

# Enhancing CCL Algorithms by Using a Reduced Connectivity Mask

Uriel H. Hernandez-Belmonte, Victor Ayala-Ramirez,  
and Raul E. Sanchez-Yanez

Universidad de Guanajuato DICIS  
hailehb@laviria.org, {ayalav,sanchezy}@salamanca.ugto.mx

**Abstract.** In this paper, we propose to use a Reduced Connectivity Mask (RCM) in order to enhance connected component labeling (CCL) algorithms. We use the proposed RCM (a  $2 \times 2$  spatial neighborhood) as the scanning window of a Union-Find labeling method and of a fast connected component labeling algorithm recently proposed in the literature. In both cases, the proposed mask enhances the performance of the algorithm with respect to the one exhibited by each algorithm in its original form. We have tested the two enhanced approaches proposed here against the fast connected component labeling algorithm proposed by He and the classical contour tracing algorithm. We have compared their execution time when labeling a set of uniform random noise test images of varying occupancy percentages. We show detailed results of all these tests. We also discuss the differences in behavior shown by the set of algorithms under test. The RCM variants exhibit a better performance than the previous CCL algorithms.

## 1 Introduction

Main tasks in computer vision include the detection and recognition of objects of interest in a binary image. Binary images are often the result of a feature detection step, where a binary feature is detected in a pixel-wise approach. This problem arises in image analysis tasks such as figure-background discrimination and motion detection, as two examples. Typically, images are preprocessed accordingly with the task and they are thresholded to output a binary image. A second step involves the association of connected pixels into single entities to be handled by further processing steps. A set of methods to perform this task are known as Connected Components Labeling (CCL) algorithms. The goal of a CCL algorithm is to label all the foreground pixels in a binary image. The label for each foreground pixel has to be the same for all the pixels belonging to the same connected component. Each connected component has to be labeled with a unique identifier. Two pixels are said to belong to the same component if there exists a path that links them.

Recent work on CCL has been directed towards: i) reducing the number of scans and ii) optimizing the connectivity mask and the definition and usage of the data structures for the labeling procedure. For the first case, the contour tracing

algorithm (Chang, Chen & Lu 2004) is an example. In this kind of methods, there is only a single scanning pass to the image because a set of heuristic rules guides the decision of what pixel is the next requiring analysis. For the second concern, two of the reference works are: i) the Union-Find approach, originally proposed by Fiorio and Gustedt (Fiorio & Gustedt 1996) and further enhanced by Wu *et al.* (Wu, Otoo & Shoshani 2005) and ii) the Connection-List approach proposed recently by He *et al.* (He, Chao & Suzuki 2008).

We propose to use a  $2 \times 2$  neighborhood as the spatial support of the connectivity mask for the CCL task. The rationale behind this design choice is that in such a region, all the pixels are connected among themselves and that the use of such a reduced connectivity mask (RCM) will speed up the computation of the label assignment for all the foreground pixels of the input image. The connectivity analysis implied by this shape of mask is performed by using a sort of operational look-up table. For any pixel under analysis, we access the look-up table entry that corresponds to the actual configuration of the neighborhood to determine what operations need to be executed. In this work, we compare the performance of our method against the enhanced Union Find approach (UF) (Wu, Otoo & Shoshani 2005) and to the Connection List (CL) method (He *et al.* 2008).

Rest of this paper is organized as follows: In Section 2, we review previous work for the CCL problem. The CCL problem is formulated in Section 3. A detailed presentation of the proposed method is addressed in Section 4. Results and their discussion are presented in Section 5. Finally, we present our main findings and conclusions in Section 6.

## 2 Taxonomy of CCL Methods

A number of methods have been developed to address the connected component labeling task. Suzuki *et al.* (Suzuki, Horiba & Sugie 2003) and Wu *et al.* (Wu, Otoo & Suzuki 2005) have classified the CCL methods in four main classes accordingly to the characteristics of the algorithms:

- i) Single-pass algorithms like those proposed by (Chang *et al.* 2004) and (Chang & Chen 2003),
- ii) Multi-pass algorithms, see, for example, (He, Chao & Suzuki 2007), (Wu, Otoo & Shoshani 2005), (Fiorio & Gustedt 1996), (Di Stefano & Bulgarelli 1999), (He *et al.* 2008), (Asano & Tanaka 2010), (He, Chao & Suzuki 2010a), (He, Chao, Suzuki & Wu 2009), (Wu, Otoo & Suzuki 2008),
- iii) Parallel computing approaches ((Greiner 94), (Han & Wagner 1990), (Hardwick 1997), (Krishnamurthy, Lumetta, Culler & Yelick 1994)), and
- iv) Hierarchical image representation methods ((Samet 1981), (Wang & Davis 1986), (Dillencourt & Samet 1992), (Samet & Tamminen 1988)).

In the three first classes (i, ii and iii) of the above taxonomy, the image under analysis is represented as a data matrix whose elements will be assigned a connected component label as the output of the algorithm.

Class iv) algorithms, instead, use more complex data structures to represent the binary image. Examples of data structures used for this end include bin-trees and quad-trees. In this work, we will focus on issues related to methods that use a data matrix representation of the image under analysis and it is out of the scope of this paper to study further details of the CCL methods belonging to the fourth category in the enumeration above.

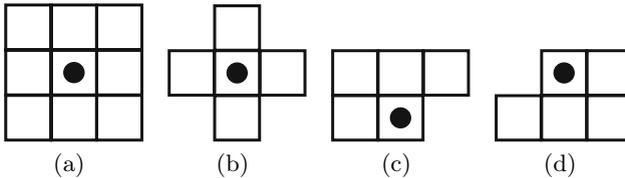
### 3 Methodology

#### 3.1 Overview of a CCL Algorithm

Any CCL algorithm needs to scan the input image a number of times depending on the scanning strategy, the connectivity mask used and the label equivalence handling method.

As we traverse the image on the first scan, we assign temporary labels to the foreground pixels. In order to ensure that all the pixels belonging to the same connected component are marked with the same label, each CCL algorithm uses a particular equivalence handling method. Some approaches use a set of heuristic rules (Chang et al. 2004) and some others use particular data structures designed for that purpose (Fiorio & Gustedt 1996), (He et al. 2009). Further scans of the input image may be needed to resolve the label assignment for all the foreground pixels.

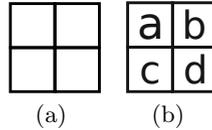
In each traversal, the image is scanned by using a connectivity mask (CM) that serves to analyze the connectivity among the pixels in the neighborhood defined by the spatial support of the CM. Different authors have proposed different CMs. We show several examples of previously used CMs in Figure 1. The spatial configuration of the CM is a critical factor in the performance of any CCL algorithm.



**Fig. 1.** Typical shapes of neighborhood masks used for connectivity analysis in CCL algorithms, the pixel under analysis is marked on them: (a)  $3 \times 3$  mask, (b) 4-neighbor mask, (c) forward scan mask and (d) backward scan mask

#### 3.2 Reduced Connectivity Mask for CCL Algorithms

Finding ways to simplify the connectivity analysis step is an important way of reducing the computational load required by CCL methods. That is why we propose a reduced connectivity mask (RCM) that covers only  $2 \times 2$  pixels. Figure 2 shows the RCM and the notation used for the pixels in this mask. For this type of mask we do not need to use a central pixel to perform the analysis.



**Fig. 2.** Reduced Connectivity Mask: (a) the spatial neighborhood, and (b) the notation used to identify each pixel in the mask

## 4 Our Approach

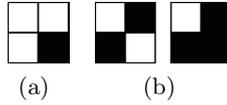
We analyze the use of two labeling strategies: i) Union-Find, originally proposed by Fiorio and Gustedt (Fiorio & Gustedt 1996) and enhanced by Wu *et al.* (Wu, Otoo & Suzuki 2005) and ii) the Connection-List approach proposed by He *et al.* (He et al. 2008), but using a  $2 \times 2$  connectivity (RCM) mask instead of their choices for the connectivity mask. We call these approaches as UF-RCM and CL-RCM respectively.

### 4.1 Labeling Procedure

The labeling procedure in CCL algorithms that use a RCM requires two image scans. The first scan is used to assign a temporary label to each foreground pixel in the image. The second scan is used to resolve the equivalence relationships between labels and to assign definitive labels to all foreground pixels in the image.

**Table 1.** Look-up table operations associated to each neighborhood configuration in the proposed approaches

Row	$a$	$b$	$c$	$d$	$L(x, y)$	<i>Union-Find</i>	<i>Connection-List</i>
0	0	0	0	0	NOP	NOP	NOP
1	0	0	0	1	$m+1$	NOP	NOP
2	0	0	1	0	NOP	NOP	NOP
3	0	0	1	1	$c$	NOP	NOP
4	0	1	0	0	NOP	NOP	NOP
5	0	1	0	1	$b$	NOP	NOP
6	0	1	1	0	NOP	<i>Union(c, b)</i>	<i>resolve(c, b)</i>
7	0	1	1	1	$b$	<i>Union(c, b)</i>	<i>resolve(c, b)</i>
8	1	0	0	0	NOP	NOP	NOP
9	1	0	0	1	$a$	NOP	NOP
10	1	0	1	0	NOP	NOP	NOP
11	1	0	1	1	$c$	NOP	NOP
12	1	1	0	0	NOP	NOP	NOP
13	1	1	0	1	$a$	NOP	NOP
14	1	1	1	0	NOP	NOP	NOP
15	1	1	1	1	$a$	NOP	NOP



**Fig. 3.** Look up table operations: (a) The only case where a new label is created for the pixel under analysis. (b) RCM configuration where two labels are merged.

The first step in label handling is to initialize the structures for each specific approach. In the case of the UF method, the *Parent* vector is initialized, whilst in the CL approach, *next*, *tail* and *rtable* arrays are initialized.

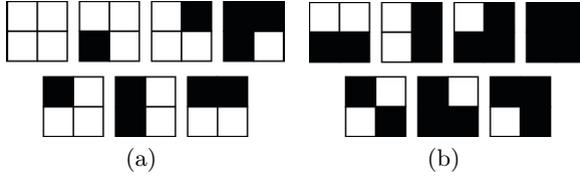
The input image is augmented by a dummy row and a dummy column composed of background pixels that are located before the first row and the first column of the image under analysis, respectively. These supplementary pixels do not interfere with the connected components actually present in the image. Conversely, they help to reduce the number of conditions to analyze in the connectivity mask enabling a faster processing in the connectivity analysis step. The scanning of the augmented image is performed in a left to right, top to bottom sequence.

Table 1 summarizes the operations needed to analyze the connectivity of each pixel. Figure 2 shows the position of each pixel (*a*, *b*, *c* and *d*) in the Reduced Connectivity Mask proposed in this paper.

We can identify four different operations on this table:

- The first case is the NOP (no operation) circumstance. We do not perform any operation when the pixel *d* (the only pixel not yet analyzed in the RCM) is a background pixel.
- The second operation occurs when only one of the pixels *a*, *b*, *c* is a foreground pixel and it has already a label assigned. Under such a condition, the label to assign to the pixel under analysis is the same label of the already assigned pixel. This operation is needed for the cases in rows 3, 5, 7, 9, 11, 13 and 15.
- The third operation needed is a label creation operation. That operation requires that the pixel under analysis be the only foreground pixel in the neighborhood defined by the connectivity mask. The operation is defined as  $m = m + 1$  because we need to increment  $m$ , the number of labels currently being used. We perform this operation only for the case in row 1.
- The fourth operation is the connectivity analysis for both labeling methods. The *Union* procedure for the Union-Find method, or the *resolve* operation for the Connection-List algorithm, is used to determine what is the correct label from the different labels already assigned to the pixels in the neighborhood. The cases for this operation are those presented in rows 6 and 7.

As said before, Table 1 specifies the operation needed for each condition that can present the 4-uple (*a*, *b*, *c*, *d*). In columns 1-4, a “0” stands for a background



**Fig. 4.** Look up table operations: (a) RCM configurations where no operation (NOP) is needed. (b) RCM configurations where the label is copied from a previously labeled pixel.

pixel, and a “1” for a foreground pixel. Fifth column of this table shows the label to assign to the  $d$  pixel and the sixth and seventh columns show the operations needed to perform in the merge operation to store the equivalence information for each method (Union-Find for the sixth column and Connection-List for the seventh column, respectively).

We call the merge procedure for each foreground pixel in the image accordingly to the labeling method. In the case of the UF-RCM approach, we call the *Find* procedure with the label associated to the pixel under analysis. The root label found by the aforementioned procedure is then assigned as the definitive label for the pixel under analysis.

In the case of the CL-RCM method, we perform instead a look up of the *rtable* array. That is  $L'(x, y) = rtable[L(x, y)]$ , with  $L'(x, y)$  being the definitive label for the pixel under analysis. After completing the second scan, all the foreground pixels are correctly labeled.

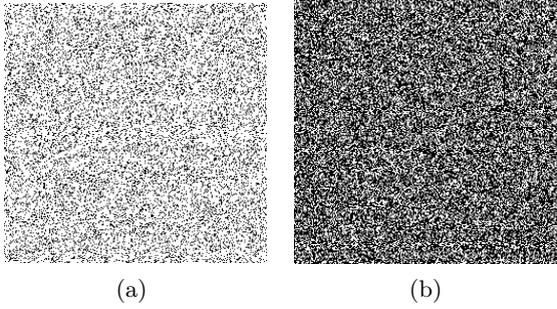
## 5 Results

### 5.1 Test Protocol

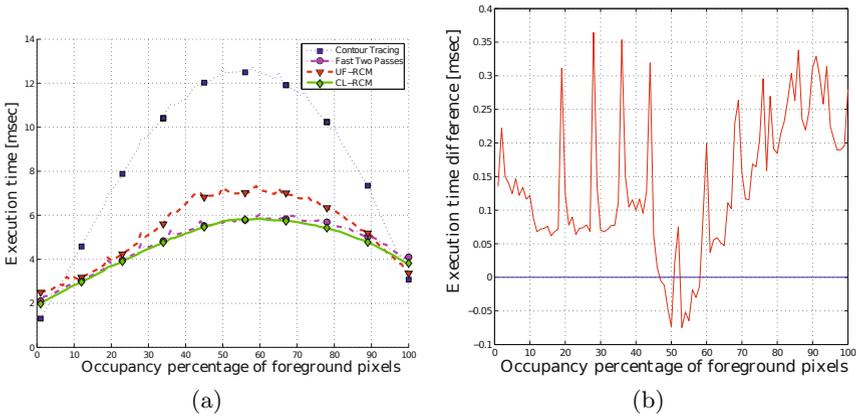
In order to evaluate our algorithm, we have experimented with a set of uniform random noise images to measure the execution time. We have compared the performance of the Contour Tracing (CT) algorithm (Chang et al. 2004), and the fast connected component labeling method (He et al. 2009) to the proposed approaches, UF-RCM and CL-RCM. The comparison of the proposed approach to these methods can provide us some insight because their particular differences.

We have generated test images using a uniform random noise across the image. The goal of this test set is to demonstrate how the algorithm performs on random input (see Figure 5). We have used eleven occupancy percentages for the foreground pixels in the images (0 to 100 percent in 10 percent steps). We have used a size of  $512 \times 512$  pixels for all the images in this data set. We have generated 40 random test images for each occupancy percentage, and we used all these images as input.

We have developed all these tests using C-language implementations for all the methods in an Intel Dual Core-based desktop computer with a processor running at 2.66 *GHz* and 4 GB RAM memory. The execution time figures are the average in 40 runs for each test.



**Fig. 5.** Two examples of uniform random noise images with varying occupation percentage of the foreground pixels: (a) 20 % and (b) 60 %



**Fig. 6.** (a) Performance results of the four methods using uniform random noise images of varying occupancy percentage of the foreground pixel as input. (b) Difference in time execution between the He’s method and CL-RCM approach. A positive value implies a better performance of the CL-RCM algorithm.

## 5.2 Results

In Figure 6 (a) we can observe the comparison of the performance exhibited by the four methods under analysis when uniform random noise images are used as inputs. In this graph, we can observe that CL-RCM method performs the best, even if it is only slightly faster than He’s method. The marginal advantage (see Figure 6 (b) for a direct comparison) of the CL-RCM method over the faster connected component labeling algorithm (He’s method) is due to the reduced size of the connectivity mask used by that approach. The improvement in time is the same that the others authors in the field, report on their works (He, Chao & Suzuki 2010b) . We can also see that the UF-RCM method exhibits a poorer performance than the CL-RCM and the fast connected component

labeling method. Nevertheless, its performance is significantly better than the one exhibited by the Contour Tracing algorithm in this test.

## 6 Conclusions

In this paper, we have proposed the use of a reduced connectivity mask (RCM) in two CCL algorithms: the Union-Find approach previously proposed by Wu and the fast connected component labeling method proposed by He *et al.* We call these approaches UF-RCM and CL-RCM respectively. The RCM consists of a  $2 \times 2$  pixel neighborhood as the scanning window. As we have seen in a set of tests involving random noise and typical benchmark pattern images, the RCM-based approaches proposed here exhibit an overall performance that is better than the others performance with respect to execution time.

We have also observed that the optimal labeling strategy strongly depends on the structure shown by the components in the input image. In the near future, we are going to explore other data structures to handle label equivalence that improve the overall method performance. We will work to automatically decide the CCL algorithm best suited to deal with a given input image, according to its content. That will also serve to improve memory usage required by our algorithm. We are also going to explore its applications in more complex tasks, as for example, motion detection clustering and optical character recognition.

**Acknowledgements.** Uriel H. Hernandez-Belmonte gratefully acknowledges Mexico's CONACYT (329356/229784) for the financial support through the scholarships given by the programs "Convocatoria de Becas Nacionales 2013 Primer Periodo" and "Convocatoria de becas CONACYT nacionales agosto - octubre 2009".

## References

- Asano, T., Tanaka, H.: In-place algorithm for connected components labeling. *Journal of Pattern Recognition Research* 5(1) (2010)
- Chang, F., Chen, C.-J.: A component-labeling algorithm using contour tracing technique. In: *International Conference on Document Analysis and Recognition*, pp. 741–745. IEEE Computer Society (2003)
- Chang, F., Chen, C.-J., Lu, C.-J.: A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding* 93(2), 206–220 (2004)
- Di Stefano, L., Bulgarelli, A.: A simple and efficient connected components labeling algorithm. In: *International Conference on Image Analysis and Processing*, pp. 322–327 (1999)
- Dillencourt, M.B., Samet, H.: A general approach to connected-component labeling for arbitrary image representations. *Journal of the ACM* 39, 253–280 (1992)
- Fiorio, C., Gustedt, J.: Two linear time Union Find strategies for image processing. *Theoretical Computer Science* 154(2), 165–181 (1996)

- Greiner, J.: A Comparison of Data-Parallel Algorithms for Connected Components. In: Proceedings Symposium on Parallel Algorithms and Architectures, pp. 16–25 (1994)
- Han, Y., Wagner, R.A.: An efficient and fast parallel-connected component algorithm. *Journal of the ACM* 37(3), 626–642 (1990)
- Hardwick, J.: Practical Parallel Divide-and-Conquer Algorithms, PhD thesis, School of Computer Science, Carnegie Mellon University (1997)
- He, L., Chao, Y., Suzuki, K.: A linear-time two-scan labeling algorithm. In: IEEE International Conference on Image Processing, pp. 241–244 (2007)
- He, L., Chao, Y., Suzuki, K.: A run-based two-scan labeling algorithm. *IEEE Transactions on Image Processing* 17(5), 749–756 (2008)
- He, L., Chao, Y., Suzuki, K.: An efficient first-scan method for label-equivalence-based labeling algorithms. *Pattern Recognition Letters* 31(1), 28–35 (2010a)
- He, L., Chao, Y., Suzuki, K.: An efficient first-scan method for label-equivalence-based labeling algorithms. *Pattern Recognition Letters* 31, 27–35 (2010b)
- He, L., Chao, Y., Suzuki, K., Wu, K.: Fast connected-component labeling. *Pattern Recognition* 42(9), 1977–1987 (2009)
- Krishnamurthy, A., Lumetta, S.S., Culler, D.E., Yelick, K.: Connected components on distributed memory machines. In: *Parallel Algorithms: Third DIMACS Implementation Challenge*, pp. 1–21. American Mathematical Society (1994)
- Samet, H.: Connected component labeling using quadtrees. *Journal of the ACM* 28(3), 487–501 (1981)
- Samet, H., Tamminen, M.: Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10(4), 579–586 (1988)
- Suzuki, K., Horiba, I., Sugie, N.: Linear-time connected-component labeling based on sequential local operations. *Computer Vision and Image Understanding* 89(1), 1–23 (2003)
- Wang, X., Davis, W.A.: Connected component labeling using modified linear quadtrees. In: *Graphics Interface 1986*, pp. 235–240 (1986)
- Wu, K., Otoo, E., Shoshani, A.: Optimizing connected component labeling algorithms. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 5747, pp. 1965–1976 (2005)
- Wu, K., Otoo, E., Suzuki, K.: Two strategies to speed up connected component labeling algorithms, Tech Report LBNL-59102 (2005)
- Wu, K., Otoo, E., Suzuki, K.: Optimizing two-pass connected-component labeling algorithms. *Pattern Analysis and Applications* 12(2), 117–135 (2008)