EFFICIENT 2×2 BLOCK-BASED CONNECTED COMPONENTS LABELING ALGORITHMS

Diêgo J. C. Santiago[1], Tsang Ing Ren[1], George D. C. Cavalcanti[1] and Tsang Ing Jyh[2]

[1]Center for Informatics, Federal University of Pernambuco
Recife, PE, Brazil – www.cin.ufpe.br/~viisar
[2]Alcatel-Lucent, Belgium
{djcs, tir, gdcc}@cin.ufpe.br, {ing-jyh.tsang}@alcatel-lucent.com

ABSTRACT

This paper presents three new efficient 2×2 block-based algorithms for connected components labeling: a two-scan which assigns provisional labels to blocks, a two-scan which assigns provisional labels to pixels and a one-and-a-half-scan which assigns provisional labels to blocks. A new stripe image representation is designed in order to perform the second pass only through the blocks containing some foreground pixel. We also improved the existing 2×2 block-based algorithms by utilizing information of a pixel during a transition in the mask, which allows checking of four neighbor pixels in the mask at most. Thus, the average number of checking operations needed to inspect the neighbor pixels in the first scan is reduced from 1.459 to 1.156, an improvement of 21%. We conducted experiments using synthetic and real images to evaluate the performance of the proposed methods compared to the existing methods. The proposed block-based one-and-a-half-scan algorithm presents the best performance in the real images dataset, which is composed of 1290 documents. Our block-based two-scan algorithm which assigns provisional labels to pixels showed to be the fastest in the synthetic dataset, especially in high density images.

Index Terms— Connected components labeling, block-based, one-and-a-half-scan, image processing, image analysis

1. INTRODUCTION

Object detection and classification is a common problem in computer vision. In binary images analysis, objects are extracted by means of Connected Components Labeling (CCL), which distinguish objects in an image by assigning a unique label for each connected component. CCL can also be used in filtering algorithms [20]. Moreover, physicists, chemists and materials scientists use CCL as a tool for simulation in the field of percolation theory [15,16]. They concentrate their efforts in trying to apply it to larger and larger lattices [15]. In all applications, especially in real time applications, speeding up the algorithm is fundamental [1,2,17,18]. Therefore, the main goal of CCL methods in the literature has been to reduce its time without lost of accuracy. These methods are also interesting as they provide some hints and new methodologies.

Many CCL algorithms have been proposed. We divide them into four classes depending on the way they check the neighborhood in the first scan in order to assign a provisional label and to determine label equivalences: pixel-based [3,7,9,10,22,23,24], run-based [8,11,14], block-based [5,6,12,13,19,21] and stripe-based [25]. According to the number of times of scanning an image for labeling, there are multi-scan [7,22], two-scan [5,6,8,9,10,12,13,19,21,23,24,25], one-and-a-half-scan [11,19] and one-scan [3,14] algorithms.

Modern algorithms derive from historical ones and try improvements by replacing some parts by a more efficient one. Santiago *et al.* [19] showed a considerate improvement in real case images by turning pixel-based and block-based two-scan algorithms into one-and-a-half-scan ones without extra checking operations in the first pass. In general, for ordinary computer architectures and pixel-based images representation, before this improvement, the fastest labeling algorithms belonged to the 2×2 block-based connected components labeling class. However, for these algorithms, it is not possible to perform one-and-a-half-scan using run data without extra comparisons, analogously to He *et al.* [11] and Santiago *et al.* [19]. Hence, we design a new image representation based on the stripe definition of Zhao *et al.* [25]. A stripe is a part of a connected component, which is in the even row and the following odd row of the image.

This paper presents three new efficient 2×2 block-based algorithms for connected components labeling: a two-scan that assigns provisional labels to blocks, a two-scan that assigns labels to pixels and a one-and-a-half-scan that assigns labels to blocks. The algorithms improve the existing 2×2 block-based algorithms by utilizing information of a pixel during a transition in the mask, which allows checking of four neighbor pixels in the mask at most. Whereas the other one-and-a-half algorithms [11,19] use the run data information in order to perform the second pass only through the foreground pixels, a new stripe image representation is designed in order to pass only through the blocks containing any foreground pixel.

The rest of this paper is organized in the following way. The next section reviews the block-based algorithms. Section 3 introduces our proposed algorithms. We show experimental results and analysis in Section 4. Finally, we present some conclusions in Section 5.

2. RELATED WORK

In the following discussion, we adopt the eight-connectivity [4]. All algorithms yield to the same result: the number of labeled connected components is the same and consecutive labels can be achieved with a common enumeration procedure [6].

The first block-based algorithm was proposed by Grana *et al.* [5]. In the first pass, their algorithm scan an image by moving over an extended mask of five 2×2 blocks as shown in Figure 1. Only the pixels *h*, *i*, *j*, *k*, *l*, *n* and *r* were considered to be checked in their algorithm. Grana *et al.* in [6] improved their previous work by using decision tree and by considering eight-connection between blocks of the mask. Thus, only the pixels *a*, *f*, *l* and *q* were ignored. Since all the pixels of a 2 × 2 square are connected to each other, their algorithm labels each block rather than each pixel. Therefore, in the second pass, it needs to access the label image and again the binary image in order to check which pixels in the block require their label to be set. However, if the provisional label is 0, i.e., all the pixels of the block are background, so it does not need to access the binary image again.

Sutheebanjard *et al.* [21] proposed a new mask by eliminating the block *C* of the original 2×2 mask. Their proposal was intended to improve performance of high-density images by creating a more balanced decision tree. In random images, their algorithm performs faster than Grana *et al.* [6], except in low-density images. The pixels *a, d* and *q* in the mask are ignored, once they are not connected to the other blocks.

Santiago *et al.* [19] were the first to propose one-and-a-half scan algorithms for pixel-based and block-based masks. A one-and-a-half scan algorithm was proposed by He *et al.* [11], but only for run-based mask. A new 2×1 block-based mask was designed so as to enable the design of block-based one-and-a-half algorithms. Also, a new 1×2 block-based mask was proposed by He *et al.* [12]. Similarly to previous He *et al.* algorithm [10], both of these methods [12,19] divide the blocks being labeled into two groups: blocks following background pixels and blocks following foreground pixels. This operation is done without any extra work by utilizing information of a pixel during a transition in the mask. Recently, He *et al.* [13] proposed to use as much as information as possible. In their method, any pixel checked in the mask when processing the current block is not checked again when the next block is processed.

All modern algorithms exploit a decision tree so as to minimize the number of neighboring pixels to be visited when it evaluates the label of the current pixel. In an eight-connectivity, among all the neighboring pixels, only one of them is sufficient to determine the label of the current pixel [9]. Thus, we need to resolve the label equivalences only when two foreground pixels with different provisional labels become connected by the current pixel, block, run or stripe [9].

3. PROPOSED ALGORITHMS

For an N×M-sized image, we use *L*(*X*) to denote the label pixel on the block *X*; *last* is used to denote the value of the last label created and *lab* the value of the last label assigned. As in most labeling algorithms, we assume that all pixels on the border of images are background pixels.

3.1. New block-based algorithm

In the first scan, the proposed algorithm scans an image by moving a mask composed of four 2×2 blocks, the mask illustrated in Figure 1 without the block *C*, finding the provisional label of the entire block at the same time. Nevertheless, only the pixels *h*, *i*, *j* and *r* are considered to be checked in our algorithm, the other pixels are ignored. The pixel *n*, checked in the previous configuration, is known without extra work. Thus, we divide the current block being labeled into two groups: blocks following background pixels, where the pixel *n* is background, and blocks following foreground pixels, where the pixel *n* is foreground.

The second scan is the same as the existing 2×2 block-based algorithms whether the first scan assigns a provisional label to each block (EBTSB means Efficient Block-based Two-Scan which assigns provisional labels to Blocks) or it is the same as the other algorithms when assigns a provisional label to each pixel in the first scan (EBTSP means Efficient BTS which assigns provisional labels to Pixels). In the later case, the provisional labels of the pixels that are not considered in the mask ("either" in Figure 2) are assigned using Lookup Table, without extra comparison, like it is in the conventional second pass.

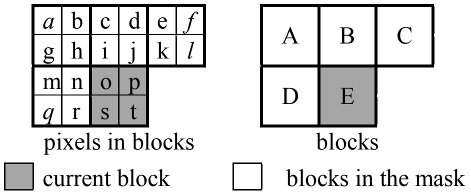

Figure 1. The 2×2 mask proposed by Grana *et al.* [5].

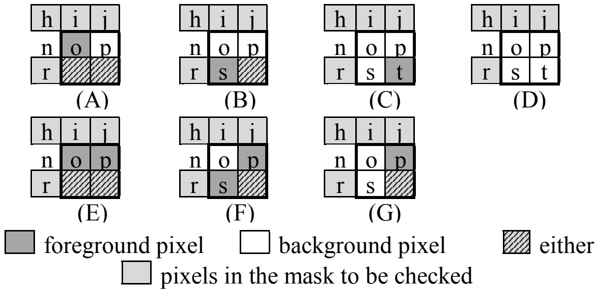

Figure 2. Seven possible cases for the mask used in the proposed algorithms regarding the four pixels in the block.

There are seven cases in our mask to be considered regarding the four pixels of the current block being labeled, as shown in Figure 2. For the neighbor pixels in the mask, there are sixteen cases, as illustrated in Figure 3.

In Cases (A) and (E), the current block *E* can be connected to the blocks *A*, *B* and *D*. In order to process these cases, the designed procedures are summarized in Table 1. For blocks following background pixels, a new provisional label is assigned in Case (1). The block *E* is labeled with the last label assigned *lab* in Cases (9-16). In the other cases, any label in the mask is assigned. In Cases (6) and (10-16), we need to consider label equivalences, which are done by the operation *resolve*. In Case (14), equivalences between the labels of the blocks *A*, *B* and *D* must be considered. For block following foreground pixel, the current block is labeled with the last label assigned *lab*. Label equivalences must be considered in Cases (3), (5-7), (11) and (13-15).

In Case (F), the current block can be connected to the blocks *B* and *D*. The procedure designed in order to process this case are summarized in Table 2. For blocks following background pixels, a new provisional label is assigned in Cases (1) and (2). The block *E* is labeled with the last label assigned *lab* in Cases (9-16) and with the label of the block *B* in Cases (3-8). In Cases (11-16), we need to consider label equivalences. For blocks following foreground pixels, the current block is labeled with the last label assigned *lab*. Label equivalences must be considered in Cases (3), (5-7), (11) and (13-15), the same as Cases (A) and (E).

The Cases (B), (C), (D) and (G) are easier. In Case (B), the current block can be connected only to the block *D*. For blocks following background pixels, a new provisional label is assigned

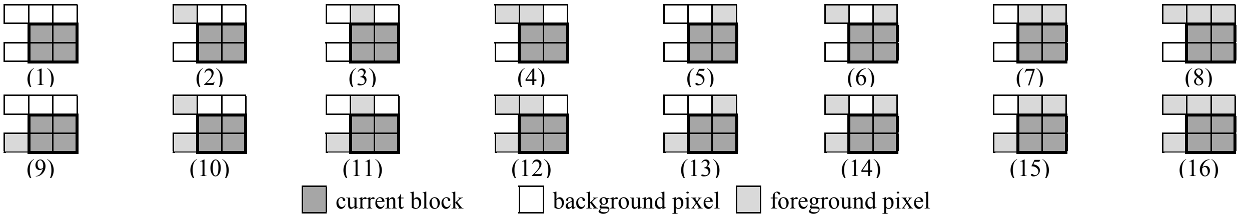

Figure 3. Sixteen possible cases regarding the neighbor pixels in our mask.

whether the pixel r is background or the block E is labeled with the last label assigned lab otherwise. For blocks following foreground pixels, the block is labeled with the last label assigned lab. Nonetheless, label equivalences must be considered when the pixel i is foreground and h is background. The same cases need to be considered in Cases (C), (D) and (G) for blocks following foreground pixels. In Case (C), the current block cannot be connected to any other block and it is labeled with a new label. In Case (D), the current block is made of only background pixels and the label of this block is 0. In Case (G), the current block can be connected only to block B. For blocks following background pixels, a new provisional label is assigned whether the pixels i and j are both background or the block is labeled with the label of the block B otherwise. For blocks following foreground pixels, the block E is labeled with the last label assigned lab if the pixel i is foreground or the label of the block B if the pixel j is foreground or a new label otherwise.

3.2. One-and-a-half-scan algorithm

As opposite to most types of labeling algorithms, the existing 2x2 block-based algorithms cannot store run data information in the first scan without extra work, once not all pixel are checked in the first scan. For instance, the pixel t is only checked if the pixels o, p and s are background. Hence, we use a new image representation to store the start of stripes. The end of the current stripe is not stored like in the other one-and-a-half algorithms [11,19]; it is found in the half scan when the next stripe starts or when the provisional label of the current block is 0.

For blocks following background pixels when the pixel r is background, in all cases, except Case (D), a stripe starts in the current block. For blocks following background pixels when the pixel r is foreground and for block following foreground pixel, a stripe starts in the current block in Cases (C) and (G). We do not need of extra work because the pixel r is always checked when block following a background pixel for all the cases, except Cases (C), (D) and (G), although in Cases (C) and (G), a stripe always starts regardless of the transition or the pixel r.

4. EXPERIMENTAL RESULTS

The experiment was performed on an Intel Core i7-4770, 3.4 GHz, RAM 8 GB, 8 cores, using a single core for the processing. All experimental results presented in this section were obtained by averaging the execution time for 200 runs, 100 runs on Ubuntu 14.04 (50 runs using the compiler GCC 4.9.2 -O2 and 50 using the compiler Clang 3.5 -O2) and 100 runs on Windows 8 (50 runs using the compiler MinGW GCC 4.9.2 -O2 and 50 using the Microsoft Visual Studio 2012 compiler /O2).

All algorithms used for the comparison, which were implemented in C, were block-based algorithms: six two-scan (2×1 BTS [19], 1×2 HCS [12], 2×2 BBDT [6], 2×2 sBBDT [21], 2×2 eBTSB and 2×2 eBTSP from Section 3.1) and two one-and-a-half-scan (2×1 BOS [19] and 2×2 eBOS from Section 3.2) algorithms. The programs of the BBDT (Block-based with Decision Tree) and sBBDT (squared Block-Based with Decision Tree) was improved and adapted from the OpenCV compliant version, provided by the authors, so as to consider all the pixels on the border of the image are background. In tests, the border has size two. The source code of all algorithms is available for download in http://cin.ufpe.br/~djcs/labeling.

In contrast to Santiago *et al.* [19], Grana *et al.* [6] and Sutheebanjard *et al.* [21], He *et al.* [13] do not provide the source code, which, claimed by the author, has about 600 lines. Their algorithm has many rules for a reader to verify and reproduce. Therefore, we do not include this algorithm in our comparison, but the previous version of this algorithm [12].

In order to evaluate the performance of the proposed first and second image scan algorithm and then to avoid the effect of the equivalences resolution operation, the equivalences resolution of all algorithms followed the Union-Find technique as presented by He *et al.* in [8].

Table 1. Operations performed for Cases (A) and (E).

Case	following background pixel	following foreground pixel
(1)	$lab=last$, $last = last + 1$	-
(2)	$lab=L(A)$	-
(3)	$lab=L(B)$	$resolve(lab,L(B))$
(4)	$lab=L(B)$	-
(5)	$lab=L(B)$	$resolve(lab,L(B))$
(6)	$lab=L(B)$, $resolve(lab,L(A))$	$resolve(lab,L(B))$
(7)	$lab=L(B)$	$resolve(lab,L(B))$
(8)	$lab=L(B)$	-
(9)	-	-
(10)	$resolve(lab,L(A))$	-
(11)	$resolve(lab,L(B))$	$resolve(lab,L(B))$
(12)	$resolve(lab,L(B))$	-
(13)	$resolve(lab,L(B))$	$resolve(lab,L(B))$
(14)	$resolve(lab,L(B),L(A))$	$resolve(lab,L(B))$
(15)	$resolve(lab,L(B))$	$resolve(lab,L(B))$
(16)	$resolve(lab,L(B))$	-

Table 2. Operations performed for Case (F).

Case	following background pixel	following foreground pixel
(1)	$lab=last$, $last = last + 1$	-
(2)	$lab=last$, $last = last + 1$	-
(3)	$lab=L(B)$	$resolve(lab,L(B))$
(4)	$lab=L(B)$	-
(5)	$lab=L(B)$	$resolve(lab,L(B))$
(6)	$lab=L(B)$	$resolve(lab,L(B))$
(7)	$lab=L(B)$	$resolve(lab,L(B))$
(8)	$lab=L(B)$	-
(9)	-	-
(10)	-	-
(11)	$resolve(lab,L(B))$	$resolve(lab,L(B))$
(12)	$resolve(lab,L(B))$	-
(13)	$resolve(lab,L(B))$	$resolve(lab,L(B))$
(14)	$resolve(lab,L(B))$	$resolve(lab,L(B))$
(15)	$resolve(lab,L(B))$	$resolve(lab,L(B))$
(16)	$resolve(lab,L(B))$	-

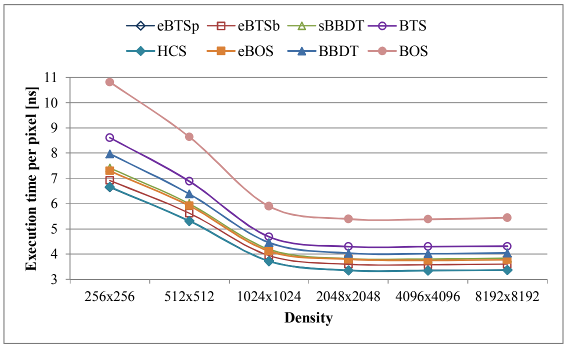

Figure 4. Performance of the algorithms scaling the size.

4.1. Synthetic dataset

Analogously to most works [6,8,9,10,11,12,13,19,21,24], we used a synthetic dataset of black and white random noise square images. The synthetic dataset is composed by 114 images, with nineteen different densities, from 0.05 to 0.95, and six different images sizes: 256x256, 512x512, 1024x1024, 2048x2048, 4096x4096 and 8192x8192. This dataset is the same as in the previous work [19].

The experimental results demonstrated that our block-based two-scan algorithm eBTSp, which assigns provisional labels to pixels, is the fastest for all sizes of image, as shown in Figure 4. Moreover, the difference of time between our two-scan and our one-and-a-half-scan algorithms is not as big as that between the two-scan and one-and-a-half-scan of the previous work [19], since the number of stripes is much smaller than the number of runs and we store only the start of stripes. We also observed that our block-based two-scan algorithm which assigns labels to pixels eBTSp had only a slightly improvement with respect to HCS [12].

We also tested 8192x8192 images varying the density of the foreground pixels, as shown in Figure 5. The experimental results show that our block-based two-scan algorithm eBTSp, which assigns provisional label to pixel, is the fastest in all densities, whereas our block-based two-scan algorithm eBTSb, which assigns provisional label to blocks, is only slower than eBTSp and HCS in all densities as well as BTS in low densities, from 0.05 to 0.20.

4.2. Document image dataset

We tested 1290 images from Tobacco800 Document Image Database, the same database of previous work [19]. However, each image was replicated forming a 3x3 grid of images, disconnected each other by the image border, in order to improve the measurement accuracy. The experimental results demonstrated that our block-based one-and-a-half-scan consumes the lowest computation time, as shown in Table 3, and reaches an improvement of about 12% on average with respect of the state-of-the-art method and about 18% on average with respect of the previous state-of-the-art method.

Efficiency is dependent on the nature of the input image. The performance characteristics are different for random images. The reason is because the real case images typically contain well-shaped connected components, whereas the random images contain irregular components. This is made clear when it is observed that all of our algorithms outperform HCS by more than 7% in real case images, whereas our best one only slightly speed it up in random images. A reason for these results is that the Cases (A) and (B) are more common for this type of images. In these cases, our algorithm does not use the pixel *t* in comparisons, whereas HCS does.

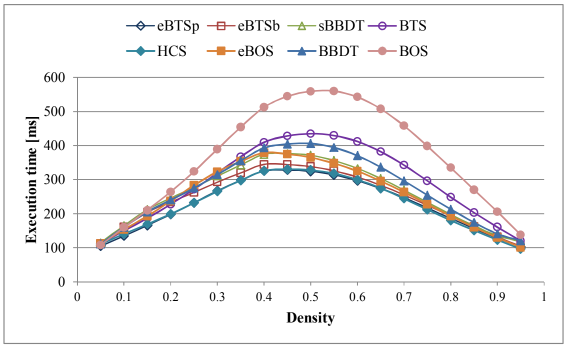

Figure 5. Performance of the algorithms varying the density.

Table 3. Execution time [ms] for Document Image Dataset.

	eBOS	BOS	BBDT	eBTSb	sBBDT	eBTSp	HCS	BTS
min	**6.0**	7.0	8.4	8.7	10.1	14.4	15.7	15.4
mean	**23.8**	27.3	28.9	29.5	32.7	39.9	43.0	44.3
max	**116**	142	117	123	128	117	121	142
std	16.0	18.3	18.4	18.7	20.7	24.6	26.5	27.5

4.3. Analysis

The time complexity of the algorithms is determined mainly by the number per pixel of comparisons, *resolve* operations, and memory accesses, to both label and binary image. Table 4 shows the metrics that are measured from the synthetic image with size 8192x8192 and density 0.50. The number of memory accesses of one-and-a-half-algorithms depends on the number of runs and stripes. Our algorithms perform the lowest number of comparisons and accesses to image. We improve the number of comparisons of sBBDT first scan method from 1.459 to 1.156, whereas, for eBTSb, the number of comparisons for the second scan maintains 0.250. Nevertheless, our algorithms and sBBDT perform the second most number of merge operations, fewer operations than HCS.

Table 4. Analysis of the time complexity of the algorithms.

	Comparisons	Label image accesses	Binary image accesses	Total accesses	*resolve* operations
BBDT	1.842	1.681	2.530	4.211	0.028
sBBDT	1.709	1.685	2.397	4.082	0.034
BTS	1.781	3.281	1.000	4.281	**0.024**
HCS	1.250	3.313	1.000	4.313	0.036
eBTSp	**1.156**	3.172	1.000	4.172	0.034
eBTSb	1.406	1.562	2.094	**3.656**	0.034

5. CONCLUSION

In this paper, we proposed three algorithms for connected components labeling. A new stripe image representation was designed in order to perform the second pass only through the blocks containing any foreground pixel. The number of comparisons and memory accesses of the existing 2×2 block-based algorithms also was reduced by utilizing information of a pixel during a transition in the mask. The experimental results demonstrated that our block-based two-scan algorithms are more efficient for random images, whereas our block-based one-and-a-half-scan algorithm is more efficient for document analysis. We expect that using more information about the transition of configurations of pixels in the mask can accelerate the algorithm and, hence, this is a topic to be investigated in future works.

6. REFERENCES

[1] R. G. Abbot and L. R. Williams, "Multiple target tracking with lazy background subtraction and connected components analysis," *Machine Vision and Applications*, vol. 20, no. 2, pp. 93-101, Nov. 2009.

[2] A. Bevilacqua, A. Lanza, G. Baccarani and R. Rovatti, "A single-scan algorithm for connected components labeling in a traffic monitoring application," in *Scandinavian Conference on Image Analysis*, 2003, pp. 677-684.

[3] F. Chang, C. J. Chen, and C. J. Lu, "A linear-time component-labeling algorithm using contour tracing technique," *Comput Vis Image Underst*, vol. 93, n. 2, pp. 206-220, 2004.

[4] R. C. Gonzalez, and R.E. Woods, *Digital Image Processing*, Prentice-Hall, New Jersey, 2002.

[5] C. Grana, D. Borghesani, and R. Cucchiara, "Fast block based connected components labeling," in *IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 4061-4064.

[6] C. Grana, D. Borghesani, and R. Cucchiara, "Optimized block-based connected components labeling with decision trees," *IEEE Transactions on Image Processing*, vol. 19, n. 6, pp. 1596-1609, 2010.

[7] R. M. Haralick, "Some neighborhood operations," *Real Time Parallel Computing: Image Analysis*, Plenum Press, New York, pp. 11-35, 1981.

[8] L. He, Y. Chao, and K. Suzuki, "A run-based two-scan labeling algorithm," *IEEE Transactions on Image Processing*, vol. 17, n. 5, pp. 749–756, 2008.

[9] L. He, Y. Chao, K. Suzuki and K. Wu, "Fast connected-component labeling," *Pattern Recognition*, vol. 42, pp. 1977-1987, 2008.

[10] L. He, Y. Chao, and K. Suzuki, "An efficient first-scan method for label-equivalence-based labeling algorithms," *Pattern Recognition Letters*, vol. 31, n. 5, 2010.

[11] L. He, Y. Chao, and K. Suzuki, "A run-based one-and-a-half-scan connected-component labeling algorithm," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 24, n. 4, pp. 557-579, 2011.

[12] L. He, Y. Chao, and K. Suzuki, "A new two-scan algorithm for labeling connected components in binary images," in *Proceedings of the World Congress on Engineering*, 2012, vol. 2, pp. 1141-1146.

[13] L. He, X. Zhao, Y. Chao, and K. Suzuki, "Configuration-Transition-Based Connected-Component Labeling," *IEEE Transactions on Image Processing*, vol. 23, n. 2, pp. 943-951, 2014.

[14] J. Martín-Herrero, "Hybrid object labeling in digital images," *Machine, Vision and Applications*, vol. 18, pp. 1-15, 2007.

[15] J. Hoshen, M. Berry and K. Minser, "Percolation and cluster structure parameters: The enhanced Hoshen-Kopelman algorithm," *Physical Review E*, vol. 56, no. 2, pp. 1455-1460, Aug. 1997.

[16] J. Hoshen, "On the application of the enhanced Hoshen-Kopelman algorithm for image analysis," *Pattern Recognition Letters*, vol. 19, pp.575-584, 1998.

[17] W. C. Hu, C. Y. Yang and D. Y. Huang, "Robust real-time ship detection and tracking for visual surveillance of cage aquaculture," *Journal of Visual Communication and Image Representation,* vol. 22, no. 6, pp. 543-556, Aug. 2011.

[18] T. B. Nguyen, and S. T. Chung, "An improved real-time blob detection for visual surveillance," in *International Congress on Image and Signal Processing*, 2009, pp. 1-5.

[19] D. J. C. Santiago, T. I. Ren, G. D. Cavalcanti, and T. I. Jyh, "Fast block-based algorithms for connected components labeling," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 2084-2088.

[20] P. Y. Simard and H. S. Malvar, "An efficient binary image activity detector based on connected components," in *International Conference on Acoustics, Speech, and Signam Processing (ICASSP)*, 2004, pp. 229-232.

[21] P. Sutheebanjard and W. Premchaiswadi, "Efficient Scan Mask Techniques for Connected Components Labeling Algorithm," *EURASIP Journal on Image and Video Processing*, 2011:14.

[22] K. Suzuki, I. Horiba and N. Sugie, "Linear-time connected-component labeling based on sequential local operations," *Comput Vis Image Underst.*, vol. 89, n. 1, pp.1-23, 2003.

[23] A. Rosenfeld and J.L. Pfaltz, "Sequential Operations in Digital Picture Processing," *Journal of the ACM*, vol. 13, n. 4, pp. 471-494, 1966.

[24] K. Wu, E. Otto and K. Suzuki, "Optimizing two-pass connected-component labeling algorithms," *Pattern Analysis and Applications*, vol. 12, pp. 117-135, 2008.

[25] H. L. Zhao, Y. B. Fan, T. X. Zhang and H. S. Sang, "Stripe-based connected components labelling," *Electronics Letters*, vol. 46, n. 21, pp. 1434-1436, 2010.