Improved Connected Component Algorithm Using Run-based Approach

***Divya Kiran and Hariharan Ramasangu**

Department of Electronic and Communication Engineering, Faculty of Engineering and Technology,

Ramaiah University of Applied Sciences, Bengaluru- 560058

*Contact Author E-mail : divyakiran.ec.et@msruas.ac.in

Abstract

Connected Component Labelling (CCL) is an algorithm to separate connected groups. CCL detects the connected regions in an image and provides information about an image.Different CCL algorithm has been developed with a goal of less computation, complexity and more accuracy. CCL finds its widestapplications based on image processing such as object detection.In this paper, an improved CCL algorithm has been proposed and compared with Stefano-Bulgarelli(SB) Algorithm and improved Stefano-Bulgarelli Algorithm. The proposed approach utilizes a novel way of labelling the connected components using mathematical relations, reducing computational complexity. The proposed algorithm has been used to obtain the information related toconnected regions with less computations. Obtained results show that proposed algorithm is more accurate in obtaining numbers of the connected regions than Stefano-Bulgarelli(SB) Algorithm and improved Stefano-Bulgarelli Algorithm.

Key Words: Connected Component, Run-Length, Blob Detection, FPGA

1. INTRODUCTION

Connected Component Labelling finds its utility in various applications of image processing, computer vision, etc. Object detection is the most common application where CCL is used [1]. CCL can be defined as the algorithm labelling the vertices based on the connectivity and relative values of their neighbours. Connected components are labelled using the concept of equivalence classes of an equivalence relation that is defined by the vertices of the graph [2].

CCL algorithms involves raster scanning and label equivalence method. Raster-scan is the method of scanning in a rectangular manner one row at a time [1].

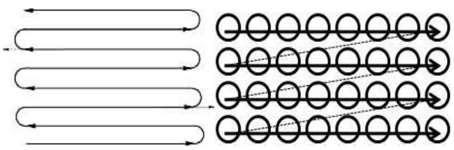

Fig. 1 Raster Scanning

In raster scanning as shown in Fig.1, the scan starts from the first column of row and ends at the last column then again starts from the first column of the next row. Raster-Scan provides the information of previous row but information of the next column in the same row or next row is not known. For the ease of analysis, binary image is preferred over RGB image for CCL. From the image, connected regions are identified as “Object”. It is also called binary large object (blob) which is a large cluster of connected pixels [3].

In this paper, a modified CCL algorithm has been proposed, and centroid and area of connected regions have been obtained. The rest of the paper is structured as follows: section II deals with the survey of different approaches of CCL algorithms, section III outlines the proposed algorithm and its implementation, and the results are discussed in section IV.

2. DIFFERENT APPROACHES OF CCL

Classical algorithm approach performs two raster scans of image or matrix. In the first scan the neighbourhood between pixels are identified and temporary labels are assigned. During the first scan, equivalent labels are stored into equivalence classes. In the second scan, temporary labels are replaced by equivalent labels. Neighbourhood can be identified using 8-connectivity or 4-connectivity. This paper only accounts 4-connectivity neighbourhood.

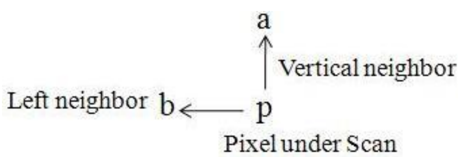

Fig. 2 Pixel Neighbourhood

For an image ‘*I*’, foreground pixel is ‘*F*’ i.e. ‘*1*’ and ‘*B*’ is the background pixel i.e. ‘*0*’. Let ‘*p*’ be the pixel under scan, ‘*a*’ be the immediate vertical neighbour and ‘*b*’ be the immediate left neighbour of ‘*p*’ as shown in Fig. 2. The labelled neighbour ‘*N*’ is given by (a, b).

$N_F = N \cap F$, where N_F is foreground labelled pixel. Classical or general CCL algorithm depicts five conditions needed to be checked while labelling:

1. If ‘*a*’ and ‘*b*’ are empty then ‘*p*’ has a new label.
2. If ‘*a*’ is ‘*F*’ and ‘*b*’ is empty then *label (p) = label (a)*.

3. If ‘*a*’ is empty and ‘*b*’ is ‘*F*’ then *label (p) = label (b).*
4. If *a* = *b* = *F* then *label (p) = label (a) or label (p) = label (b).*
5. If *a* = *F* and *b* = *F* but a ≠ b then *label (p) = label (a) or label (p) = label (b).*

The last condition determines the accuracy of the algorithm and is the main focus of the almost all the approaches discussed in the literature. Rosenfeld [4] have outlined the concept of equivalence tables ‘T’, where equivalent labels are stored as n-tuples in form of ($l_{i_2} \dots l_{i_n}$). In the first scan ‘T’ are ordered in increasing order. The second table ‘T’ is obtained by scanning ‘T’ moving the current entry ($l_{i_2} \dots l_{i_n}$) from ‘T’ to second table i.e. ‘T’ and replacing T ($l_{i_2} \dots l_{i_n}$) with l_{i_1} in remaining entries of ‘T’.

Gonzales [1] have suggested the usage of general, formal tools for handling equivalence relations to manage equivalence between labels. B(n) is the numbers of labels for n × n matrix. If B = 1, it can be filled up during the first scan and then the matrix representing the transitive closure of the relation, can be obtained by means of tools such as Warshall’s algorithm [5].

Stefano [2] have improved classical CCL algorithm called as SB algorithm, focusing on conflicts handling. In this the first scan temporary labels are assigned along with the identification of conflicts and resolving it. In the second scan, temporary labels have been replaced with the updated class identifier of its equivalence class.

Pandey [6] have improved SB algorithm addressing partial merging problem and modified equivalent handling loop to realize complete merging. In the “for loop” which is used to maintain equivalence labels, label lx is merged from C[l_j] to C[l_i], where C is the equivalence class. If C[l_x] = C[l_j] irrespective of whether $l_x < l_x$ or $l_x > l_x$. Once all such lx are merged, the class of label l_j is changed from C[l_j] to C[l_i]. This avoids partial merging as encountered in SB algorithm.

Elisa [7] have suggested a technique for CCL to improve quality of labelling and operation speed. The same problem has been addressed by Lee et al. [8] and they have outlined a parallel architecture for connected component labelling to overcome the limitation in Von Neumann architecture for real time processing.

Roy [9] have suggested a fast CCL algorithm using a parallel architecture that labels 512 X 625 size video in 5ms in the worst case. Similar approach has been propounded by Xiong [10], in which Dual Connected Component Labelling (DCCL) technique is implemented on FPGA. Since the classic CCL technique processes either black or white pixel, DCCL technique has been proposed which takes care of both pixels at the same time. DCCL uses the 2 × 3 mask and an equivalent table which stores the connectivity information of the pixels. The system has achieved the real time processing of 32 fps.

Paralic [11] has implemented an algorithm to reduce the time taken by the connected component labelling technique. The main focus has been the skin colour detection for the image size of 176 × 144 and the average computing time is 0.403ms.

3. PROPOSED CCL APPROACH

In the SB and improved SB algorithms, accuracy of the algorithm and minimum conflicts have been focused. The image processing applications, where these algorithms have been considered, require information of connected regions. These algorithms require another set of extensive computation for obtaining those information such as area and position of the connected region. Class identifier used in the algorithms update the labels but do not update itself i.e. C(1,4,4,4) may have only two labels (1 and 4) as shown in Fig.3, but to find the number of connected components, another computation to identify unique labels is needed. For these two algorithms one assumption is needed i.e. first row and first column of the image matrix should be ’0’ else the algorithm fails because each pixel is compared with same row previous column pixel as well as same column previous row pixel.

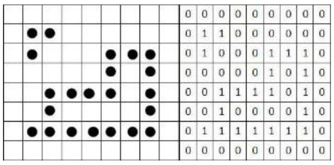

Fig. 3 Binary shape and matrix

Accounting these problems an approach with mathematical relations to identify the connectivity has been proposed. Proposed algorithm combines the both pixel based and run based approach. It identifies the row and column of the pixel and hence avoids the assumption of first row and first column be ‘0’. Algorithm compares the pixel under scan with immediate left pixel of the same row for ’Background’ or ’Foreground’; accordingly positions of start and end of connectivity chain is recorded.

Each individual block is labelled separately and then block of next row is compared with block of previous row to identify the neighbourhood. If neighbourhood is identified, the labels are updated else new label is defined. Equivalence class identifier is been incorporated to resolve any labelling conflicts. The uniqueness check has been done in the same time and hence the maximum label number provides information about the total number of objects in the image. The proposed approach is a row based scanning; each row is scanned and foreground pixels are checked. All connected foreground pixels are labelled as one block of run and information of start and end position of block is stored.

Positional information of connected block is termed as R, and $R = R_S(i, j) \cup RE(i, j)$ where, $R_S(i, j)$ and $R_E(i, j)$ is the position of start and end of the connected foreground blocks. After scan of two consecutive rows identified blocks are compared and checked for the identified relationships to establish neighbourhood between both the blocks. Let R_{CR} (i, j) and R_{PR} (i, j) be the blocks of current row and previous row then foreground neighbourhood N_F is defined as

$$N_F = R_{CR}(i, j) \cap R_{PR}(i, j)$$

If N (F) is not empty then only blocks are neighbours. When neighbourhood is identified then current row is assigned label of previous row only if $L_{CR} > L_{PR}$, where L_{CR} is current row label and L_{PR} is previous row label.

3.1 Conditions for Identifying Connected Pixels in a Row

Following conditions have been checked for pixel under scan $P(i, j)$ of 'n x m' matrix, to find the connected foreground block run $R(i, j)$ where, i is row position and j is column position.

1. If $P(i, j) = F$ and is at first column i.e. '$j = 1$' then it is start of run and $R_S(i, j) = R_S(i, 1)$
2. If $P(i, j) = F$ and is at last column i.e. '$j = m$' then it is end of run $R_E(i, j) = R_E(i, m)$
3. If $P(i, j) = F$ and $P(i, j-1) = B$ where, $j \neq 1$ and m then $R_S(i, j) = R_S(i, j)$
4. If $P(i, j) = B$ and $P(i, j-1) = F$ where, $j \neq 1$ and m then $R_E(i, j) = R_E(i, j-1)$

3.2 Identification of Neighbourhood between Runs of Different Rows

After scanning a row and storing start and end positions of runs it is needed to find the connectivity between the consecutive rows. Relations between start and end positions of previous block run and current block run have been identified. Based on these relations, some conditions have been defined, which enables to label the connected runs uniquely.

Let,

j_{CR} is column of current row pixel

j_{PR} is column of previous row pixel

$R_{S_{CR}}(i, j)$ is start position of connected run of current row

$R_{S_{PR}}(i, j)$ is start position of connected run of previous row

$R_{E_{CR}}(i, j)$ is end position of connected run of current row

$R_{E_{PR}}(i, j)$ is end position of connected run of previous row

$Diff_{SS}$ is difference of start positions of runs at current row and previous row

$Diff_{EE}$ is difference of end positions of runs at current row and previous row

$Diff_{SE}$ is difference of start position of run at current row and end position of previous row

$Diff_{ES}$ is difference of end position of run at current row and start position of previous row

The identified relations between start and end column of connected runs of consecutive rows are as follows

1. $Diff_{SS} = j_{CR}$ of $R_{S_{CR}}(i,j)$ - j_{PR} of $R_{S_{PR}}(i,j)$
2. $Diff_{EE} = j_{CR}$ of $R_{E_{CR}}(i,j)$ - j_{PR} of $R_{E_{PR}}(i,j)$
3. $Diff_{SE} = j_{CR}$ of $R_{S_{CR}}(i,j)$ - j_{PR} of $R_{E_{PR}}(i,j)$
4. $Diff_{ES} = j_{CR}$ of $R_{E_{CR}}(i,j)$ - j_{PR} of $R_{S_{PR}}(i,j)$

The following conditions have been defined to identify the connectivity between runs and consecutive rows

1. If $Diff_{SS}$, $Diff_{EE}$, $Diff_{SE}$ and $Diff_{ES}$ are positive, the corresponding rows are not connected.
2. If $Diff_{SS}$, $Diff_{EE}$, $Diff_{SE}$ and $Diff_{ES}$ are negative, the corresponding rows are not connected.
3. If $Diff_{SS}$ is positive and $Diff_{EE}$ is negative or vice versa rows are connected.
4. If $Diff_{SE}$ is positive and $Diff_{ES}$ is negative or vice versa rows are connected.

During first scan all distinct regions are labelled uniquely and equivalence classes have been updated. Let if $L_{CR} = 3$ and $L_{PR} = 1$ then 1 and 3 are equivalents and equivalence class is {1, 3}. In the next run all the equivalence classes are updated with the least value in the equivalence class. Block diagram of Proposed Algorithm is illustrated in Fig. 4.

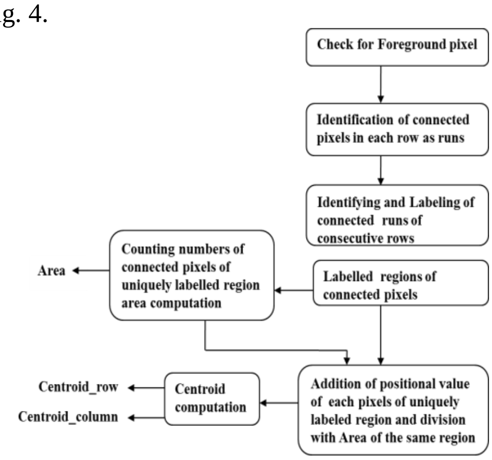

Fig. 4 Block Diagram of Proposed Algorithm

4. RESULTS AND DISCUSSION

For determining the efficiency of the algorithm different images have been processed through the algorithm. Standard binary images from online database have been utilized for comparison and artificial images have been used to find the accuracy of the algorithm. The obtained results shows the accuracy of the algorithm with respect to the identification of connected regions. Efficiency of the algorithm is determined by the requirement of computation needed to find the total number of connected regions and obtaining the related information.

Fig. 5 shows the artificial test images, utilized to find the accuracy of the algorithm in different environment. Both the test images have same objects but at different positions and brightness level. It has been observed that change in the position of the object changes its centroid value. This observation supports the idea of employing this algorithm for image processing based application. Effect of brightness level and camera angle on algorithm has been observed, and tabulated in Table 1.

Test images shown in Fig. 5 are utilized for comparison of proposed Algorithm with the SB algorithm [4] and Improved SB Algorithm [5].

Fig. 5 Test Images to Determine Efficiency of Proposed Algorithm [11]

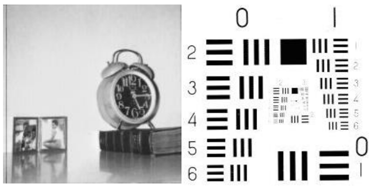

Fig. 6 Test Images for Comparison [12]

As shown in Table 1 proposed algorithm identifies all connected regions and more accurately. With respect to the conflicts proposed algorithm has the same number of conflicts as SB algorithm but more than Improved SB algorithm.

Table 1. Comparison of connected regions obtained

Test Images	Proposed Algorithm	SB Algorithm [4]	Improved SB Algorithm [5]
Test Image 1	97	284	284
Test Image 2	140	155	154

Though number of conflicts are more than improved SB Algorithm but it provides more accurate number of connected regions than improved SB Algorithm. The proposed approach provides an ease in obtaining the information of the objects with less complex computation. Proposed approach is also suitable for efficient hardware implementation because of its simplicity and accuracy.

5. CONCLUSION

This paper presents modified CCL algorithm using run-based approach has been proposed. This algorithm has been compared with the SB algorithm and improved SB algorithm. The proposed algorithm provides accurate object numbers in the image and also the number of object identification is more easy than the algorithms discussed in [4, 5]. Obtained results show that proposed algorithm is more accurate in obtaining numbers of the connected regions than algorithms of SB and improved SB. This algorithm supports efficient hardware implementation as well as is applicable for image processing based applications such as motion detection.

REFERENCES

[1] Gonzalez, Rafael C., and Richard E. Woods. (1992) Digital Image Processing. 42-45, Addison-Wesley,

[2] Di Stefano, L., & Bulgarelli, A. (1999). A simple and efficient connected components labeling algorithm. In *Image Analysis and Processing, 1999. Proceedings. International Conference on* (pp. 322-327). IEEE.

[3] Kiran, D., Rasheed, A. I., & Ramasangu, H. (2013, December). FPGA implementation of blob detection algorithm for object detection in visual navigation. In Circuits, Controls and Communications (CCUBE), 2013 International conference on (pp. 1-5). IEEE.

[4] Rosenfeld, A., & Pfaltz, J. L. (1966). Sequential operations in digital picture processing. Journal of the ACM (JACM), 13(4), 471-494.

[5] Warshall, S. (1962). A theorem on boolean matrices. Journal of the ACM (JACM), 9(1), 11-12.

[6] Pandey, J. G., Karmakar, A., Mishra, A. K., Shekhar, C., & Gurunarayanan, S. (2014, July). Implementation of an improved connected component labeling algorithm using FPGA-based platform. International Conference on Signal Processing and Communications (SPCOM) (pp. 1-6). IEEE.

[7] Calvo-Gallego, E., Aldaya, A. C., Brox, P., & Sánchez-Solano, S. (2012). Real-time FPGA connected component labeling system. In Electronics, Circuits and Systems (ICECS), 2012 19th IEEE International Conference on (pp. 593-596). IEEE.

[8] Lee, D. R., Jin, S. H., Thien, P. C., & Jeon, J. W. (2007, October). FPGA based connected component labeling. In Control, Automation and Systems, 2007. ICCAS'07. International Conference on (pp. 2313-2317). IEEE.

[9] Roy, P., & Biswas, P. K. (2008, December). VLSI implementation of fast connected component labeling using finite state machine based cell network. In Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on (pp. 238-243). IEEE.

[10] Xiong, J., Nguyen, T. M., & Wu, Q. J. (2011, May). Fpga implementation of blob recognition. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on* (pp. 125-131). IEEE.

[11] Paralic, M. (2012, July). Fast connected component labeling in binary images. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on* (pp. 706-709). IEEE.

[12] USC-SIPI. (2012) The USC-SIPI image database of the University of Southern California. [Online]. http://sipi.usc.edu/database/database.