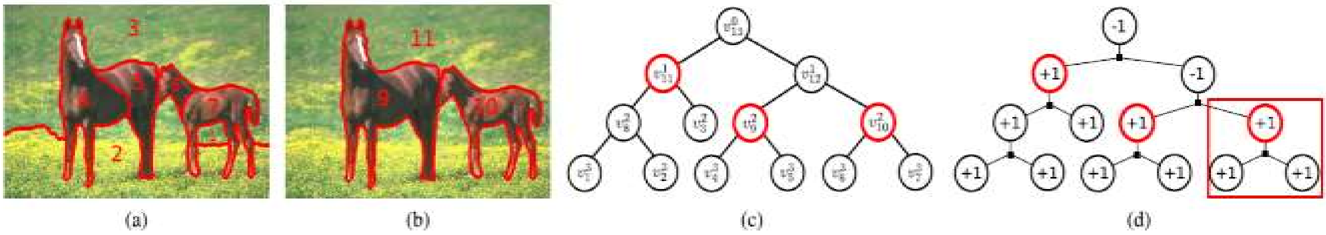



Fig. 1. Example of (a) an initial segmentation, (b) a consistent final segmentation, (c) a merge tree, and (d) the corresponding conditional model factor graph (Section III-B) with correct labeling. In (c), the leaf nodes have labels identical to those of the initial regions. The red nodes correspond to regions in the final segmentation. The red box in (d) indicates a clique in the model.

## II. RELATED WORK

### Hierarchical Merge Tree

Consider a graph, in which each node corresponds to superpixels, and an edge is defined between two nodes that share boundary pixels with each other. Starting with the initial over-segmentation, finding a final segmentation, which is essentially the merging of initial superpixels, can be considered as combining nodes and removing edges between them. This superpixels merging can be processed in an iterative manner: each time a pair of neighboring nodes is combined in the graph, and corresponding edges are updated. To represent the order of such merging, we use a full binary tree structure, which we call the hierarchical merge tree (or merge tree for short) throughout this paper.

### Boundary Classifier

To score each clique, we train a boundary classifier to predict the probability of each merge. To generate training labels that indicate whether the boundary between two regions exists or not, we compare both the merge and the split case against the ground truth under certain error metric, such as the Rand error [10] and the variation of information [11], [12]. The case with smaller error deviates less from the ground truth and is adopted.

## III. LITERATURE SURVEY

There are two different perspectives of image segmentation [1]. One is edge detection, which aims at finding edges between different perceptual pixel groups. The other one is region segmentation, which partitions an image into disjoint regions. Usually, edge detection focuses on assigning a binary label to each pixel with certain confidence indicating if it belongs to an edge or not and it does not guarantee closed contours. Though the closed contours and thus regions they encircle can be recovered from edges, such transformation with high accuracy is usually non-trivial. Apart from this, region segmentation does to find the cluster membership of per pixel, and related contours of an object. It can be trivially generated as the outmost points of a region. Most region segmentation methods also take benefit of the edge detection outputs as boundary cues to help with the search for correct partitioning. Our proposed system belongs to the region segmentation class, and in this section we emphasize reviewing existing related works in this class. First, we briefly summarize related edge detection works. Early edge detections are mostly based on image derivatives [2], [3] or filter banks responses [4], [5]. More recent works utilize richer information such as colors and textures.

## IV. PROPOSED SYSTEM

### ITERATIVE HIERARCHICAL MERGE TREE MODEL

In this model, the performance upper bound of the hierarchical merge tree model is determined by the quality of the tree structure. If all true segments exist as nodes in the tree, they may be picked out by the inference algorithm using predictions from well-trained boundary classifiers. If a desirable segment is not represented by any node in the tree, the system is unable to generate the segment. Hence, the merging saliency function, which is used to determine merging priorities, is critical to the entire performance. With a best merging saliency function, we can add the upper bound of performance and thus update segmentation accuracy. Statistics over the boundary strengths can be used to indicate merging saliency. We adopted a negated median of boundary pixel strengths as the primary representation of saliency. Since a boundary classifier is essentially designed to measure region merging likelihood, and it has advantages over simple boundary statistics because it takes various features from both boundary and regions, we propose to use the merging probabilities predicted by boundary classifiers as the merging saliency to construct a merge tree.

## Contour Detection

We link the contour detector with a generic grouping to produce high-quality image segmentations. This generic grouping algorithm consisting of two steps. Firstly, we show a new image transformation called the Oriented Watershed Transform for constructing a set of initial regions from an oriented contour signal. Second, using an agglomerative clustering procedure, we form these regions into a hierarchy which can be represented by an Ultra metric Contour Map, the real-valued image obtained by weighting each boundary by its scale of disappearance. Contour detection couples multi-scale local brightness, color, and texture cues to a powerful globalization framework using spectral clustering. The local cues, computed by applying oriented gradient operators at every location in the image, define an affinity matrix representing the similarity between pixels. From this matrix, we derive a generalized Eigen problem and solve for a fixed number of Eigen vectors which encode contour information. To recombine this signal with the local cues using classifiers, we obtain a large improvement over alternative globalization schemes built on top of similar cues.

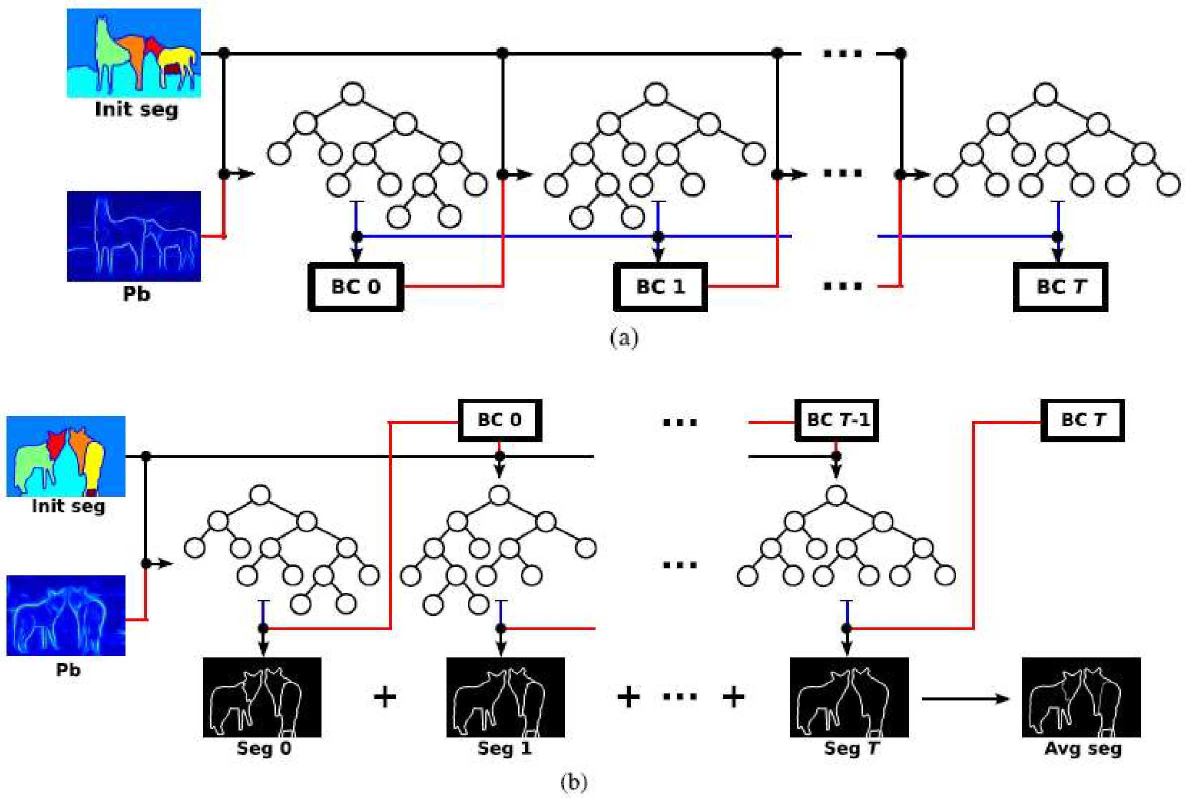



Fig. 2. Illustrations of (a) the training and (b) the testing procedure of the iterative hierarchical merge tree model. Starting with the fixed initial superpixels ("Init Seg"), the first iteration uses boundary probability ("Pb") statistics for merge tree generation, and the training procedure iteratively augments the training set by incorporating new samples from merge trees and trains a new boundary classifier ("BC"), which is used for merge tree generation from the same initial superpixels in the next iteration.

We develop an iterative approach to alternately collect training samples from a merge tree for the training of boundary classifiers and construct a merge tree with the trained classifier. As illustrated in Fig. 2(a), we initially use the negated median of boundary strengths to construct a merge tree, collect region merging samples, and train a boundary classifier $f^0_b$. Then, the boundary classifier $f^0_b$ is used for generating a new merge tree from the same initial superpixels from which new training samples are generated. We next combine the samples from the current iteration and from the previous iterations, remove duplicates, and train the next classifier $f^1_b$. This process is repeated for T iterations or until the segmentation accuracy on a validation set no longer improves. In practice, we fix the iteration number to T = 10 for all data sets. Eventually, we have a series of boundary classifiers $\{f^t_b\}_{t=0}^{T}$ from each training iteration. The training algorithm is illustrated in Algorithm. 1. At testing time, we take the series of trained classifiers and iterate in a way similar to the training process, as shown in Fig. 2: at each iteration t, we take the previous boundary classifier $f^{t-1}_b$ to construct a merge tree over the same initial superpixels $S_0$ and use the current classifier $f^t_b$ to predict each merge score in the merge tree, based on which a final segmentation $S_t$ is inferred. Lastly, we convert each segmentation into a binary closed contour map by assigning boundary pixels 1 and others 0 and average them for each image over all iterations to generate a segmentation hierarchy in the form a real-valued contour map. The testing algorithm is illustrated in Algorithm 2.

**Algorithm1: Iterative Training**.

**Input:** Original images $\{I_i\}_{i=1}^{N_{tr}}$, boundary maps $\{Pb_i\}_{i=1}^{N_{tr}}$, and iteration number $T$
**Output:** Boundary classifiers $\{f_b^t\}_{t=0}^{T}$
Generate initial superpixels $\{S_{oi}\}_{i=1}^{N_{tr}}$
**for** $t : 0, 1, \ldots, T$ **do**
  **if** $t == 0$ **then**
    Generate $\{Tr_i^0\}_{i=1}^{N_{tr}}$ from $\{S_{oi}\}_{i=1}^{N_{tr}}$ using $\{Pb_i\}_{i=1}^{N_{tr}}$
  **else**
    Generate $\{Tr_i^t\}_{i=1}^{N_{tr}}$ from $\{S_{oi}\}_{i=1}^{N_{tr}}$ using $f_b^{t-1}$
  **end if**
  Generate samples $\{(X_i^t, Y_i^t)\}_{i=1}^{N_{tr}}$ from $\{Tr_i^t\}_{i=1}^{N_{tr}}$
  Train $f_b^t$ using $\cup_{t'=1}^{t}\{(X_i^{t'}, Y_i^{t'})\}_{i=1}^{N_{tr}}$
**end for**

**Algorithm2: Iterative Testing.**

**Input:** Original images $\{I_i\}_{i=1}^{N_{te}}$, boundary maps $\{Pb_i\}_{i=1}^{N_{te}}$, and boundary classifiers $\{f_b^t\}_{t=0}^{T}$
**Output:** Hierarchical segmentation contour map $\{C_i\}_{i=1}^{N_{te}}$
Generate initial superpixels $\{S_{oi}\}_{i=1}^{N_{te}}$
**for** $t : 0, 1, \ldots, T$ **do**
  **if** $t == 0$ **then**
    Generate $\{Tr_i^0\}_{i=1}^{N_{te}}$ from $\{S_{oi}\}_{i=1}^{N_{te}}$ using $\{Pb_i\}_{i=1}^{N_{te}}$
  **else**
    Generate $\{Tr_i^t\}_{i=1}^{N_{te}}$ from $\{S_{oi}\}_{i=1}^{N_{te}}$ using $f_b^{t-1}$
  **end if**
  Score merges with $f_b^t$ and infer segmentations $\{S_i^t\}_{i=1}^{N_{te}}$
  Binarize $\{S_i^t\}_{i=1}^{N_{te}}$ to contour maps $\{C_i^t\}_{i=1}^{N_{te}}$
**end for**
$\{C_i\}_{i=1}^{N_{te}} = \{\sum_{t=0}^{T} C_i^t/(T+1)\}_{i=1}^{N_{te}}$

**Performance Evaluation**

We figure out the performance of using single ("SC") or ensemble boundary classifiers ("EC") with our hierarchical merge tree model. We also compare the proposed constrained conditional model ("CCM") formulation and greedy tree model ("Greedy") previously proposed in [9] and [10]. The greedy tree model portray the same hierarchical merge tree structure and scores each tree node only based on local merges the node is involved with, based on which a subset of highest-scored nodes that conform with the region consistency constraint are greedily selected. The training is done using the 200 training in BSDS300.

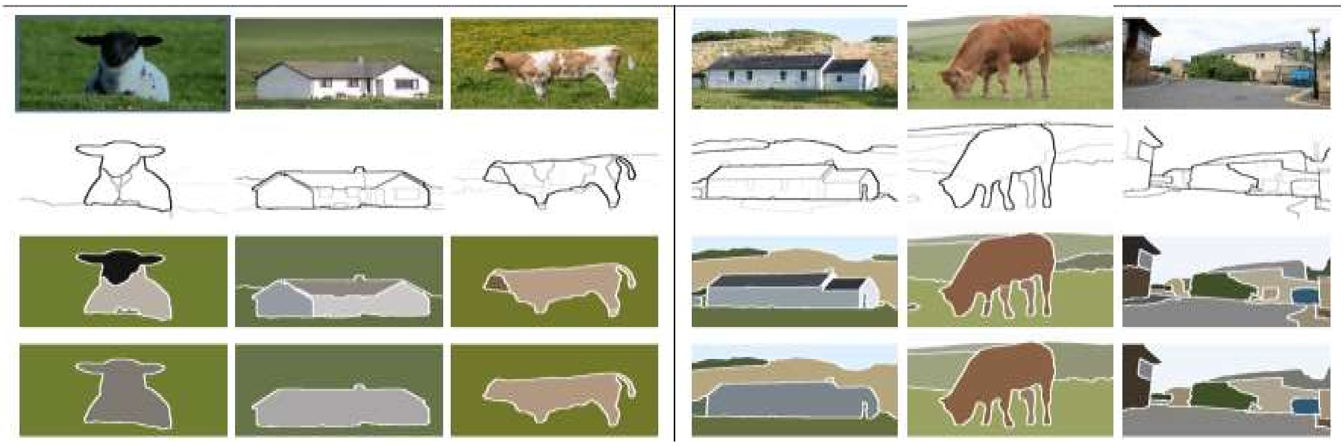
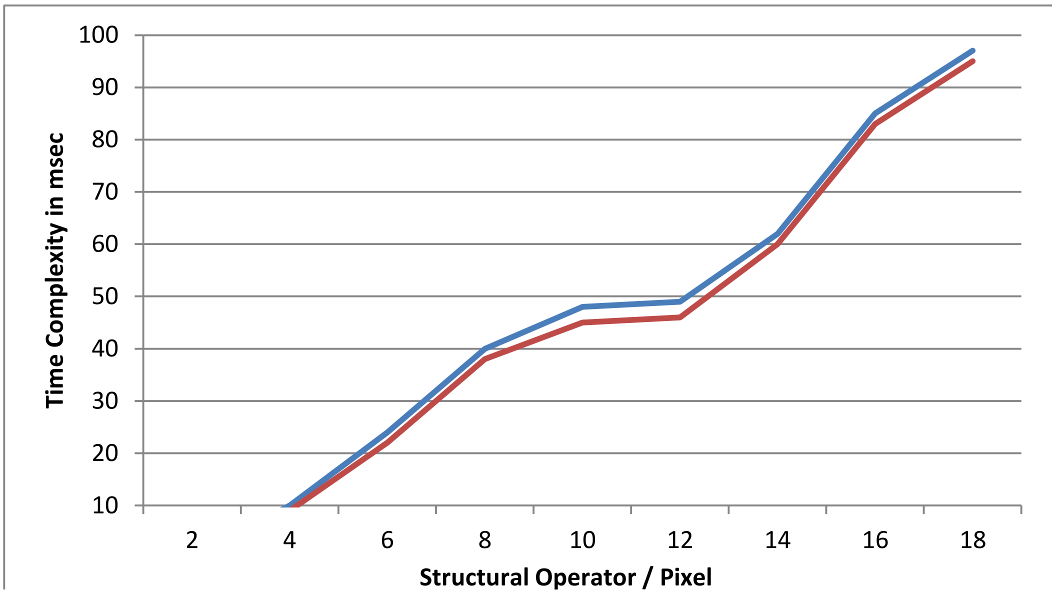


Fig 3: For each image, from top to bottom: the original image, the hierarchical contour map, the ODS covering segmentation, and the OIS covering segmentation. The training uses BSDS300 training images.

**Results:**




## CONCLUSION AND FUTURE WORK

We proposed a hierarchical image segmentation framework, namely the hierarchical merge tree model that limits the search space to one that is induced by tree structures and thus linear with respect to the number of initial superpixels. The framework allows the use of various merging saliency heuristics and features, and its supervised nature grants its capability of learning complex conditions for merging decisions from training data without the need for parameter tuning or the dependency on any classification model.

## REFERENCES

1.P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 33, no. 5, pp. 898–916, 2011.
2. L. Guigues, J. P. Cocquerez, and H. L. Men, “Scale-sets image analysis,” International Journal of Computer Vision, vol. 68, no. 3, pp. 289–317, 2006.
3. L. Najman and M. Schmitt, “Geodesic saliency of watershed contours and hierarchical segmentation,” IEEE Trans. on Pattern Analysis and Machine Intell., vol. 18, no. 12, pp. 1163–1173, 1996.
4. L. Najman, “On the equivalence between hierarchical segmentations and ultrametric watersheds,” Journal of Mathematical Imaging and Vision, vol. 40, no. 3, pp. 231–247, 2011.
5. Y. Xu, E. Carlinet, T. Geraud, and L. Najman, “Efficient com-putation of attributes and saliency maps on tree-based image representations,” in Proc. of Intl. Symp. on Mathematical Morphology, ser. LNCS, vol. 9082. Springer, 2015, pp. 693–704.
6. T. Liu, E. Jurrus, M. Seyedhosseini, M. Ellisman, and T. Tasdizen, “Watershed merge tree classification for electron microscopy image segmentation,” in Proc. Int. Conf. Pattern Recognit., Nov. 2012, pp. 133–137.
7. T. Liu, C. Jones, M. Seyedhosseini, and T. Tasdizen, “A modular hierarchical approach to 3D electron microscopy image segmentation,” J. Neurosci. Methods, vol. 226, pp. 88–102, Apr. 2014.
8. Z. Ren and G. Shakhnarovich, “Image segmentation by cascaded region agglomeration,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2013, pp. 2011–2018.
9. J. Nunez-Iglesias, R. Kennedy, T. Parag, J. Shi, and D. B. Chklovskii, “Machine learning of hierarchical clustering to segment 2D and 3D
10. R. Xiaofeng and L. Bo, “Discriminatively trained sparse code gradients for contour detection,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 584–592.
11. P. Dollár and C. Zitnick, “Structured forests for fast edge detection,” in Proc. Int. Conf. Comput. Vis., 2013, pp. 1841–1848.