LETTER

A New First-Scan Method for Two-Scan Labeling Algorithms

Lifeng HE[†a)], Yuyan CHAO[††], *Nonmembers*, *and* Kenji SUZUKI[†††], *Member*

SUMMARY This paper proposes a new first-scan method for two-scan labeling algorithms. In the first scan, our proposed method first scans every fourth image line, and processes the scan line and its two neighbor lines. Then, it processes the remaining lines from top to bottom one by one. Our method decreases the average number of times that must be checked to process a foreground pixel will; thus, the efficiency of labeling can be improved.

key words: *labeling algorithm, connected component, first scan, pattern recognition*

1. Introduction

Labeling of connected components in a binary image is one of the most fundamental operations in pattern analysis, pattern recognition, computer (robot) vision, and machine intelligence [6], [7]. Especially in real-time applications such as traffic-jam detection, automated surveillance, and target tracking, faster labeling algorithms are always desired.

Many algorithms have been proposed for addressing this issue, because improving the efficiency of labeling is critical in many applications. For ordinary computer architectures and 2D images, there are mainly two types of labeling algorithms:

(1) Raster-scan algorithms. These algorithms process an image in the raster-scan way. There are multi-scan algorithms [17] and two-scan algorithms [7]–[12], and one-and-a-half scan algorithm [13];
(2) Label propagation algorithms. These algorithms access an image in an irregular way. There are run-based algorithms [2], [16] and contour-tracing algorithms [3], [14].

According to experimental results on various types of images, the two-scan labeling algorithm proposed in Ref. [10], an improvement of the algorithm proposed in Ref. [8], is the most efficient one, and has been used for various applications [1], [4], [5]. For convenience, we denote this algorithm as *MECTSL* (Most Efficient Conventional Two-Scan Labeling) algorithm.

Similar to other two-scan labeling algorithms, in the first scan, the MECTSL algorithm processes an image line by line. It assigns provisional labels to the foreground pixels in the scan line, and resolves the connectivities of these foreground pixels and those on the above neighbor line.

In fact, the connectivities of the foreground pixels on the scan line and those on the below neighbor line can also be found easily when we process the foreground pixels on the scan line. Comparing to resolving these connectivities later when scanning the next line, where we need to check the connectivies again, it would be more efficient to do that at the same time as processing the scan line.

This paper presents a totally new first-scan method for two-scan labeling algorithms. We first scan image lines every four lines. For each scan line, we assign provisional labels to the foreground pixels on the scan line and its two neighbor lines, and resolve the label equivalences among those labels. Then, we scan the unprocessed remaining lines one by one from top to bottom. For each line, we assign provisional labels to the foreground pixels on the scan line and resolve label equivalences among these labels and those assigned to the foreground pixels on its two neighbor lines. Experimental results showed that the efficiency of our first-scan method is superior to that of the MECTSL algorithm.

2. Review of the MECTSL Algorithm

For an $N \times M$ binary image, we use $b(x, y)$ to denote the pixel at (x, y) in the image, where $1 \leq x \leq N$, $1 \leq y \leq M$, and also its value. All pixels in the edge of an image are considered to be background pixels. Because 4-connectivity is a subcase of 8-connectivity, we will only consider 8-connectivity in this paper.

Similar to other two-scan labeling algorithms, the MECTSL algorithm completes labeling in two scans by four processes:

(1) Provisional label assignment, i.e., assigning a provisional label to each foreground pixel;
(2) Equivalent label finding and recording, i.e., finding all provisional labels assigned to foreground pixels that are connected and using some data structures to record them as equivalent labels;

Manuscript received March 8, 2012.
Manuscript revised April 6, 2012.

[†]The author is with the College of Electrical & Information Engineering, Shaanxi University of Science and Technology, Shaanxi 710021, China and the Graduate School of Information Science and Technology, Aichi Prefectural University, Nagakute-shi, 480–1198 Japan.
[††]The author is with Graduate School of Environment Management, Nagoya Sangyo University, Owariasahi-shi, 488–8711 Japan.
[†††]The author is with Department of Radiology, Division of the BiologicalSciences, The University of Chicago, Chicago, IL 60637, USA.
a) E-mail: helifeng@ist.aichi-pu.ac.jp

DOI: 10.1587/transinf.E95.D.2142

(3) Label-equivalence resolving, i.e., finding a representative label for all equivalent provisional labels;
(4) Label replacement, i.e., replacing each provisional label by its representative label.

The first process is completed in the first scan, the second and the third processes are completed during the first scan and/or between the first scan and the second scan, and the fourth process is completed in the second scan.

The MECTSL algorithm uses equivalent label sets and a representative label table to record equivalent labels and resolve the label equivalences. For convenience, an equivalent label set with the representative label u is denoted as $S(u)$, and that the representative label of a provisional label l is t is denoted as $t = T[l]$.

In the first scan, this algorithm uses the mask shown in Fig. 1 (a), which consists of three scanned (processed) neighbors of the current foreground pixels, to assign provisional labels to foreground pixels, and to record and resolve label equivalences. At any moment in the first scan, all equivalent provisional labels are combined in an equivalent label set with the same representative label.

For the case where the current foreground pixel follows a background pixel (Fig. 1 (b)), if there is nolabel (foreground pixel) in the mask, this means that the current foreground pixel does not connect with any scanned foreground pixel, and the current foreground pixel belongs to a new connected component at this point. The algorithm assigns a new provisional label m to the current foreground pixel, which is initialized to 1, and establishes the equivalent label set $S(m) = \{m\}$; it sets the representative label table as $T[m] = m$, and $m = m+1$ for later processing. Otherwise, i.e., if there are some foreground pixels in the mask, all of such foreground pixels and the current foreground pixel belong to the same connected component. Therefore, the current foreground pixel can be assigned any of the labels in the mask.

On the other hand, for the case where the current foreground pixel follows another foreground pixel (Fig. 1 (c)), the current foreground pixel can be assigned the same label of that foreground pixel.

In both cases, if there are provisional labels belonging to different equivalent label sets, all provisional labels in those sets are equivalent labels, and therefore should be combined. Suppose that u and v are equivalent labels that belong to $S(T[u])$ and $S(T[v])$, respectively. If $T[u] = T[v]$, the two label u and v belong to the same equivalent label set already, thus, nothing needs to be done. Otherwise, without loss of generality, suppose that $T[u] < T[v]$, i.e., $T[u]$ is the smallest label in the two equivalent label sets, then the combination of the two equivalent label sets can be completed by the following operations:

$$S(T[u]) = S(T[u]) \cup S(T[v]);$$
$$(\forall s \in S(T[v]))(T[s] = T[u]).$$

As soon as the first scan is finished, all equivalent labels of each connected component have been combined into an equivalent label set with a unique representative label. In the second scan, by replacement of each provisional label with its representative label, all foreground pixels of each connected component will be assigned a unique label.

3. Outline of Our Proposed First-Scan Method

Our first-scan method consists of two parts: Scan 1-A and Scan 1-B.

In Scan 1-A, from line 3 shown in Fig. 2, it uses the mask shown in Fig. 3 to scan image lines every four other lines, i.e., in the order of line 3, line 7, line 11, ... (i.e., the black lines in Fig. 3). For each pixel on the scan line being processed, it assigns provisional labels to the foreground pixels on the line and its two neighbor lines (i.e., the gray lines in Fig. 2), and resolves the label equivalences among these labels. By Scan 1-A, all foreground pixels in each area consisting of black and gray lines in Fig. 2 will be assigned provisional labels, among which label equivalences have been resolved.

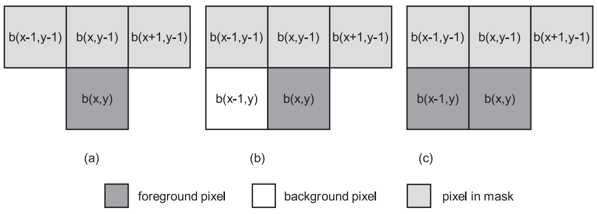

Fig. 1 Mask used in the MECTSL algorithm.

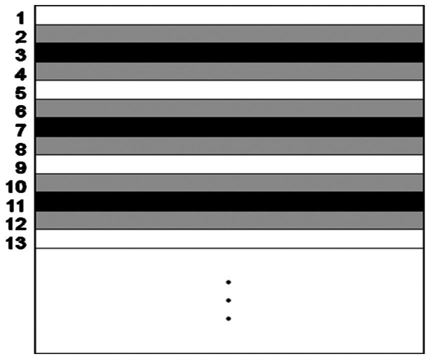

Fig. 2 Scanning method of the first scan in our proposed method.

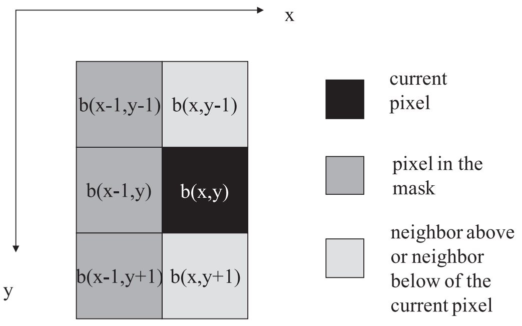

Fig. 3 Mask used in Scan1-A.

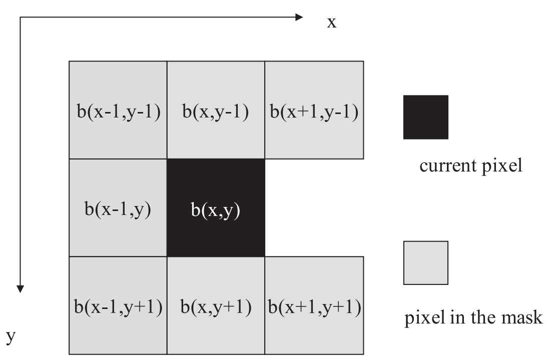

Fig. 4 Mask used in Scan1-B.

After Scan 1-A, Scan 1-B uses the mask shown in Fig. 4 to scan the lines unprocessed in the Scan 1-A (i.e., the white lines in Fig. 2) in the order of line 5, line 9, line 13, ... For each scan line, it assigns provisional labels to the foreground pixels on the line, and resolves the label equivalences among these labels and those on its two neighboring lines.

When Scan 1-B is finished, all foreground pixels in the given image will be assigned provisional labels, and all label equivalences will be resolved, i.e., all equivalent labels will be combined into an equivalent label set with a unique representative label.

In the second scan, similar to other two-scan labeling algorithms, by replacing the provisional label of each foreground pixel with its representative label, we can complete the whole labeling process.

Notice that, in our algorithm, the label equivalences are recorded and resolved in exactly the same way introduce as in the MECTSL algorithm.

4. Comparative Evaluation

We implemented the MECTSL algorithm and our algorithm with the C language on a PC-based workstation (Intel Pentium D 3.0 GHz + 3.0 GHz CPUs, 2 GB Memory, Mandriva Linux OS). Because our method is a new first-scan method (as we described above, the second scan of our method is exactly the same with the MECTSL method), we will compare the performances of the two methods only on the first scan. All data in this section were obtained by averaging of the execution time for 10,000 runs with a single core.

Noise images consist of forty one 512×512-sized noise images were generated by thresholding of the images containing uniform random noise with 41 different threshold values from 0 to 1000 in steps of 25. The densities[†] of these images are from 0.04 to 0.99.

On the other hand, 50 natural images, including landscape, aerial, fingerprint, portrait, still-life, snapshot, and text images, obtained from the Standard Image Database (SIDBA) developed by the University of Tokyo[††] and the image database of the University of Southern California[†††], were used for realistic testing of labeling algorithms. In addition, seven texture images, which were downloaded from the Columbia-Utrecht Reflectance and Texture Database[††††], and 25 medical images obtained from a medical image database of The University of Chicago were used for testing. All of these images were 512×512 pixels in size, and they were transformed into binary images by means of Otsu's threshold selection method [15]. The densities of the transformed images are from 0.05 to 0.61.

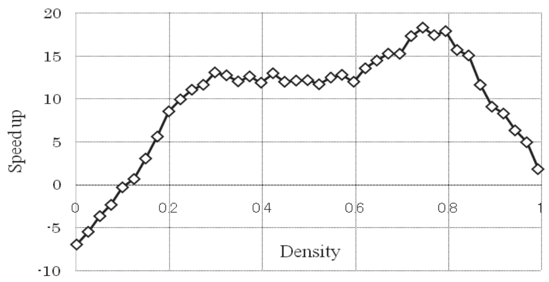

Fig. 5 Speed-up (%) of our method compared to the MECTSL method on the 512 × 512 noise images.

Figure 5 shows the speed-up of our algorithm compared to the MECTSL algorithm on the 512 × 512 noise images, where the vertical axis is defined as $(t_1 - t_2)/t_1 \times 100\,(\%)$, where t_1 is the execution time of the MECTSL algorithm and t_2 is that of our proposed method.

From Fig. 5, we can find that our proposed algorithm is relatively efficient for images with densities from 0.2 to 0.9 (the largest speed-up ratio happens for the noise image with density about 0.75). Because the foreground pixels in such images have complicated connectivity, it shows that our algorithm is efficient for images with complicated connectivity.

On the other hand, because our algorithm processes an image piecewise, it is not as efficient as the MECTSL algorithm, which does that successively. Therefore, as shown in Fig. 5, our algorithm is not as efficient as the MECTSL algorithm for the noise images whose densities are lower than 0.1, where the advantage of our algorithm for resolving connectivity cannot be exerted.

For high-density noise images, because our method processes an image every four lines in Scan1-A, the number of provisional labels assigned by our algorithm is much larger than that assigned by the MECTSL algorithm. For example, for the highest density noise image used in our test, the number of provisional labels assigned by our algorithm and that assigned by the MECTSL algorithm are 129 and 2, respectively. For each provisional label, we need to apply some operations to establish a new equivalent label set and initialize the representative label table. Moreover, because the connectivity of the foreground pixels in this case is simple, the advantage of our method for resolving connectivity

[†]The density of an image is defined as Nf/N, where Nf is the number of foreground pixels in the image and N that of all pixels in the image.

[††]http://sampl.ece.ohio-state.edu/data/stills/sidba/index.htm

[†††]http://sipi.usc.edu/database/

[††††]http://www1.cs.columbia.edu/CAVE/software/curet/index.php

Table 1 Comparison of various execution times (*ms*) for natural images, medical images, and textural images.

Image type		The MECTSL algorithm	Our proposed algorithm
natural	Max.	1.83	1.57
	Mean	0.96	0.89
	Min.	0.44	0.39
medical	Max.	0.98	0.95
	Mean	0.73	0.70
	Min.	0.59	0.58
textural	Max.	1.48	1.38
	Mean	1.09	0.95
	Min.	0.79	0.53

cannot be exerted, and the effect of the efficiency of our algorithm becomes weaker and weaker with the increase of the density of an image from 0.8.

The experimental results on the natural images, the medical images, and the textural images are shown in Table 1. Because images in each categories have special characteristics with different connectivities, our algorithm shows different performances on them.

5. Concluding Remarks

In this paper, we presented a totally new method for the first scan of label-equivalence-based two-scan labeling algorithms. By our method, the number of times for checking pixels for assigning provisional labels and processing label equivalences is decreased; thus, the efficiency of labeling is improved. Experiments demonstrated that our method was more efficient than the first scan of conventional label-equivalence-based two-scan labeling algorithms.

Acknowledgements

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Japan, Grant-in-Aid for Scientific Research (C), 23500222, 2011.

References

[1] A. Alexey, K. Tomas, W. Florentin, and D. Babette, "Real-time image segmentation on a GPU. Facing the multicore-challenge," Lect. Notes Comput. Sci., vol.6310, pp.131–142, 2011.
[2] D.H. Ballard, Computer Vision, pp.178–182, Prentice-Hall, Englewood, New Jesey, 1982.
[3] F. Chang, C.J. Chen, and C.J. Lu, "A linear-time component-labeling algorithm using contour tracing technique," Comput. Vis. Image Understand., vol.93, pp.206–220, 2004.
[4] C. Wolfe, T.C.N. Graham, and J.A. Pape, "Seeing through the fog: An algorithm for fast and accurate touch detection in optical tabletop surfaces," ACM International Conference on Interactive Tabletops and Surfaces (ITS '10), pp.73–82, New York, NY, USA, 2010.
[5] B. Dellen, E.A. Erdal, and F. Wrgtter, "Segment tracking via a spatiotemporal linking process including feedback stabilization in an n-D lattice model," Sensors, vol.9, no.11, pp.9355–9379, 2009.
[6] R.C. Gonzalez and R.E. Woods, Digital Image Processing, 2nd ed., pp.157–163, Addison Wesley, 1992.
[7] R.M. Haralick and L.G. Shapiro, Computer and Robot Vision I, pp.28–48, Addison-Wesley, Reading, MA, 1992.
[8] L. He, Y. Chao, and K. Suzuki, "A linear-time two-scan labeling algorithm," 2007 IEEE International Conference on Image Processing (ICIP), pp.V-241–V-244, San Antonio, Texas, USA, 2007.
[9] L. He, Y. Chao, and K. Suzuki, "A run-based two-scan labeling algorithm," IEEE Trans. Image Process., vol.17, no.5, pp.749–756, 2008.
[10] L. He, Y. Chao, K. Suzuki, and K. Wu, "Fast connected-component labeling," Pattern Recognit., vol.42, pp.1977–1987, 2009.
[11] L. He, Y. Chao, and K. Suzuki, "An efficient first-scan method for label-equivalence-based labeling algorithms," Pattern Recognit. Lett., vol.31, pp.28–35, 2010.
[12] L. He, Y. Chao, and K. Suzuki, "Two efficient label-equivalence-based connected-component labeling algorithms for three-dimensional binary images," IEEE Trans. Image Process., vol.20, no.8, pp.2122–2134, 2011.
[13] L. He, Y. Chao, and K. Suzuki, "A run-based one-and-a-half-scan connected-component labeling algorithm," Int. J. Pattern Recognit. Artif. Intell., vol.24, no.4, pp.557–579, 2011.
[14] Q. Hu, G. Qian and W.L. Nowinski, "Fast connected-component labeling in three-dimensional binary images based on iterative recursion," Computer Vis. Image Understand., vol.99, pp.414–434, 2005.
[15] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. Syst. Man Cybern., vol.9, pp.62–66, 1979.
[16] Y. Shima, T. Murakami, M. Koga, H. Yashiro, and H. Fujisawa, "A high-speed algorithm for propagation-type labeling based on block sorting of runs in binary images," Proc. 10th Int. Conf. Pattern Recognition, pp.655–658, 1990.
[17] K. Suzuki, I. Horiba, and N. Sugie, "Linear-time connected-component labeling based on sequential local operations," Comput. Vis. Image Understand., vol.89, pp.1–23, 2003.