Fast Implementations of Max/Min Tree for Interactive Image Segmentation

Xiaoqiang Huang, Mark Fisher and Yanong Zhu
School of Computing Sciences, University of East Anglia
Norwich, United Kingdom, NR4 7TJ
Email: xq.huang@uea.ac.uk, {mhf, yz}@cmp.uea.ac.uk

Abstract

This paper outlines a memory efficient data structure for building max/min trees and a fast strategy of reconstructing images from the max/min trees. The max and min tree are multi-scale image representations formed by considering a hierarchy of connected upper and lower level sets of pixels in an image, respectively. This paper aims to examine the performances of using max/min tree for image segmentation. Therefore a friendly graphic user interface has been implemented for the test purpose. This tool has been used to successfully extract measurements of cellular structures from microscope images of onion epidermis as the specimen undergoes mechanical test.

1 Introduction

Segmentation is a fundamental problem in the fields of image processing and computer vision. Its aim it to subdivide an image into its constituent regions or objects. The level to which the subdivision is carried out depends on the problem being solved. That is, segmentation should stop when the objects of interest in an application have been isolated. Thus, the ‘scale’ of the region or objects of interest needs to be taken into account in image segmentation.

Subdividing an image into a group of regions or objects at different scales requires a scale-dependent hierarchical representation of the image. This hierarchical representation of image should preserve scale causality [4], i. e. new level surfaces must not be created in the scale-space representation when the scale parameter is increased.

Tree based image decomposition have been drawing more and more attention in the last a few years as a number of new emerging industrial applications require tools that could decompose a raw image into a number of semantic regions or objects. These applications include object based image indexing, retrieval and compression (as the MPEG 4 and MPEG 7 standards require).

This paper describes an interactive image segmentation tool using the max/min tree [6] as the underlying decomposition structure of the image. The max and min tree have achieved some success in image filtering, segmentation and image retrieval. The segmentation tool proposed in this paper is underpinned by an efficient implementation of the max/min tree and by a fast strategy of performing iterative area opening/closing by the trees. This tool can be used to examine the performance of the max/min tree on solving real problems.

This paper is organised as it follows: a more detailed introduction to the max/min is presented in the next section, which also serves as an introduction of the efficient implementations of the max/min tree and fast strategies of implementing iterative area opening/closing via the max/min tree. Based on the material covered by this section, the interactive image analysis tool based on the max/min tree is introduced in section 3. An application of using the tool is also given in that section. Conclusions and further work will be given in the final section.

2 The Max/Min Tree

To understand the max tree, the image is considered to be a 3D relief. The nodes of the tree represent the connected upper level sets for all possible gray-level values. The leaves of the tree correspond to the regional maxima of the image. The links between the nodes describe the inclusion (father-child) relationship of the connected sets. An example of a max tree created from a simple image is shown in figure 1 [1].

2.1 Efficient implementations of the Max/Min Tree

Three sub-problems need to be addressed before the max tree can be successfully constructed:

A. Find all possible tree nodes from the original image

B. Create the father-child relationships (links) between each possible pair of tree nodes

[1]This figure is reproduced from [6]

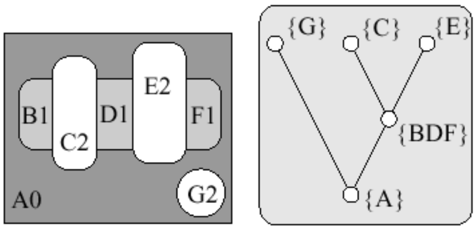

Figure 1: left: a simple image only with seven flat zones, right: the max tree for the image on the left

C. Create an efficient data structure to store information (region attributes) at nodes in the max tree.

The first sub-problem has been successfully addressed by a recursive flooding algorithm proposed in [6], however a number of approaches to the remaining sub-problems have been described. The max tree built by Meijster et. al. adopted an array as the underlying data structure for the tree. The advantage of using the array is that it provides direct access to its elements i.e. the complexity of visiting a random element in an array is $O(1)$. However, there are two major problems in using an array as the underlying data structure. Because the size of a max tree is unknown before the tree is built using a static array is not memory efficient. Secondly, when a node is deleted from the tree its corresponding element must also be removed from the array. Maintaining the array structure can become very cumbersome.

A more efficient data structure using a linked list associated with a hash table has recently been proposed by the authors of this paper [1]. The linked list provides dynamic allocation of computer memory thus the memory allocated for the max tree is just enough. However, the nodes in a linked list must be accessed serially so the complexity of visiting a random node in a linked list is $O(N)$ (N is here regarded as the length of a linked list). To overcome this problem a hash table is built immediately after the recursive flooding step. Using this table, the elements in the linked list are accessed directly so that the complexity is reduced to $O(1)$.

Our implementation of the max tree starts with applying the recursive flooding algorithm to recover scale tree nodes from the original image. Each flooding operation recovers a new node (and associated image pixels) and a father-child relationship between two nodes. These nodes are all appended to a linked list but due to the complexity ($O(N)$) of visiting elements in a linked list we do not visit tree nodes to establish relationships in this stage. Instead, a temporary linked list $Temp_L$ is created to store the information about the relationships and when the whole recursive flooding algorithm is finished, a hash table is created to index the linked list representing nodes of the created max tree. With the help of the hash table, the complexity of visiting a random scale tree node now becomes $O(1)$ (although the tree is still stored in the linked list). Finally, the information stored in the linked list $Temp_L$ is eventually retrieved and the father-child relationships are added as attributes at the scale tree nodes. The complexity of the whole process is $O(N)$. Experiments carried out in [1] confirms that by using the hash table the max/min tree can be created in linear time.

2.2 A fast strategy for implementing iterative area opening/closing using the Max/Min tree

Both Salember [8] and Meijster [5] claimed that once a max tree of an image has been constructed, computing an area opening at scale λ reduces to removing all nodes representing regions which have an area smaller than λ from the tree, and then reconstructing the image at the new scale.

The area opening operator is often applied as a nonlinear filter for preprocessing an image. In most cases, one is interested in applying an area opening operator at different scales λ to the same image. In this case, it is beneficial to retain the max tree structure unchanged after computing an area opening operation. Thus, once a max tree of an image has been constructed, computing an area opening reduces to analysing the max tree nodes and reconstructing the opened image only by those tree nodes which have an area greater than or equivalent to λ.

The strategy proposed by Meijster et. al. to reconstruct the opened image is described as follows. Their strategy actually consist of two steps. First, they assign the correct value for the output gray level of each node according to their area properties. Second, they assign each pixel the correct gray value according to the node the pixel belongs to. These two steps are processed serially.

We use different strategy to reconstruct the output image. The strategy is briefly described here and full details can be found in our recently published paper [2]. The area property of a node is always smaller than that of its father (except the root node as it does not have a father node). Thus if a node is determined to have a smaller area value than the threshold λ, there is no need to check its descendant nodes at all. Therefore, we start checking the root node and proceed to recursively check each of its children nodes. During each check, the pixel in the output image O is assigned the gray level value of the node it belongs to as long as the area property of the current node is greater than the λ. When a node whose area property is smaller than the λ is found, the current checking stops and all the pixels belonging to this node and its descendant nodes are output directly into the reconstruction image with the gray level value of the father node of the current node. With the help of the father-child relationships embedded in the tree, the descendant nodes of the current node can easily be visited in a recursive way.

The main differences between our strategy and that of Meijster et. al. 's is firstly not all max tree nodes are checked and secondly the checking and the reconstruction of the output image are done in a parallel way. Also the strategy is more memory efficient. Both strategies are data structure independent. In another words, both the strategies could work either using the array [5] or using the linked list combined with hash table [1] as the underlying data structures.

The experiments carried out in [2] confirms that our strategy runs about 25% faster that from Meijster et. al

3 The interactive image analysis tool

3.1 An introduction of the GUI

To assess the performance of using max/min tree for image segmentation, we have created an interactive image analysis tool [2] and an example is given in figure 2. We hope this free tool could benefit those whose are interested in using tree structure for image filtering and segmentation. The original image is displayed in the upper left corner. The images in the second column show the results of applying area opening/closing to the original image. The images in the third column are reconstructed from complementary tree nodes of the images in the second column. The images in the fourth column show regions at a specific scale λ. The image in the lower left corner is a result of applying the image in the fourth column as a segmentation mask to the original image (of course either a region from the max tree or the min tree can be used as the mask). A set of scrollbars are provided in the GUI so that an user can interactively change the scale parameter λ. Two parameters are displayed under the images in the fourth column. The first parameter (Scale) shows that current scale selected by the scrollbar. The second parameter (Real) shows the real scale of the region. This is simply because there is not always a region whose scale is equal to the current scale decided by the scrollbar and thus the image is the forth column does not change until there is a new region found whose scale is equal to that decided by the scrollbar.

3.2 An Application: Quantification of cellular structures in plants

While microscopy has been used extensively in a descriptive way for plant structure, its quantification for comparison and modelling is comparatively primitive. Defining

[2]The GUI can be downloaded from-http://www2.cmp.uea.ac.uk/∼xh/TreeGUI/

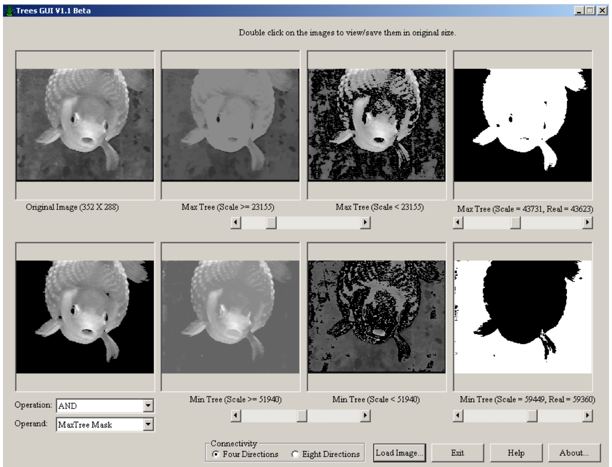

Figure 2: An example of using the GUI for object extraction

the relationship between the architecture of the tissues and their mechanical properties involves being able to describe the time-dependent change in cellular structures in plant tissues in response to applied mechanical stresses. The automatic extraction and measurement of features in 2-D cellular structure is a vital fist step in modelling the mechanical properties of these systems [7, 3].

The max/min tree GUI has been successfully used to interactively extract cellular structures of onion epidermis. The GUI allows the user to logically combine max and min tree masks, thereby enabling both cell walls (dark structures) and individual cells to be extracted and their areas (no. of pixels) measured by reconstructing min and max tree representations of the image, respectively (see figure 3 and 4).

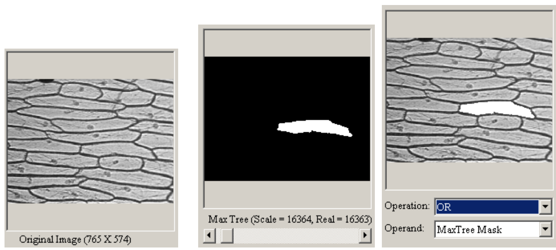

Figure 3: extract individual cell from the image

3.3 Limitations of the max/min tree representations

Although the max/min tree are proven very useful in solving some real problems, it is also proven that one can not always extract the region or object of interest using the

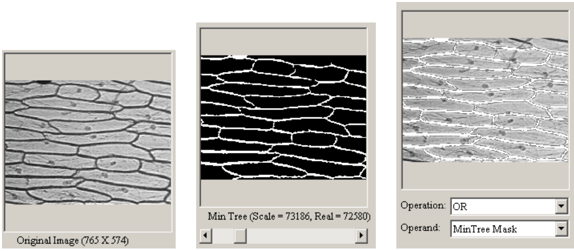

Figure 4: extract cell wall from the image

max/min tree.

For example, the person in left image in figure 5 can not be extracted either by using the max tree or the min tree. This is because the max tree tend to extract light object from the image and the min tree the dark object. However, the person in figure 5 is composed of by both light and dark objects. Thus, using a max/min tree to extract objects from image will fail in this situation.

Figure 5: left: the original image. middle: an example of failure of using the max/min tree to extract region of interest. right: use the image in the middle as the mask to the original image in the left

4 Conclusion

A fast approach for building and using max/min trees is proposed in this paper. The approach underpins an interactive image analysis tool that can be used segment objects from images. An application of the tool in analysing microscope images of onion epidermis shows the approach is useful when the image is constrained and the lighting can be controlled.

5 Acknowledgement

We would like to thank Dr. Andrew Smith of the Institute of Food Research [3] for his collaboration in supplying the images of onion epidermis.

References

[1] Xiaoqiang Huang, Mark Fisher, and Dan Smith. An efficient implementation of max tree with linked list and hash table. In *Proceedings of International Conference on Digital Image Computing-Techniques and Applications (Accepted)*, pages 299–308, Macquarie University, Sydney, Australia, December 2003.

[2] Xiaoqiang Huang, Mark Fisher, Yanong Zhu, Richard Aldrige, and Dan Smith. An efficient strategy for implementing iterative area openings using the max tree. In *Proceedings of Australian and New Zealand Intelligent Information Systems Conference (Accepted)*, pages 115–120, Macquarie University, Sydney, Australia, December 2003.

[3] A. Lee, B. Piggott, R. Aldridge, and M. Fisher. Extracting cell walls from onion epidermis cell images. In *BMVA symposium at the Royal Statistical Society*, London, April 2003.

[4] Tony Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.

[5] A. Meijster and M. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):484–494, 2002.

[6] Luis Garrido Ostermann. *Hierarchical Region Based Processing of Image and Video Sequences: Application to Filtering, Segmentation and Information Retrieval*. PhD thesis, Department of Signal Theory and Communications, Universitat Politecnica de Catalunya, Barcelona, Spain, April 2002.

[7] B. Piggott, A. Smith, M. Fisher, and R. V. Aldridge. The use of image processing in observing the effect of applied stress on onion epidermal cellular structures. In *Proceedings of 7'th Australian and New Zealand Intelligent Information Systems Conference*, pages 211–216, Perth, Western Australia, November 2001.

[8] P. Salembier, A. Oliveras, and L. Garrido. Anti-extensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4):555–570, 1998.

[3] http://www.ifr.bbsrc.ac.uk/