International Journal of Scientific and Technological Research www.iiste.org
ISSN 2422-8702 (Online)
Vol 1, No.1, 2015

Skip Graph Based Image Segmentation

Mustafa Aksu (Corresponding author)
Kahramanmaras Vocational High School, Sutcu Imam University
Kahramanmaras, Turkey
E-mail: m.aksu@ksu.edu.tr

Murat Canayaz
Yuzuncu Yil University, Van, Turkey
E-mail: mcanayaz@yyu.edu.tr

Ali Karci
Department of Computer Engineering, Inonu University
Malatya, Turkey
E-mail: alikarci@inonu.edu.tr

Abstract

In this paper, new approaches based on skip lists and skip graphs will be introduced for image segmentation and representation procedure. Skip graph is a graph approach which is derived from skip list data structure. Skip list data structure includes nodes and trees as in conventional graphs. The graph is a very common approach used in image processing applications. Skip graph approach mentioned in this study has not been tested yet in the image processing. Our study is important, since it is the first in this field. In this approach, image is divided to meaningful region using a variety method of segmentation. These regions will be represented in the skip graph structure to ensure the transformation graph of image. Various operations on images are performed on the graph. In this approach, many processes can be easily performed such as cleaning the image off dust, noise and removing color tones, adding or subtracting objects to images, since image is represented level by level.

Keywords: Skip graph, image segmentation, skip list, representation, graph theory.

1. Introduction

Graph is simply considered as the collection of dots or nodes and lines connecting these points, but the graph theory started to be one of the methods to solve many problems encountered in daily life. When graph theory was first introduced by Euler perhaps it was not intended to be a common use. However, it has already begun to be used in many fields [1, 2, 3, 4, 5, 6, 7] and is expected to be yet undiscovered source of inspiration for researchers in many areas. In the usage areas of graph theory image processing is also available which allows researchers to obtain more information about the image. Studies on the medical data on this subject [8, 9, 10] are evidence that confirms the importance of graph method.

In this study, unlike previously done, skip graphs which based on skip list were used for the image processing instead of the graph method. Some information is given about the skip lists in the second section and skip graph were described in the third chapter. Finally, representation of image as a skip graph, and the results is shown in the last section.

2. Skip List

Skip List algorithm is an algorithm similar to the structure of binary tree search proposed by Pugh [1, 11, 12]. In this algorithm; each element of the list, as in the graphs, considered as nodes and each element is expressed with a key and value. Linked lists are used in skip list algorithms and linked list elements are placed in different levels intended to provide convenience for searching. As known, in sequential linked lists process were done in O (n) complexity, by the help of skip list algorithm this complexity has been reduced. Levels of elements in this algorithm are randomly placed. Firstly, all elements are located at level 1, then starting from the left side and continuing upward with the next nodes pointers are appointed representing each level. Process made on this list can be made in a shorter time, simply by creating an index on the elements of a linked list. Guide time complexity of transactions is shown in Table 1.

Table 1. Time complexity of skip list

The time complexity of basic operations on a skip list is as follows:	
Operation	**Time Complexity**
Insertion	O(log N)
Removal	O(log N)
Search	O(log N)

Skip list as seen on table can be used for addition, deletion, and searching operations (Figure 1). If we look closely; time complexity of this algorithm becomes O (log N), if connected and ordered list are used. In comparison with other algorithms, it provides a significant time difference.

According to the addition algorithm, keys and values of elements to be added are acquired starting from top to bottom. The new value is added to the level at which the key is mapped. The pointer is updated according to the new value added. Detailed information about the skip list is available in Pugh [1]'s article.

Various data structures and algorithms were also created apart from skip list data structure such as skip list based priority queues [13], skip graphs [2], tiara (peer-to-peer network maintenance algorithm) [14] and corona [15].

Time complexity is O(N) for search, insertion and deletion processes when linked and ordered lists are used. On the other hand, the time complexity in which these processes are performed is O(lg N) in skip list data structure [11].

Representation of a linked list containing the elements {A, B, C, D, F, G, H, J} as skip list [1] is shown in Figure 2.

```
Search(list, searchKey)
    x := list→header
    -- loop invariant: x→key < searchKey
    for i := list→level downto 1 do
        while x→forward[i]→key < searchKey do
            x := x→forward[i]
    -- x→key < searchKey ≤ x→forward[1]→key
    x := x→forward[1]
    if x→key = searchKey then return x→value
        else return failure

Delete(list, searchKey)
    local update[1..MaxLevel]
    x := list→header
    for i := list→level downto 1 do
        while x→forward[i]→key < searchKey do
            x := x→forward[i]
        update[i] := x
    x := x→forward[1]
    if x→key = searchKey then
        for i := 1 to list→level do
          if update[i]→forward[i] ≠ x then break
          update[i]→forward[i] := x→forward[i]
        free(x)
        while list→level > 1 and
            list→header→forward[list→level] = NIL do
          list→level := list→level – 1

Insert(list, searchKey, newValue)
    local update[1..MaxLevel]
    x := list→header
    for i := list→level downto 1 do
        while x→forward[i]→key < searchKey do
            x := x→forward[i]
        -- x→key < searchKey ≤ x→forward[i]→key
        update[i] := x
    x := x→forward[1]
    if x→key = searchKey then x→value := newValue
    else
        lvl := randomLevel()
        if lvl > list→level then
            for i := list→level + 1 to lvl do
                update[i] := list→header
            list→level := lvl
        x := makeNode(lvl, searchKey, value)
        for i := 1 to level do
            x→forward[i] := update[i]→forward[i]
            update[i]→forward[i] := x
```

Figure 1. Skip list algorithms (searching, deletion and insertion) [1]

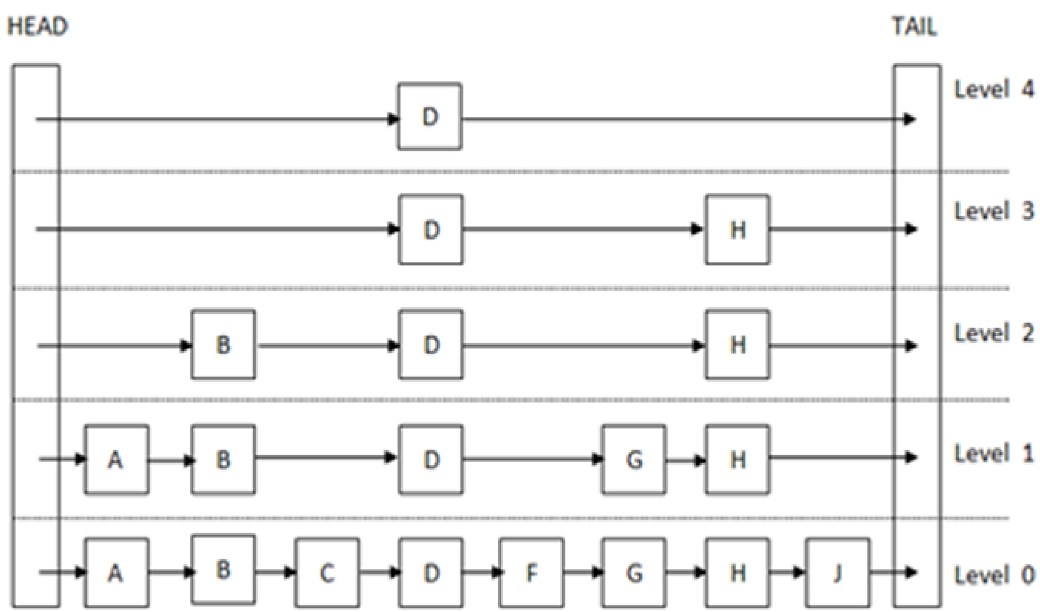


Figure 2. Skip list (Data structure based on a linked list)

Skip graphs [2] are developed by using above mentioned skip list algorithms, and thus intended to make transactions by means of graph simulation.

3. Skip Graph

Skip graph is a new approach inspired from skip list [2] to be used for searching purposes in peer to peer networks. In skip list, on each level operations are being performed on a single ordered list, however in this approach operations are performed on multiple linked lists. In addition, in this approach a random unique key value and m (v) vectors that indicates this key are allocated for each node v on levels. Provided that each v and w nodes have the same labels, are located within the same level.

In addition, each node in the skip graphs also contains information about adjacent nodes. In Table 2, variable information that is contained in each node of skip graph is given as summary.

Table 2. Each node of skip graph (information) [2]

Variable	Type
key	Resource key
neighbor[R]	Array of successor pointers
neighbor[L]	Array of predecessor pointers
m	Membership vector
maxLevel	Integer
deleteFlag	Boolean

According to Aspnes, the following conditions must be provided for each node-x which is available on an ideal skip graph on the basis of given variable values in the table.

Let xRi (xLi) be the right (left) neighbor of x at level i.

If xLi, xRi exist:

$$xLi = xLi^{k}\text{-}1$$

$$xRi = xRi^{k}\text{-}1.$$

$$\text{Invariant (} xLi < x < xRi. \text{ And } \quad xLiRi = xRiLi = x)$$

k : shows right or left adjacent order at its i^{th} level.

Figure 3 shows relationship between nodes adjacent to node- x in an ideal skip graph.

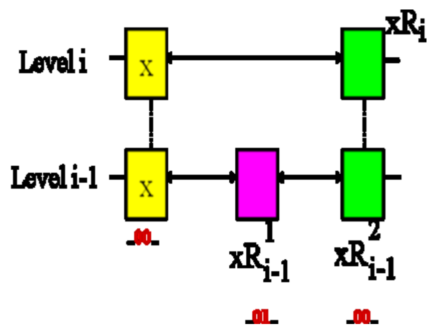


Figure 3. Relationship between nodes [2]

Detailed information about the skip graph; i.e algorithms of searching, adding, deleting operations can be reached from Aspnes and Shah's article [2]

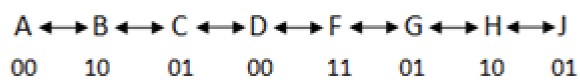


Figure 4. An eight member linked list

Figure 5 shows how the 8 member list is expressed as skip graph as given in Figure 4 with the membership values. In Figure 5, Level 0 is an expression of the complete list of all the nodes in it. And in Level 1, the nodes are grouped according to membership values. Initial membership values of A, C, D, G, J nodes are "0" and initial membership values of B, F, H nodes are "1", and hence grouping in Level 1 is done according to this.

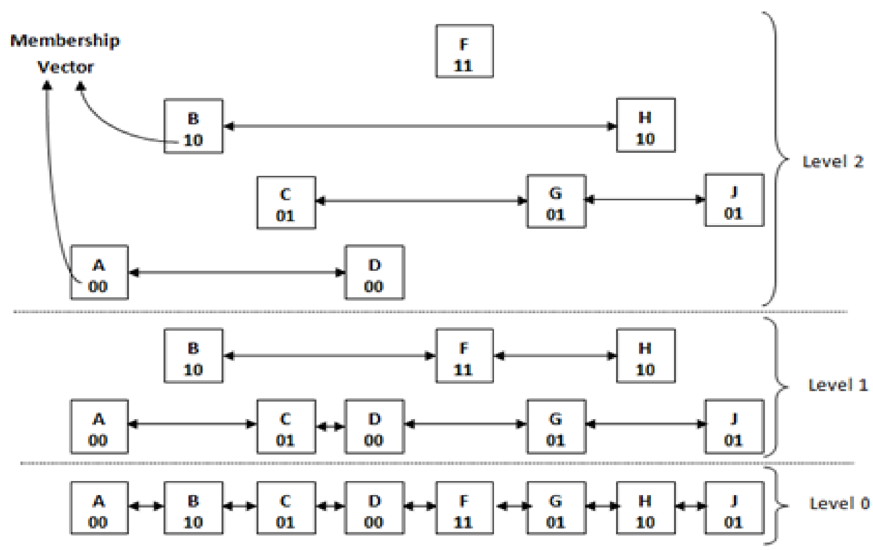


Figure 5. Representation of eight-member skip list as skip graph from Figure 4.

Whereas in Level 2, those which has membership values "00" compose first group, which has "01" is another group, those which has "10" is third group and finally the ones with "11" is divided into last group. If Level 1 and Level 2 groupings are checked, new lists (groupings) are composed according to membership values. Membership values can be created by utilizing features of regions, such as color and adjacency. By using these membership values, segmentation is performed in skip graphs.

4. Skip Graph Based Image Representation and Segmentation

An image which will be presented as skip graph:

a) Divided into regions (R1,…, Rn) as in Figure 6 by considering the properties and each region (R1, R2, ...) represents a node in skip graph.

b) Or each pixel which makes up the image represents a node.

That each pixel of an image is taken as a node of a skip graph will lead to problems for large images. Therefore, image must be divided into regions and it should be represented in levels in skip graph. As pre- transaction, region growing process or watershed process must be used to build regions [3]. Detailed information about the construction of regions and their algorithms, articles of Tremeau and Colantoni [3] can be viewed.

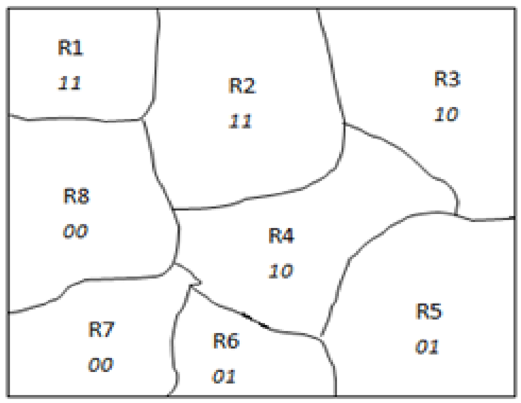


Figure 6 - Image (Region Based)

Then image is divided into regions by using pre-transaction as in Figure 6 and represented in levels as skip graph as in Figure 7. Level 0 of Figure 7, is the same as image in Figure 6.

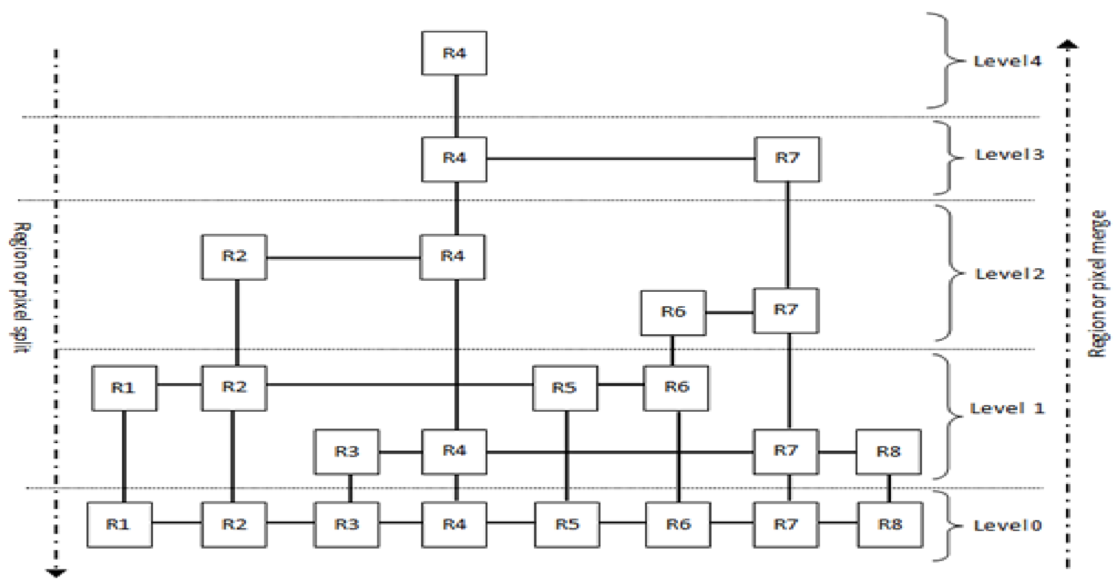
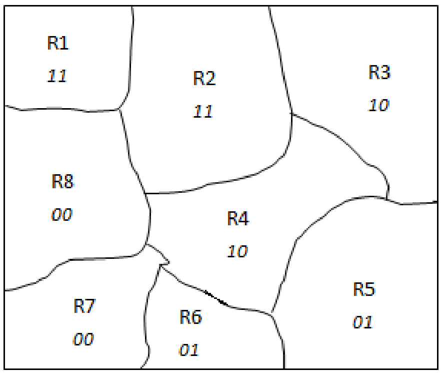
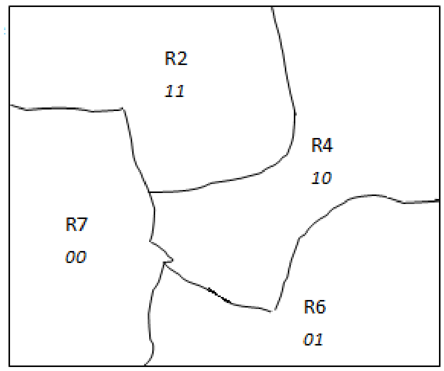


Figure 7 - Skip graph representation of the image as levels in Figure 6

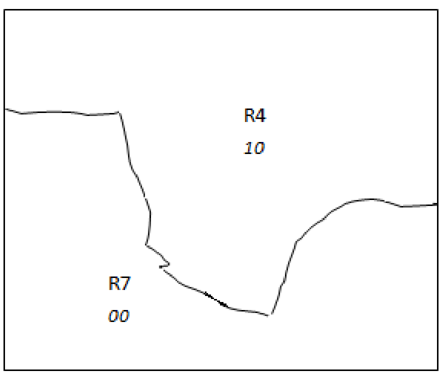
Level 0 of Figure 7 represents all regions of image as a linked list. Level 1 represents the first degree merging process according to region properties (color, density, adjacency, etc). Some regions in Level 1 combines as R2=R2 U R1; R4=R4 U R3 ; R6=R6 U R5 ; R7=R7 U R8 ; due to similarities of their properties and adjacency (membership vector). Similarly in Level 2, R7=R7 U R6; R4=R4 U R2 is observed. If the process continued in this way, whole image representation is achieved from regional level to complete image. In addition, each level also represents the segmentation of image according to its regional characteristics. While the top level represents the entire image (Level 4).



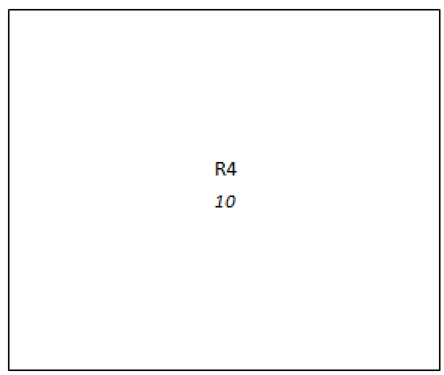
a) Level 0-1



b) Level 2



c) Level 3



d) Level 4

Figure 8 - Status of image in each level in Figure 7

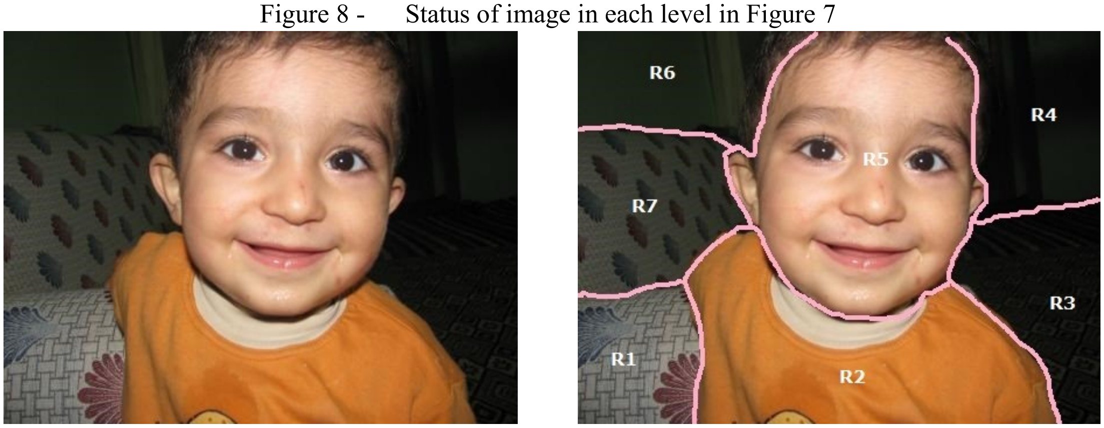


Figure 9- Region based image segmentation

4. Discussion

Image based studies of Graph Theory done mostly in the areas of representation of image as graph and segmentation process by using graph method (Figure 9). In this context, a new approach in our study shows that image can be represented as skip graph. Representation of the image as skip graph makes it possible to use it in segmentation process as well. Looking at Figure 5, if structure between Level 0 - Level 2 adapted to image, it can be used in segmentation process gradually according to its membership values. Skip Graphs can be used in peer to peer network as well as in image segmentation.

5. Result

In this study a new approach, representation of the image as skip graph and usage of this representation in image segmentation is emphasized. Mid results usually cannot be achieved in segmentation process. In this example each level (level 0, Level 1, Level 2, ..) represents different phase of the segmentation process that's why it is required to take into account the level for which the segmentation is requested.

In our study, firstly, it's mentioned that image must undergo image processing applications such as (watershed process / region growing process) before representing it as skip graph.

Without pre-processing each pixel of the image need to show by a node. In large images problems will arise due to large numbers of nodes. Therefore, each region of image will be represented by a single node in skip graph. Later, these regions will be joined according to different characteristics (membership value) starting from bottom level to top levels (Figure 7). Thus, the theoretical knowledge is given about the realization of segmentation process gradually.

In can be evaluated that this study can be used as source in various fields especially in image processing applications. Moreover, by adding a node to the image or by subtracting a particular region which has membership value, (removing a particular color tone) or noise cleaning or even shade cleaning operations can be performed on skip graph algorithms[2] by using addition and deleting.

References

[1]. Pugh, W., Skip Lists: A Probabilistic Alternative to Balanced Trees. Communications of the ACM, 33(6):668–676, 1990.

[2]. Aspnes, J. and Shah, G., "Skip graphs," in Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 384–393, 2003.

[3]. Tremeau A., Colantoni P., Region Adjacency Graph Applied to Color image segmentation, University Jean Monnet, Institut d'lngenierie de la Vision – IEEE Trans Image Process.;9(4):735-44, 2000.

[4]. Denis N. G., Christos A. F., Automatic synthesis of mathematical models using graph theory for optimisation of thermal energy systems, Volume 48, Issue 11, Pages 2818–2826, 2007.

[5]. Yin S., Zhao X., Wang W., Gong M., Efficient multilevel image segmentation through fuzzy entropy maximization and graph cut optimization, Pattern Recognition, Volume 47, Issue 9, Pages 2894-2907, ,doi:10.1016/j.patcog.2014.03.009, 2014.

[6]. Kalinin P., Sirota A., A graph based approach to hierarchical image over-segmentation, Computer Vision and Image Understanding, Volume 130, Pages 80-86, doi:10.1016/j.cviu.2014.09.007, 2015.

[7]. Wang H. , Zhang H., Ray N., Adaptive shape prior in graph cut image segmentation, Pattern Recognition, Volume 46, Issue 5, May 2013, Pages 1409–1414,doi:10.1016/j.patcog.2012.11.002

[8]. Huang Q. et al, A robust graph-based segmentation method for breast tumors in ultrasound images, Volume 52, Issue 2 Pages 266–275, 2012.

[9]. Jianbo S., Jitendra M., Normalized Cuts and Image Segmentation, IEEE Transactions on PAMI, Vol. 22, No. 8, 2000.

[10]. Zhang Y., Cheng X., Medical Image Segmentation Based on Watershed and Graph Theory, 3rd International Congress on Image and Signal Processing, 2010.

[11]. Pugh W. A Skip List Cookbook, Dept. of Computer Science, University of Maryland, College Park, UMIACS–TR–89–72.1, 1989.

[12]. Pugh W. Concurrent Maintenance of Skip Lists, Dept. of Computer Science, University of Maryland, College Park, TR–2222, 1989.

[13]. Lotan I. , Shavit N., ”SkipList-Based Concurrent Priority Queues,” in Proc. International Parallel and Distributed Processing Symposium, Mexico, pp. 263-268, 2000.

[14]. Clouser T., Nesterenko M., Scheideler C., “Tiara: A self-stabilizing deterministic skip list and skip graph,” Theoretical Computer Science, Vol. 428, pp. 18-35, 2012.

[15]. Nor R. M., Nesterenko M., Scheideler C., “Corona: A Stabilizing Deterministic Message-Passing Skip List,” Theoretical Computer Science, Vol. 512, pp. 119-129, 2013.