Analysis of Finite State Machine and Classical Iterative Cluster Labeling for 3D Images on Mobile Computing Strategies

Venkata Durga.Kiran Kasula[#1], Sudheesha Cheepi[*2], S.S.Reddy L[#3].
[1] *Dept of CSE, K L University.Green Fields, Guntur, India*
[2]*CSE, Green Fields, K.L.Univerisity,Vijayawada, India.*
[3] *Dept of CSE,LBR College of Engineering.Vijayawada,India*

***Abstract*- The model presents an efficient Finite State Machine on Hoshen-Kopelman (HK) using the nearest eight neighborhood rule .This approach uses the classical iterative Cluster Labeling method for the digital 3-D images. The model is going to propose the system where classical iterative can be implemented in passes and concentrates on test data, binary and color images on any mobile devices as well as randomly generated data for the cluster identification. Though conversation was provided along with a probable remedy for hardware blueprints.**

***Keywords*- Cluster, Cluster analysis, Hoshen-Kopelman algorithm, Finite State machine, mobile devices performances, hybrid method, Cluster Labeling.**

I. INTRODUCTION

Finite State machine is a behavior model composed of a finite number of states, transitions between those states, and actionsm, related to a flow graph in which one can scrutinize the way logic runs when certain conditions are met. It has finite internal memory, an input feature that reads symbols in a sequence, one at a time without going backward; and an output feature, which may be in the form of a user interface, once the model is implemented. The operation of an FSM begins from one of the states , goes through transitions depending on input to different states and can end in any of those available, however only a certain set of states mark a successful flow of operation.

Hoshen-Kopelman Algorithm*:* The Hoshen-Kopelman (HK) Algorithm is a simple algorithm for classification clusters on a grid, where a grid is a regular network of cubicle, where each cell may be either "occupied" or "unoccupied". The HK algorithm is a well-organized means of identifying clusters of adjoining cells. The general idea of the H-K algorithm is that we scrutinize through the grid looking for engaged cells. To each engaged cubicle we tendency to assign a label corresponding to the cluster to which the cell belongs. If the cell has zero engaged neighbors, then we assign to it a cluster label we have not yet used (it's a new cluster). If the cell has one engaged neighbor, then we assign to the current cell the same label as the engaged neighbor (they're part of the same cluster). If the cell has more than one engaged neighboring cell, then we choose the lowest-numbered cluster label of the engaged neighbors to use as the label for the contemporary cell. Furthermore, if these adjacent cells have contradictory labels, we must make a note that these different labels correspond to the same cluster.

The HK algorithm has previously been implemented using a finite-state machine (FSM) to improve upon its performance, but that implementation is limited to a neighborhood rule in which only the four cardinal neighbors are considered to be connected to a point in the map.

II. EXISTING SYSTEM

In the space we have the data sets. The study a variety of types of data sets typically requires the recognition of distinct subsets based upon some common qualities, a process known as cluster identification or simply clustering. Often cluster identification is performed using some type of distance metric, which defines the resemblance between two data elements. With spatial data, the distance metric used is often simple Euclidean distance, but this is not universal. A subclass of cluster identification techniques considers pure connectivity among cluster components rather than some measure of similarity. In the application of these techniques, the data are characteristically represented as a lattice, with each of each data point connected to some number of neighbors according to the lattice structure being for identified such cluster structures can generally be classified as recursive and iterative. Adjacency-based cluster recognition algorithms are the iterative type. The work is typically regarded as the first illustration of an iterative adjacency-based cluster identification algorithm. This method labels all clusters in a lattice through forward propagation. The lattice is traversed row-wise, and a cluster site is assigned the label belonging to any formerly labeled adjacent cluster site. Because this method does not label an intact cluster at once like the recursive method, an additional data structure must be maintained to keep track of the cases where numerous formerly disjoint clusters are exposed to be the same cluster as the lattice continues to be traversed. In these cases, a single cluster will have numerous labels associated with it, and the additional data organization fundamentally maintains a table of uniformity classes that can be worn to consign each cluster a single integrated label in a second pass of the lattice.

The major dynamic restraining the applicability of the recursive technique is usually considered to be the intact image having to vigorous in random access memory. The recursive technique requires a gigantic stack to accumulate confined and state variables for a bulky numeral of consecutive recursive function calls then dealing with images containing big substance. The stack is usually restricted by the system, and the hazard that a stack overflow may disrupt prolonged unsupervised or critical real time processes appears. One resolution to this hitch is to evade the recursion by converting the algorithm to iterative, which requires creating and organization the data structures. The solution is classical iterative labeling an amalgamation of random scan and hybrid methods.

III. **ENVIRONMENT AND MOTIVATION**

*A. Recursive technique***:**

The recursive, or depth-first, approach has its extraction depending on the application, This method makes either one or two passes in excess of the lattice. When a cluster site (meaning a site that needs to be assigned a cluster label) is encountered, the algorithm then examines each associated, or neighbor, site for other cluster sites that have not yet been assigned a label. This process is repeated recursively for all such neighbor cluster sites until every location in the complete cluster structure has been assigned the appropriate cluster label. At this the algorithm continues its bypass along the lattice until it encounters the next unlabeled cluster site. While this routine can be worn to classify all clusters inside a lattice, it has the supplementary advantage of allowing for the classification of a single cluster exclusive of exploratory the complete lattice, so long as at slightest one site of the cluster is acknowledged. Another benefit of the recursive technique is that it is not essential to uphold a supplementary data structure for supervision the cluster labels. However, in practical terms, this scheme can be inadequate by the require to accumulate the complete lattice in memory to conquer its pitiable vicinity of allusion, and, even more highly, the amount of hoard space obligatory for the number of recursive function calls for bulky clusters. Although the hoard space requirements for a purely recursive method have been exposed to mature unfairly bulky as the lattice size grows, has developed a recursive practice that addresses this constriction.

B. Depth first search:

A depth first search is a algorithm for verdict the biconnected component (a graph is a maximal biconnected sub graph) of an undirected graph and an enhanced adaptation of an algorithm for verdict the robustly allied components of a directed graph.

IV. **CLASSICAL ITERATIVE CLUSTERING**

This paper describes the procedure for digital images, and presents the consequences of numerous tests versus the hybrid technique, its counterpart, and versus the classical two pass approach, which is an iterative & recursive method that combines row scan with modernize operations on a global uniformity table to calculate the labels, optimized with approximately linear Amalgamation-Search algorithm for the manipulation of the uniformity chart. In classical recursive labeling the recursive function is called formerly for every picture element belonging to an object, and computations such as escalating object area, averaging picture elements co-ordinates, or recording extreme coordinates for the bounding box, can be performed anywhere within the recursive function. Such computations are equally possible with the hybrid technique, provided that they are performed at the accurate place. A single entitle to Label involves the labeling of several picture elements (a whole burst) in the forward scan, which is the accurate place for the computations, to ensure that they are performed once for each picture element. If a computation involves picture element coordinates, just note that during the forward scan in the pseudo code above they are referred as (m, y), because x is used to continue the preliminary position for the second advance scan. So the replica proposes the classical iterative clustering method with a three pass approach which the tests consist of the labeling of all objects and just the recognition of objects touching two opposite border of the image, in uniform pseudo-random binary images and in pseudo-random distributions of square blocks of foreground picture elements. The consequence of image size is also considered. We use these unreal sets as they are effortlessly reproducible neutral scenarios, which permit the study of an algorithm with respect to an intact collection of dissimilar parameters, such as density of foreground picture elements, average object size, or number of objects.

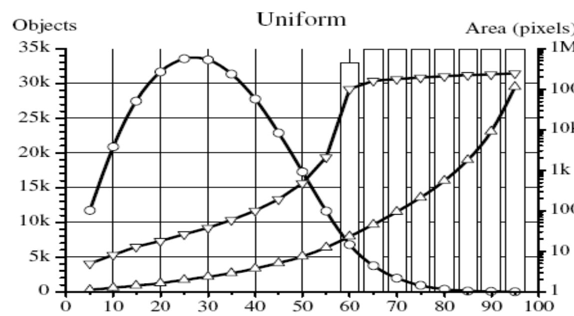

Fig 1 . Density of foreground pixels

In classical iteration clustering labeling the function is called unambiguously , every Picture element fit in to an object, and computations such as escalating object area, averaging picture elements coordinates, or recording extreme coordinates for the image box, can be performed anywhere within the function. Such computations are uniformly probable with this practice, provided that they are performed at the accurate place.

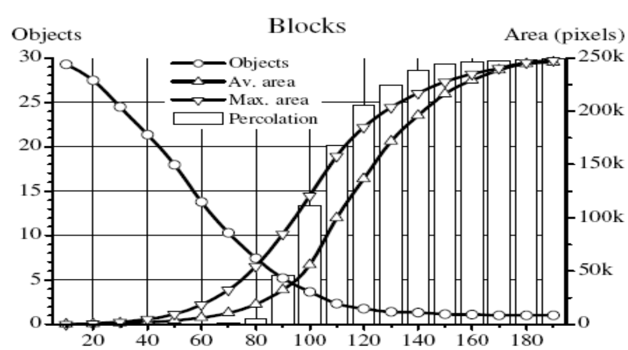

Fig 2 . Block Size

The aim of the paper is to dropping the number of function calls, and thus the stress on the system stack, lacking any considerable thrashing of performance, but observance all the precious individuality of the practice unfussiness, including no need of supplementary data structures, ease of accomplishment, flexibility, and the intact labeling of each object at a time and examine the images of 2 and 3 dimensions and also reduces the memory usage concepts.

Hybrid algorithm:

0 -> label
for all x, y if $p_{x,y} < 0$ {increase label, Label(x, y)}
Label(x, y)
while p_{x-1}, y < 0 decrease x
x -> m
while $p_{m,y} < 0$ {label ! $p_{m,y}$, increase m}
while x < m
if $p_{x,y-1} < 0$
increase y
Label(x, y − 1)
if $p_{x,y+1} < 0$
Label(x, y + 1)
increase x
end of Label

One of the main features of clustering is its capability to effortlessly distinguish the objects at some stage in their labeling, in a single pass over the image.

A. Classical Three-pass approach:

Classical three pass approach is an iterative method that combines row examine with modernize operations on a global uniformity table to calculate the labels, optimized with linear Amalgamation-Search algorithm for the exploitation of the uniformity table.

The fundamental algorithm requires a bulky amount of passes earlier than accomplishment to the concluding labels. Given an image with p picture elements, a labeling algorithm is said to be optimal if it uses O(p) time. Because the numeral of passes over the image depends on the content of the image, multi-pass algorithms are not measured optimal. To manage the figure of passes, one may exchange the direction of scans or directly manipulate the uniformity information.

1) One pass algorithm:

One-pass algorithms go through the image only once, but normally with an uneven access pattern. An algorithm in this collection scans the image to situate an unlabeled object picture element and then assigns the same label to all connected object picture elements.

2) *Two pass algorithm:*

Two-pass labeling algorithm scales linearly with the numeral of picture elements in the image, which is finest in computational complexity.

- Scanning phase: In this phase, the image is scanned once to assign conditional labels to object picture elements, and to verification the uniformity information among provisional labels.
- Analysis phase: This phase analyzes the label uniformity information to establish the final labels.

1) Third Pass algorithm:

Labeling phase: This third pass assigns final labels to object picture elements using a second pass through the image. Depending on the data structure used for representing the uniformity information, the analysis phase may be incorporated into the scanning phase or the labeling phase. One of the most resourceful data structures for representing the uniformity information is the amalgamation-search data structure.

Three -pass connected component labeling algorithm is based on two optimization strategies, the first one uses a decision tree to reduce the figure of neighbors examined during the scanning phase, and the second one streamlines the amalgamation-search algorithms to minimize the work desired to manage label uniformity information. The second strategy combines an effective way of using amalgamation-search algorithms for labeling with an array-based implementation for amalgamation-search.

B. Amalgamation search algorithm:

Disjoint set data structure is a data structure that keeps track of a set of elements partitioned into a numeral of disjoint (no overlapping) subsets, and each set is acknowledged by a single representative object restricted within the set. The representative may amend as the set is altered, but the representative must remain the similar as long as the set is unaltered. A disjoint-set forest is an implementation of the disjoint-set data structure that represents sets by rooted trees. Each node in the tree contains one member and points only to its parent node, and the root of the tree is the representative for the set.

An **amalgamation-search algorithm** is an algorithm that performs two useful operations on such a data structure:

- *Search*: Determine which set a particular element is in. Also useful for determining if two elements are in the same set.
- *Amalgamation*: Combine or merge two sets into a single set.

Three functions provide useful manipulations of a disjoint-set data structure: MAKE-SET(x), AMALGAMATION(x; y), and SEARCH-SET(x). The MAKE-SET function creates a novel set whose only member object is x. The AMALGAMATION function combines the two sets containing objects x and y. Finally, SEARCH-SET returns the representative of the set containing x.

Amalgamation-search algorithm:

```
function MakeSet(x)
    x.parent := x
    x.rank   := 0
function Amalgamation(x, y)
    xRoot := Search(x)
    yRoot := Search(y)
    if xRoot == yRoot
       return
    if xRoot.rank < yRoot.rank
       xRoot.parent := yRoot
    else if xRoot.rank > yRoot.rank
       yRoot.parent := xRoot
    else
       yRoot.parent := xRoot
       xRoot.rank := xRoot.rank + 1
function Search(x)
    if x.parent == x
      return x
    else
      x.parent := Search(x.parent)
      return x.parent
```

C. Implementing the FSM on HK algorithm

The matrix contains the data on which cluster identification is performed, and each data element within the matrix is a cell. For convenience, we shall refer to the nearest-four neighboring cells using the cardinal directions north, south, east, and west.

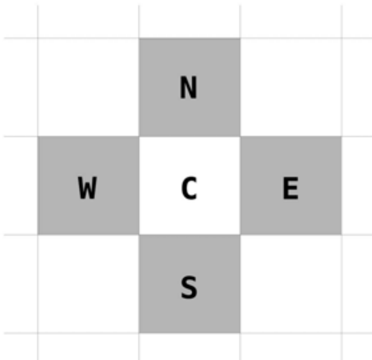

Fig 3. Four neighborhood directions

The nearest-eight finite-state machine implementation works similarly to the nearest-four. If the current cell needs to be supplementary to a cluster, the North West and East neighbors' values must be checked, along with the northeast and northwest. If any one of these neighbors belongs to a cluster, the current cell is added to that cluster. If the north neighbor does not belong to a cluster, but both the northeast and either the west or northwest neighbors do, there may be a need to perform a cluster merge (similar to west-north neighbor merges using the nearest-four neighborhood rule).

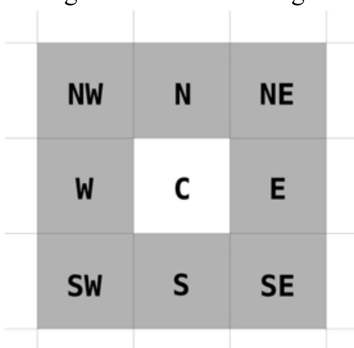

Fig 4. Eight neighbour-hood directions

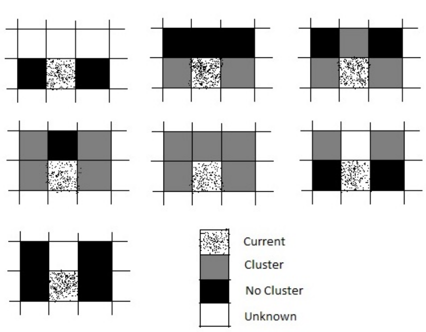

Fig 5. Seven states in nearest –eight FSM

D. Implementing the hybrid method on digital images:

Two sets with the recursive technique, with its iterative counterpart, with the two pass Amalgamation Search iterative technique, and with the hybrid technique. Because of their dissimilar approaches to the difficulty, the performance of the recursive technique depends mostly on the number of foreground picture elements, while the hybrid and iterative techniques should also show dependence with their spatial arrangement. Thus, the Uniform set is the worst case scenario for the hybrid and Amalgamation-Search techniques, while the classical recursive technique should not explain any noticeable preference. I recorded the average time per image for each algorithm, and monitored the recursive function calls (RC hereafter) of each of the two algorithms issuing recursive calls.

Fig 6. A 2-Dimensinoal image

Fig 7. A 3-Dimensional image

V. MOBILE PERFORMANCE

Applications that would obviously be suited to a fixed-location, workstation computing environment, it is posited that Classical iterative Clustering could be helpful in low-powered, entrenched, or mobile computing environments. One can visualize, for occurrence, researchers wishing to execute analysis on 2D and 3D images collected from field via manual observations or GPS on a computing device. While FSM-HK does not outperform HK in the computing environment used in the tests, the underlying causes of its lackluster presentation are explored, and potentially suitable hardware specifications are discussed.

VI. CONCLUSION

The classical iterative clustering in FSM over HK-algorithm is evaluating in west, northwest, north, northeast and east directions. In the model optimization strategies form an authoritative three-pass labeling algorithm that is faster than known labeling algorithms for 2D images. The classical iterative clustering performs much enhanced than preceding hybrid method. Cluster identification conserve the compensation of pure recursive labeling while really comforting the memory requisites.

ACKNOWLEDGEMENT

We are greatly delighted to place my most profound appreciation to Er.K.Satyanarayana Chancellor of K.L.University, Dr.L.S.S.Reddy Guide, Dr.K.Raja Sekhara Rao Principal, S.Venkateswaralu Head of the department, and Dr.Subramanyam in charge for M.Tech under their guidance and encouragement and kindness in giving us the opportunity to carry out the paper. Their pleasure nature, directions, concerns towards us and their readiness to share ideas enthused us and rejuvenated our efforts towards our goal. We also thank the anonymous references of this paper for their valuable comments.

REFERENCES

[1] M.L.Aldridge and Micheal W.Berry, "*Performances of a Finite-State Machine Implentations of Iterative Cluster Labeling on Desktop and Mobile Computing Platforms*", IEEE Computational science and Engg., vol21, No.11, Nov2009.

[2] M.L. Aldridge, "*Finite state Machine Approach to Cluster Identification Using the Hoshen-Kopelman Algorithm*," PhD dissertation, Univ. of Tennessee,May 2008.

[3] J.Martin-Herrero, "*Hybrid Object Labelling in Digital Images*", Machine Vision and Applications,vol.18,pp.1-15,Feb.2007.

[4] M.L. Aldridge, "*A Parallel Finite State Machine Implementation of a Nearest-Eight Hoshen-Kopelman Adaption for Landscape Analyisis*," Proc.45th Ann.Southeast RegionalConf.,pp.391-394,2007.

[5] Pankaj K. Agarwal, Lars Arge, and Ke Yi. "*I/O-efficient batched union-find and its applications to terrain analysis*" In SCG '06: Proceedings of the twenty-second annual symposium on Computational geomentry, pages 167-176, New York, NY, USA,2006.ACM Press.

[6] Qingmao Hu, Gouyu Quin, and Weislaw L. Nowinski. "*Fast connected-component labelling in three-dimensional binary images based on iterative recursion*", Comput. Vis.Image Underst.,99:414-434,2005.

[7] Kesheng Wu, Ekow Otoo, and Arie Shoshani "*Optimizing connected component labelling Algorithms*" In Proceedings of SPIE Medical Imaging Conference 2005, San Diego, CA, 2005.LBNL report LBNL-56864

[8] J.Martin-Herrero, "*Hybrid Cluster Identification,*" J.Physics A: Math. and General, vol.37 pp 9377-9386,2004.

[9] Coindreau, O.,Clotens, p., Vignoles, GL.: "*Direct 3D microscale imaging of C/C Composites with computed holotomography*", Nucl. Instrum.Methods Phys. Res., Sect. B, 200 295-302(2003).