A Modified Recursive X-Y Cut Algorithm for Solving Block Ordering Problems

Phaisarn Sutheebanjard
Graduate School of Information Technology
Siam University
Bangkok, Thailand
mr.phaisarn@gmail.com

Wichian Premchaiswadi
Graduate School of Information Technology
Siam University
Bangkok, Thailand
wichian@siam.edu

Abstract—**To achieve the best results from an OCR system, the pre-processing steps must be performed with a high degree of accuracy and reliability. There are two critically important steps in the OCR pre-processing phase. First, blocks must be extracted from each page of the scanned document. Secondly, all blocks resulting from the first step must be arranged in the correct order. One of the most notable techniques for block ordering in the second step is the recursive x-y cut (RXYC) algorithm. This technique works accurately only when applied to documents with a simple page layout but it causes incorrect block ordering when applied to documents with complex page layouts. This paper proposes a modified recursive x-y cut algorithm for solving block ordering problems for documents with complex page layouts. This proposed algorithm can solve problems such as (1) the overlapping block problem; (2) the blocks overlay problem, and (3) the L-Shaped block problem.**

Keywords; recursive x-y cut; block ordering;

I. Introduction

Optical Character Recognition (OCR) is used widely in electronic publishing because it cans significantly speedup the acquisition of text. In the past few decades, many researchers have been investigating techniques for electronically converting printed documents into a structured format in order to reuse them in a cost effective manner. One of the most important issues in the OCR pre-processing phase is the correct extracting and ordering of blocks of content from documents with a complex layout. It is difficult to archive the correct result if the structure of the block layout in the document is complex.For example, one of the complex layout problems is the L-Shaped problem which was studied by [1] as shown in Fig. 1. They proposed the elimination block technique in which any block, which prevented use of the recursive x-y cut process, would be eliminated from such a document. This step is repeated until the x-y cut process has been completed. After completing the x-y cut process, the eliminated blocks are inserted back into the spaces in the document left by the elimination procedure in order to determine the correct order of the blocks. Unfortunately, they did not mention the method used for putting the eliminated blocks back into local spaces in the document nor did they provide a clear solution for solving this problem.

This paper proposes a modified recursive x-y cut algorithm for solving block ordering problems for OCR document page segmentation. This algorithm can be used to solve not only the L-shaped block problem but also other complex layout problems. The details of the complex layout problems will be described later in this paper.

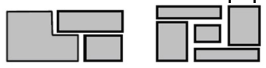


Figure 1. L-shapes defeating the recursive x-y cut [2]

II. Fundamentals

A. Block Ordering Problem

This paper discusses three types of complex document page layouts. They are documents pages with overlapping blocks, overlaying blocks, and L-shaped blocks [1, 2] as shown in Fig. 2. These problems cannot be solved with the well-known recursive x-y cut technique [2-4]. Ishitani presented the reading order determination technique for document elements to solve the L-shaped problem but did not identify the problem-solving methodology clearly. This paper proposes a modified recursive x-y cut algorithm for solving block ordering problems, especially the L-shaped block problem. Besides solving the L-shaped problem, this modified algorithm can be used to solve the other two defined block ordering problems.

Three types of block ordering problems are defined as the following: First, the overlapping block problem is where a block in one column overlaps with another block on a different column as shown in Fig. 2 (a). Second, the overlaying block problem is a block inserted on a page, which overlays other blocks contained on different columns as shown in Fig. 2 (b). Third, the L-shaped block problem is where adjacent blocks form an L-shape as shown in Fig. 2 (c).

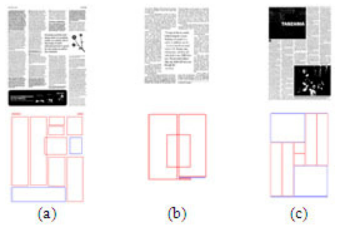


Figure 2. Complex layout document problems
(a) Overlapping block problem
(b) Overlaying blocks problem
(c) L-Shape block problem

The document layout as shown in Fig. 2 represent two colors, red for blocks of character and blue for blocks of pictures.

B. *Background on Recursive X-Y Cut*

One of the most important processes in Optical Character Recognition (OCR) system is the page reading order of the document elements (this step is called block ordering) because it must be consistent with the human reading order. The recursive x-y cut technique has been widely used in block extraction [2-4] and block ordering [1]. This technique is used for checking for horizontal and vertical spaces, so it works well for simple geometric layouts. But in a complex layout document as shown in Fig. 2, the original recursive x-y cut does not yield the correct results. The basic idea of the recursive x-y cut technique was proposed by Nagy and Seth [5]. The recursive x-y cut technique attempts to subdivide a rectangle such as a page into smaller rectangles by making cuts along a horizontal and vertical direction until it cannot further partition the rectangle. Currently, the recursive x-y cut method is widely used in OCR processes for block extraction and block ordering.

1) Recursive X-Y Cut for Block Extraction: Block extraction processes can be classified into three groups, namely, top-down [7], bottom-up [8], and a hybrid approach [9, 10]. The recursive x-y cut technique uses the top-down approach. After performing the block extraction, blocks must be arranged in the correct order [2-4]. There is a technique that can be used to customize the recursive x-y cut for the fast computation such as using bounding boxes of connected components [7] and by using black pixels instead of using image pixels.

2) Recursive X-Y Cut for Block Ordering: The block ordering process is defined as putting the extracted blocks into a human reading order. The input to the block ordering process is a rectangle block extracted from a scanned page. Using a sorting pattern from top-down and left-to-right is a simple way to arrange blocks contained on the page. But this sorting pattern technique can only be used with a document that contains a single column. Other techniques are proposed for solving the block ordering problem for documents that contain more than one column such as soft ordering [11], nested segmentation [12], and recursive x-y cut [1, 2].

III. THE PROPOSED METHOD

The proposed system is divided into 2 processes. The first process is block extraction to extract blocks (block of picture and/or block of character) and create optimum image by using the technique of Premchaiswadi and Sutheebanjard [6]. The second process is block ordering to determine the page ordering. The input of the block ordering process is the optimum image from the block extraction process. This research mainly focuses on the block ordering process.

During the process of block ordering we use the recursive x-y cut on the optimum image for faster computation. And we use the binary tree to represent the data structure to represent the layout of objects in a two-dimensional space. This is obtained by applying the x-y cuts recursively on the optimum image. Each leaf node of a binary tree and some parent node correspond to a block of characters or a picture. In this section we proposed the new algorithm to solve the problems that the recursive x-y cut algorithm fails to solve for block ordering of complex document layouts such as:

- Overlapping block problem: we solve with a normalized block weight and block height algorithm.
- Overlaying block problem: we solve by removing the overlap block.
- L-Shape block problem: we solve using a get first node technique.

In the block ordering process we solve the first and second problem listed above before using the recursive x-y cut as one of the preprocessing steps. The L-shaped problem we solve during the run time of the recursive x-y cut process. This technique can be separated into 3 steps as: First solve the overlapping block problem with normalize block width of all blocks in the same block column and normalize the block height of all blocks in the same heading text block row. Second, solve the problem of the overlaying block problem by finding the block that lies in between other blocks and do not draw it in the optimum image. Third, solve the L-shaped block problem during the run time of the x-y cut process by checking if this tree node cannot project into both the horizontal and vertical then remove the first top-left block in this border node.

A. *Reduce Block Border*

In the section, we discuss finding the perimeter of a block of characters and a picture in the optimum image with a dilation of 3x3 pixels (dilation only block of characters). But it should ne noted that in the process of block ordering we just use the rectangle border. First, we calculate block border (left, top, right, and bottom) from the perimeter. Second, if it is a block of characters we reduce each side by 1 pixel (1 pixel in the optimum image equals 12 pixels in the actual image); so it will be the actual border size as shown in Fig. 3.

B. *Normalize Block Width*

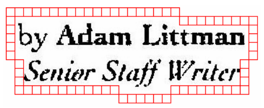
Some blocks may be wider than other blocks in the same column. We normalize the widths to in the same column by checking every pair of block in the same column. If a column has blocks with different widths we will find the new width and new center and assign to the wider block using (1) and (2).

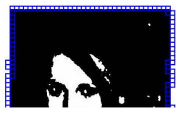
$$w = \frac{w_1 * h_1 + w_2 * h_2}{h_1 + h_2} \tag{1}$$

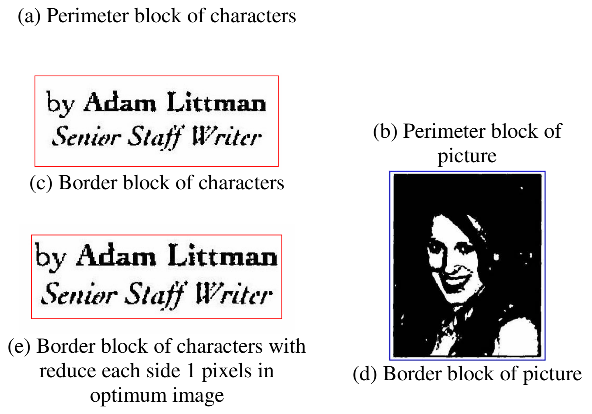
And we will find the new horizontal center

$$center\_x = \frac{center\_x_1 + center\_x_2}{2} \tag{2}$$

w_1, h_1, and $center\_x_1$ are property of first block and w_2, h_2, and $center\_x_2$ are property of second block. w, and $center\_x$ are property that will assign to the wider block.







(a) Perimeter block of characters

(b) Perimeter block of picture

(c) Border block of characters

(d) Border block of picture

(e) Border block of characters with reduce each side 1 pixels in optimum image

Figure 3. Perimeter and border block

We assign center_x to be the new center for the wider block and assign w to be new width of the wider block. The results of this process were shown in Fig. 4.

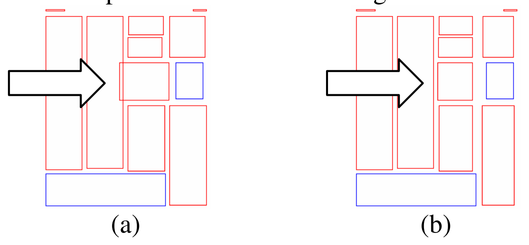


Figure 4. Result of normalize block width.

C. *Normalize Block Height*

Most bottom-up approaches to block extraction have a problem with header text as shown in Fig.5 where each word is shown in a separate block. In our approach, we didn't merge these into the same block , we just arranged them in the right order. As seen in Fig. 5 (c). it is not possible to separate blocks in the horizontal (show in blue dash) because 2 block are touching. In this case we will reduce the block height. If block is height less than a thredshold value (in this paper use 16 pixels in the optimum image or 192 pixels in the actual image) we will reduce its height using (3) and (4).

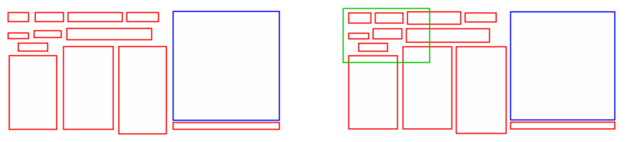
$$h = \frac{w_1 * h_1 + w_2 * h_2}{w_1 + w_2} \tag{3}$$

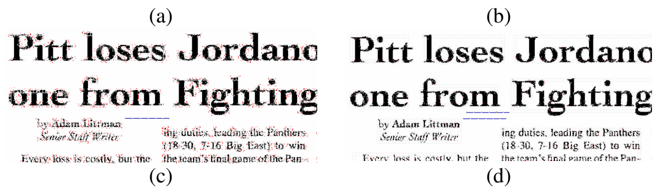
And we will find the new vertical center

$$center\_y = \frac{center\_y_1 + center\_y_2}{2} \tag{4}$$

The result as shown in Fig. 5 (d) between rows we have a space for the recursive x-y cut (shown by the blue dash).

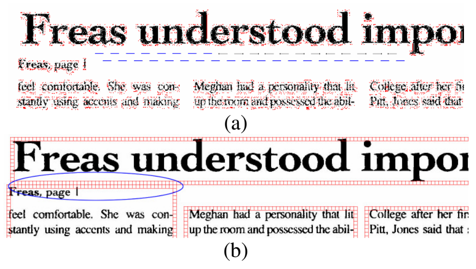
The process of normalizing block height cannot solve every case of vertical overlap such as shown in Fig. 6 (a) because after reducing the block height as shown in Fig. 6 (b) it is still a touching block (in the blue oval). For this case, we will fix in the next section.





(a)
(b)
Pitt loses Jordano
one from Fighting
by Adam Littman
Senior Staff Writer
ing duties, leading the Panthers (18-30, 7-16 Big East) to win the team's final game of the Pan-
Every loss is costly, but the
(c)
(d)

Figure 5. Normalize block height.



Freas understood impo
Freas, page 1
feel comfortable. She was constantly using accents and making
Meghan had a personality that lit up the room and possessed the abil-
College, after her fi Pitt, Jones said that
(a)
(b)

Figure 6. Touching block problem

D. *Finding and Removing Overlay Blocks*

In a magazine or newspaper, the abnormal layout as shown in Fig. 7 has a block overlapping another block. In this case, we will remove it as shown in Fig. 7 (c) and (d).

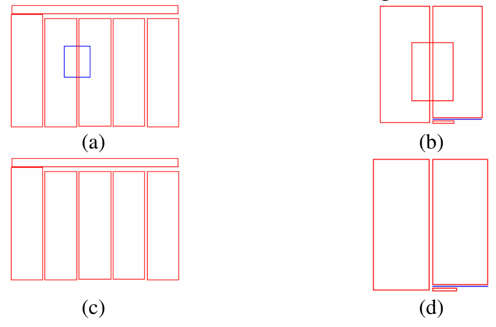


Figure 7. Overlaying block
(a) overlaying block with picture
(b) overlaying block with block of character

E. *Draw Blocks in Optimum Image*

In the processes described above (Reduce Block Border, Normalize Block Width, Normalize Block Height, Mask Overlap Block) we computed all measures from the scanned input data. In this section we draw all block borders except masked overlap blocks using the new optimum image.

F. *Generate Binary Tree*

This section we will demonstrate the algorithm to solve the general recursive x-y cut problems. The first step is to create the first node in a tree structure and assign all blocks to it. Second, compute the border of this node from the border of the extracted blocks. Third, use the recursive x-y cut technique using this border node as the start of the horizontal cut. If cut can be made in the horizontal at x, create a child node (left node) and move all nodes above x to the left node and recursively start with the left node for a vertical cut. Create a child node (right node) and move all nodes below x to a right node and recursively use the right node as the start for a horizontal cut. But if it cannot cut vertically any of x it will cut in the vertical with this node and increase the level by 1. We do the same thing for the vertical cuts as well. If it can cut in vertical at y, create a child node (left node) and move all nodes on the left side of y to left node and recursively use the left node as the start of a horizontal cut. Create child node (right node) and move all nodes to the right side of y to right node and recursively use the right node as the start of a vertical cut. But if it cannot cut any of y it will cut horizontally with this node and increase the level by 1.

```
HorizontalCut(level, node)
  If level = 3 return
  If can cut at x
    create leftNode
    create rightNode
    assign upper block to left node
    assign lower block to right node
    compute left node border
    compute right node border
    VerticalCut(1, nodeLeft)
    HorizontalCut(1, nodeRight)
    return
  VerticalCut(level+1, node)
VerticalCut(level, node)
  If level = 3 return
  If can cut at y
    Create leftNode
    Create rightNode
    Assign left block to leftNode
    Assign right block to rightNode
    compute left node border
    compute right node border
    HorizontalCut(1, nodeLeft)
    VerticalCut(1, nodeRight)
    return
  HorizontalCut(level+1, node )
```

G. *Get First Node*

In the case of L-Shape [1] block problem as shown in Fig. 8 (a) the general recursive x-y cut cannot achieve the right ordering because it cannot cut in both the horizontal and vertical directions. In this case, we will delete the border of the first block from the optimum image as shown in Fig. 8 (b) and create a left child node and move this block to the left node. Create a right child node and move another block to the right node and start recursively the x-y cut at the vertical as shown in algorithm below.

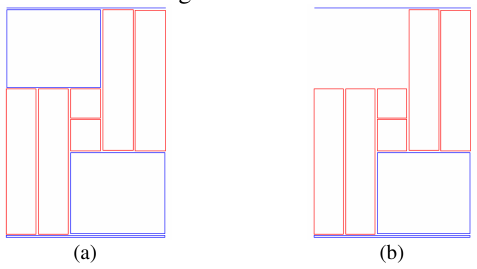


Figure 8. L-shape problem solved by removing the first block

```
HorizontalCut(level, node)
  If level = 3
    If block in this node > 1
      Delete first block in optimum image
      create leftNode
      create rightNode
      assign first block to left node
      assign another block to right node
      compute right node border
      VerticalCut(1, nodeRight)
      return
    return
  If can cut at x
    create leftNode
    create rightNode
    assign upper block to left node
    assign lower block to right node
    compute left node border
    compute right node border
    VerticalCut(1, nodeLeft)
    HorizontalCut(1, nodeRight)
    return
  VerticalCut(level+1, node)
VerticalCut(level, node)
  If level = 3
    If block in this node > 1
      Delete first block in optimum image
      create leftNode
      create rightNode
      assign first block to left node
      assign another block to right node
      compute right node border
      VerticalCut(1, nodeRight)
      return
    return
  If can cut at y
    Create leftNode
    Create rightNode
    Assign left block to leftNode
    Assign right block to rightNode
    compute left node border
    compute right node border
    HorizontalCut(1, nodeLeft)
    VerticalCut(1, nodeRight)
    return
  HorizontalCut(level+1, node )
```

H. *Get Results*

After obtaining the binary tree, block order traversal is determined. In the case of a no-overlay block problem, parent nodes represent the overall boundary block of child nodes and child nodes represent the boundary of each block of characters or a picture see Fig. 9 (a). In other cases, some parent nodes represent the block that overlays with other blocks and a child node still represents the boundary of each block of characters or a picture see Fig. 9 (b). (In Fig. 9 (b) the overlaying block is block B)

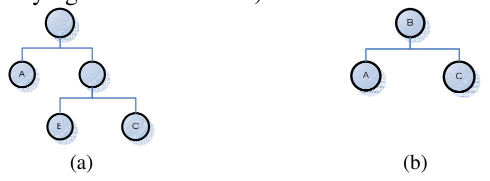


Figure 9. (a) Example of binary tree with no overlap block
(b) Example of binary tree with overlaying block

IV. Experimental Results

The experimental results of each block ordering problem are described in Fig. 10. The yellow rectangle color show the center of each block and the green line show the sequence of block. Fig. 10 (a) shows the vertical overlapping block problem. This type of problem can be solved and blocks can be segmented vertically. Fig. 10 (b) shows the horizontal overlapping block problem. This type of problem can also be solved and blocks can be segmented horizontally. Without this algorithm blocks cannot be correctly segmented vertically and horizontally. The overlaying block problem is shown in Fig. 10 (c) and (d). With overlapped blocks, blocks cannot be segmented vertically using the traditional x-y cut algorithm. This complicated problem can be effectively solved by using this modified algorithm. The most complicated problem that the x-y cut algorithm can not solve is the L-shape block problem shown in Fig. 10 (e), since the block cannot be segment vertically and horizontally. This new modified algorithm based on recursive x-y cut can handle this problem completely.

V. Conclusion

We performed experiments using newspaper and magazine articles because of their layout complexity. The three types of layout problems described previously frequently occurred in both the newspaper and magazine articles. In these experiments, the L-shaped block problem was the most complicated since no previously published information could identify a method of solving this problem explicitly. This paper proposed a new algorithm based on the recursive x-y cut technique that successfully solved all the described problems including the L-shaped block problem. needed for accurate page segmentation for OCR applications. The advantages of this algorithm are two fold. First, it can solve the most common complex page layout types of block problems. Second, it is fast computation because it works using the optimum image instead of real image. The size of the image is reduced by approximately 144 times its original size.

References

[1] Y. Ishitani, "Document transformation system from papers to XML data based on pivot XML document method," ICDAR.2003, pp.250-255, 2003.

[2] J. L. Meunier, "Optimized XY-Cut for Determining a Page Reading Order," ICDAR.2005, pp.347-351, 2005.

[3] A. Takasu and K. Aihara, "Information extraction from scanned documents by stochastic page layout analysis," SAC'08, 2008.

[4] S. Watcharabutsarakham, "Page Segmentation for Content Sequence," ICOSP, 2006.

[5] G. Nagy and S. Seth. "Hierarchical representation of optically scanned documents," ICPR, volume 1, pp. 347–349, 1984.

[6] W. Premchaiswadi, P. Sutheebanjard and N. Premchaiswadi, "The fast scheme for document page segmentation in OCR using window and optimum image," WSEAS, pp7-12, 2006.

[7] J. Ha, R. M. Haralick and I. T. Phillips, "Recursive X-Y Cut using Bounding Boxes of Connected Components," IEEE, pp 952-955, 1995.

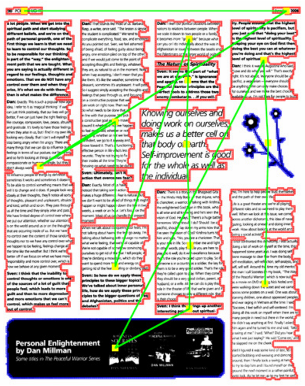
[8] D. Drivas and A. Amin, "Page Segmentation and Classification Utilising Bottom-Up Approach" IEEE, pp 610-614, 1995.

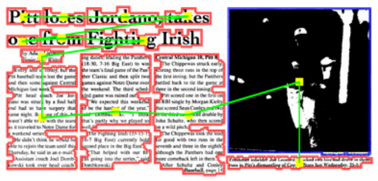
[9] P. Sutheebanjard, W. Premchaiswadi and Nucharee Premchaiswadi, "A Fast Block Extraction Method for Document Page Segmentation," EECON26, Vol. 2, pp.1069-1074, November 2003.

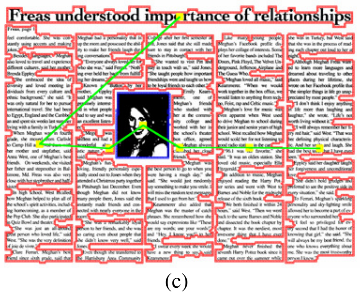
[10] B. Kruatrachue, P. Suthaphan, "A Fast and Efficient Method for Document Page Segmentation for OCR," Electrical and Electronic Technology, 2001. Vol. 1, pp.381-383, Augugst 2001.

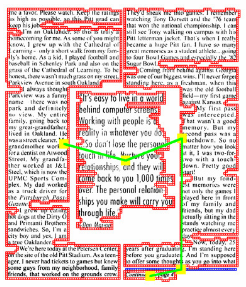
[11] P. E Mitchell and H. Yan, "Document page segmentation and layout analysis using soft ordering," ICPR, pp.458-461, 2000.

[12] X Hao, J T L Wang and P A Ng, "Nested segmentation an approach for layout analysis in document classification," 2nd ICDAR, 1993


(a)


(b)


(c)


(d)

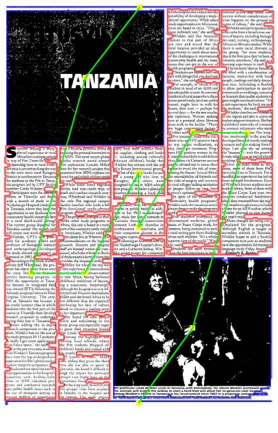

(e)

Figure 10. Experimental results
(a) Vertical overlapping block problem.
(b) Horizontal overlapping block problem.
(c) Overlaying block problem.
(d) Overlaying block problem
(e) L-Shape block problem.