Available online at www.sciencedirect.com

Image and Vision Computing 24 (2006) 94–100

www.elsevier.com/locate/imavis

Sliding window based approach for document image mosaicing

P. Shivakumara [a,*], G. Hemantha Kumar [b], D.S. Guru [b], P. Nagabhushan [b]

[a] *Department of Computer Science, School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore*
[b] *Department of Studies in Computer Science Manasagangotri, University of Mysore, Mysore-570006, Karnataka, India*

Received 11 February 2004; received in revised form 19 September 2005; accepted 24 September 2005

Abstract

There are situations where it is not possible to capture a large document with a given imaging media such as Scanner and Xerox machine in a single stretch because of their inherent limitations. This result in capturing a large document in terms of split components of an image. Hence, the need is to mosaic the split components into a large document image.

In this paper, we present a new and simple approach to mosaic the two split images of a large document based on matching sum of values of pixels of window in the split images. The method compares the sum of values of pixels of window in split images to identify Overlapping Region (OLR) in the split images. The OLR, a region in common, helps in mosaicing of two split images of large document. However, a small OLR is assumed to be available at the end of split images of a large document. In addition to this, the OLR in the split images depends on the size of the window. Experimental results show that the performance of the proposed method is satisfactorily.

Keywords: Window matching; Overlapping region; Document image mosaicing

1. Introduction

Many a times, it may not be possible to capture the complete image of a large document in a single exposure as most of he image capturing media works with documents of definite size. In such cases, the document has to be scanned part by part producing split images. Thus, the document image analysis and processing require mosaicing of split images to obtain a complete image of the document. Hence, the document image mosaicing is the process of merging split images that are obtained either by scanning or capturing a single large document image a part by part with some sort of OLR in order to restore a original and complete document image without any duplication of portions.

Several researchers have proposed the methods for obtaining the large image from its split images. Schutte and Vossepoel (1995) described the usage of flat bed scanner to capture large utility map [9]. The method selects the control points in different utility maps to find the displacement required for shifting from one map to the next. These control points are found from pair of edges common to both the maps. However, the process requires human intervention to mask out the region not common to both the split images in image mosaicing.

[1,2] have proposed method for Document Image Mosaicing (DIM). A feature-based approach through estimation of the motion from point correspondence is proposed. The exhaustive search adopted was computationally expensive because of the rotation of an image employed during matching. In addition, the method demands 50% OLR in the split images to produce mosaiced image. However, the approaches are limited to only text documents and are prone to failure in case of general documents containing pictures. But in practice, a typical document contains both text and pictures.

An automatic mosaicing process for split document images containing both texts and pictures, based on correlation technique is proposed by [3]. Here correlation technique was used to find the position of the best match in the split images. However, accuracy is lost at the edges of the images. Moreover, the correlation of two images of practical size is computationally very expensive. In order to find a solution, additional constraints like a priori knowledge were introduced. Here, the sequence in which the images were captured and their placement (generally, referred as image sequencing) is known. Template matching procedure was used to search OLRs, present in the split document images. Usually, template-matching procedure is a time consuming method. In addition, this approach assumes that the printed text lies on straight and

[*] Corresponding author. Tel.: +65 687 46806.
E-mail addresses: hudempsk@yahoo.com (P. Shivakumara), shiva@comp.nus.edu.sg (P. Shivakumara).

0262-8856/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2005.09.015

horizontal baselines, which is not always possible in many of the pragmatic applications.

The same authors of this paper proposed [4–8] new techniques to tackle the above-mentioned problems. The proposed technique works for any type of document without considering the nature of the content present in the document to produce a complete large document image without having a priori knowledge about the order of image sequence. The proposed technique demands at least 1 pixel wide (1–2%) OLR in the split images. The OLR is present at the right and the left ends of the first and the second split images, respectively. The technique is based on PMA (Pattern Matching Approach). A PMA is employed to determine the OLR in the split images of a large document image.

From the above literature, it is found that the some methods work for all kinds of documents but at the cost of computations. Some are simple and efficient but are not accurate. Hence, we strongly believe that there is a need for developing method for Mosaicing the split images and which is to be accurate as well as simple.

The approach presented in this paper, works for any type of document irrespective of its content. The method works based on matching sum of values of pixels of window of split images to identify overlapping region in the split images. The overlapping region, region in common, helps in mosaicing of two split images. Thus, the crux of work relies on finding out the OLR in the split images efficiently. This is achieved by comparing the sum of values of pixels of each window in the split images. The advantage of this approach is that no exhaustive search is required. The proposed method has the worst case complexity $O(n^3)$ where n is the number of columns in the split images. However, the method works based on assumption that the OLR is present at the right end of the split image 1 and the left end of the split image 2.

This paper is organized as follows. Section 2 describes the proposed methodology to mosaic two split images. Section 3 explains the time complexity analysis of proposed algorithms. Experimental results are reported in Section 4. Conclusion is given at the end.

2. Proposed methodology

In this section, new approaches based on sliding window are presented. The method computes sum for each window in both the images split image 1 and split image 2 and it compares the sum in the split images to identify the OLR in the split images. The OLR is obtained by comparing the sum of values of pixels of window of split images. The OLR in the split images depends on the size of the window considered. If the OLR in the split images is lesser than the size of the window then the proposed method fails to identify the OLR in the split images.

The proposed method is divided into three sections. In first section we describe a unidirectional scanning method. The bidirectional scanning method is introduced in Section 2. An improved unidirectional scanning method is presented in Section 3 to overcome the drawbacks of the algorithms presented in Sections 1 and 2.

2.1. Unidirectional scanning method (USA)

In this sub section, we present an algorithm to obtain a mosaiced image from its split images with time complexity $O(n^2)$ in worst case based on comparing sum of values of pixels of window of the split images of a large document. If match found then the pointers of split image 1 and split image 2 are moved to next column. This procedure is repeated till the overlapping region is found in the split images. If the match is not found then the pointer of the split image 1 is moved to the next column of split image 1 unconditionally. But the pointer of the split image 2 remains as in the first column of the split image 2. This is true because of the fact that the OLR is present at the right and left the ends of split image 1 and image 2, respectively. However, the method fails when the split image 1 contains more OLR compared to OLR in the split image 2. The following Fig. 1 shows that the process of obtaining mosaiced image from two split images S_1 and S_2. In which n and m represent number of columns and number rows of split images, respectively. i and j are the pointers of S_1 and S_2 respectively keep track of the positions of columns and rows of images. The pointer i is moving towards right as shown in the Fig. 1 to find OLR by comparing with jth column of S_2. If whole column of S_1 and S_2 matches then the both the pointers incremented by one. Otherwise only i move towards right j remains as in the first column position. The algorithm terminates when the pointer i of S_1 reaches n. Here, the windows are moving in column wise. If all 3×3 windows match in column then the pointer of S_1 and pointer of S_2 moved to next column. If even one window does not match then the pointer of S_1 moves to next column but the pointer of S_2 remains in first column. This procedure is continued till i reach n of S_1. It is cleared from the Table 1 that as size of the window increases the number of comparisons also decreases. The percentage of comparisons is also calculated for 64×64 image of 4096 pixels. For 34×34 sized window the algorithm gives no overlapping in the split images. This is because the overlapping in the split images is about 33×33. This shows that the overlapping region in the split images is depending upon the size of window.

The result of sliding window over split image 1 and split image 2 stored in matrix 1 and matrix 2, respectively. These two matrices are considered as input for finding an overlapping region in algorithm 1, algorithm 2 and algorithm 3.

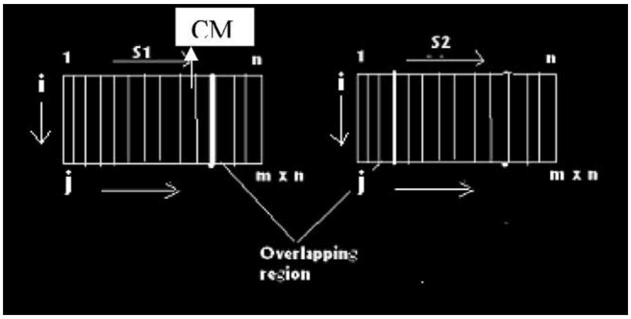

Fig. 1. Mosaicing of two split images using unidirectional scanning method.

Table 1
comparative study by varying size of window with the number of comparisons for english document (Fig. 4)

Size of window	No. of comparisons at window level	In terms of % for 64× 64 image (4096 pixels)
2×2	2185	53.3
3×3	1975	48.2
4×4	1866	45.5
5×5	1772	43.2
6×6	1683	41.0
7×7	1597	38.9
9×9	1431	34.9
10×10	1351	32.9
15×15	981	23.9
20×20	661	16.1
25×25	392	9.5
30×30	171	4.1
33×33	63	1.5
34×34	No overlap	–

Algorithm 1. : Unidirectional Scanning Method

Description: i and j are the pointers to columns of split image (S_1) and Split image (S_2). n

is the number of columns present in the split image (S_1). CM is a Boolean variable. The value (CM = true) means whole column is matched. The value (OLR = 1) means overlapping region is found.

Input: S_1 is matrix 1 and S_2 is matrix 2

Out put: Mosaiced image

Method Begins

```
  Step 1: For each Column (j) of $S_1$ and $S_2$ do
 Step 1.1: For each Row (i) of $S_1$ and $S_2$ do
                                              Step 1.1.1:
If ($S_1$ (i, j) = $S_2$ (i, j)) then
     i = i + 1
   for end
 Step 1.2: If (entire column matches) then
     CM = True
   else
     j = j + 1
   end if.
 for end.
  Step 2: If ((i = m) and (CM = True)) then
   j = j + 1 in both $S_1$ and $S_2$
 for end
  Step 3: If ((j = n) and (CM = True)) then
   OLR = 1
 else
   (OLR = 0)
  Step 4: If (OLR = 1)
 Translate the split image ($S_2$) such that the overlapping
 regions of both the images match each other with respect to
 their coordinates.
else
 Report existence of no overlapping region

Method ends
```

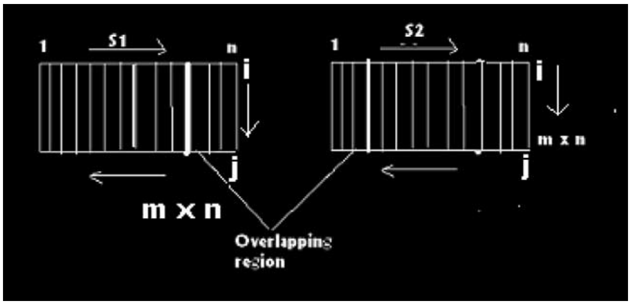

Fig. 2. Mosaicing of two split images with O(n^2) time complexity by back tracking.

2.2. Bidirectional scanning method (BSM)

The algorithm 1 fails when the OLR is more in the split image 1 compared to split image 2. In order to overcome the drawback of algorithm 1 in this section, we present an extended version of algorithm 1 with same time complexity. The method scans the split image from right to left as well as left to right by comparing the sum of values of pixels of window but algorithm 1 scans the image only from left to right to identify the overlapping region in the split images. This method works fine for some cases but not for all cases. That is presented in Fig. 3. The Fig. 2 is similar to the Fig. 1 except some back tracking. The algorithm first scans left to right as in the case of algorithm 1. In spite of this, the algorithm scans from right to left by changing the pointers positions. The algorithm finds the distance between the first column and the matching area begins while scanning left to right as in the case of algorithm 1. Similarly the algorithm computes the distance from first column of split image 1 to beginning position of OLR while scanning the split images from right to left. This is shown in Fig. 2. It works even the split image 1 contains more overlapping region. However, the algorithm fails to handle the situation shown in the Fig. 3. In this case actual match starts from 4th column of S_1 but if we employ the algorithm2 we get no match is found though there is an OLR in the S_1 and S_2. The remedy for this is presented in Section 2.3 with high time complexity. This is because the algorithm matches 3rd column of S_1 with first column of S_2. Both the pointers are incremented by one since match exists. Now i pointer pointing to 4th

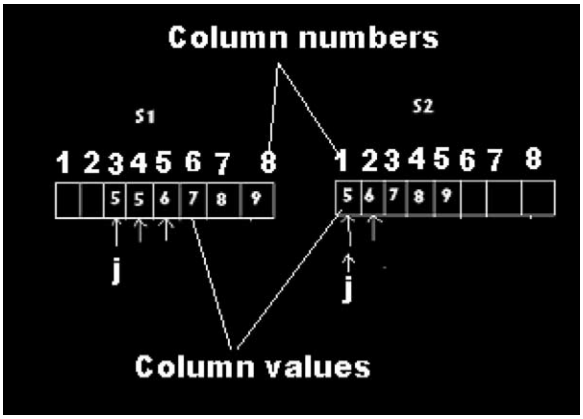

Fig. 3. Situations where the algorithm 2 fails.

column of S_1 and j is pointing to second column of S_2. There is no match in these positions. Then the pointer i moved to 6th column of S_1 mean time the pointer j of S_2 return back to first column position. The algorithm continues to find match by incrementing the i in S_1. This result in no OLR found in the split images. In Fig. 3 the values 5, 6,7, 8, 9 are assumed to be values in respective column sums (that means whole column is matching).

Algorithm 2. : Bidirectional Scanning Method

```
Input: $S_1$ is Matrix 1 and $S_2$ is Matrix 2
Output: Mosaiced image
Method Begins

Step 1: Find out the distance ($D_1$) between the first column
        and starting position of OLR in $S_1$ by employing
        Algorithm 1.
Step 2: Change the positions of pointers (i) and (j) of $S_1$ to
        nth column of $S_1$ and the pointers (i) and (j) of $S_2$ to
        nth column of the $S_2$ (refer Fig. 2).
Step 3: Find out the distance ($D_2$) between the first column
        and starting position of OLR in $S_1$ by employing
        Algorithm 1.
Step 4: If ($D_2 < D_1$) then position of actual OLR in $S_1$ starts
        from $D_2$.
Step 5: Translate the split image ($S_2$) such that the
        overlapping regions of both the images match each
        other with respect to their coordinates.

Method ends
```

2.3. An improved unidirectional scanning method (IUSM)

In this section, we present an algorithm to mosaic split images of different cases. This algorithm works for all type of documents. However, time complexity is high compared to algorithm 1 and algorithm 2. The presented algorithm uses the algorithm 1 to mosaic split images for different cases. For each column the algorithm invokes the algorithm 1 to mosaic the split images. Let n be the number of column in the split image1. For each column the algorithm invoke the algorithm 1 to identify the OLR in the split images. If the pointer of S_1 of invoking algorithm reaches n without failing then the method is said to be terminate successfully. Otherwise the method again invokes an algorithm from next column of the split image 1. This procedure is continued till we get overlapping region. For an instance, consider situation given in Fig. 3. The algorithm invokes the algorithm 1 from column 3rd then the pointer of S_1 moves to 5th column mean time the pointer in S_2 moves to 2nd column. As a result of this no mosaiced image is obtained since no match found in the split images. That means count (refer algorithm 3) becomes $=0$ and i reaches n. In this situation pointer j will return back to first column of S_2 and i return back to next column (4th) in S_1. In this way, the algorithm successfully gives the results when the count $!=0$ and i reaches n.

Algorithm 3. : Improved Unidirectional Scanning Method

```
Input: $S_1$ is Matrix 1 and $S_2$ is Matrix 2
Output: Mosaiced image
Method Begins

  Step 1: For each column invoke the Algorithm 1 to
          identify the overlapping region in the split images.
Step 1.1: If (the column of $S_1$ doesn't match with column of
          $S_2$) then note down the values of count and the
          value of pointer i
Step 1.2: If ((count! $=0$) and ($j=n$)) then
              Report that overlapping region is found
            else
              go to next column of $S_1$
          for end
  Step 2: Repeat the step 1 till Algorithm 1 satisfies the
          condition given in Step 1.2.
  Step 3: Translate the split image ($S_2$) such that the
          overlapping regions of both the images match
          with respect to their coordinates.

Method ends
```

3. Complexity analysis

The major step involved in document image mosaicing is in identifying the overlapping region in the split images efficiently. Hence, the complexity of the proposed algorithm depends on the method for finding out the overlapping region in the split images. For algorithm 1 the time complexity is $O(n^2)$ where n is the number of columns in the split images. However, this algorithm fails in some cases to obtain mosaiced image. We propose one more algorithm to overcome the drawback with same time complexity $O(n^2)$. Again this algorithm fails in some cases that is shown in the Section 2.2. The third algorithm works fine for all types of documents but it requires more time i.e. $O(n^3)$. This can be reduced by modifying the algorithm that is beyond the scope of this paper. The time complexity of three algorithms is same for best case. It is observed from the Table 2 that as number of comparisons increases the complexity of the algorithms also increases.

4. Experimental results

In this section, we present a few out of many experimental results. For each algorithm we have given different cases which ensure that our proposed algorithm works for any type of document irrespective of its content. In the following examples

Table 2
Comparative study of three methods with respect to English document (Fig. 4)

Algorithms	Number of comparisons of windows (3×3)	Time complexity
Algorithm 1	1975	$O(n^2)$
Algorithm 2	3941	$O(n^2)$
Algorithm 3	63357	$O(n^3)$

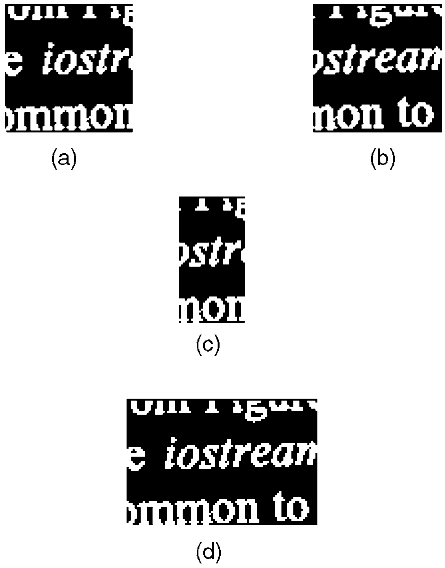

Fig. 4. Text document image.

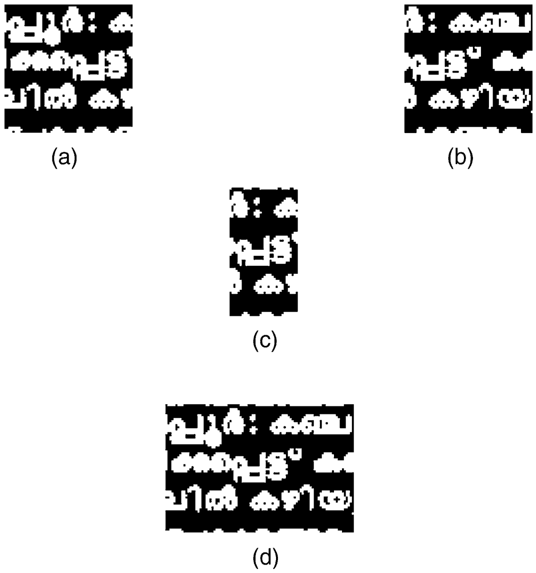

Fig. 6. Document containing Malayalam language text lines.

(a) and (b) are two input images. (c) is the overlapping region is obtained by the proposed methods. (d) is the final mosaiced image.

Case 1: In this case, the experimental results are given for the algorithm 1. This algorithm fails to give expected results as shown in the Fig. 8. In which, the input split image 1 (a) contains more OLR compared to split image 2 (b). In such situations the proposed algorithm fails to give complete overlapping region present in both the images. This is the major draw back of this method. Hence, an improved version of this method is presented in case 2 in order to eliminate the drawback of the method (Figs. 4–11).

Case 2: In this case we present the experimental results of algorithm 2. This is an improved version of algorithm 1 to eliminate the drawback of the algorithm 1. The detailed algorithm is described in Section 2.2. This algorithm works fine if the split image 1 contains more overlapping region compared to split image 2 or vice versa with same time complexity. This is shown in Fig. 12 and Fig. 13. With reference to the above case 1 this algorithm works satisfactorily particularly for Fig. 12. From Fig. 12 and Fig. 13 we noticed that the algorithm identifies the complete overlapping region in the split images.

Case 3: In this case, we present the experimental results of algorithm 3 which works any type of document irrespective of its content. However, the complexity of the algorithm is high compared to algorithm 1 and 2 this algorithm identifies the complete overlapping region in the split images. This is shown in the following Fig. 14 and Fig 15.

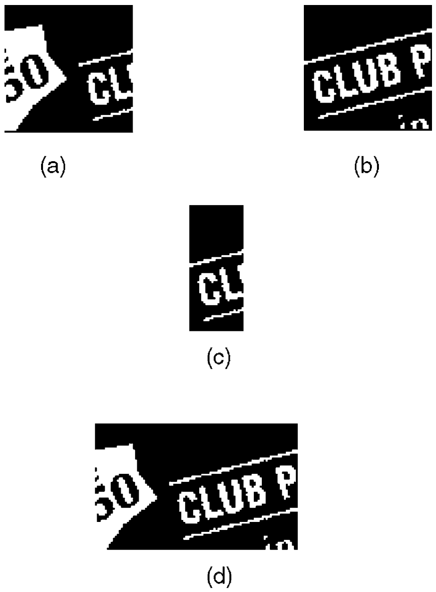

Fig. 5. Text and picture document image.

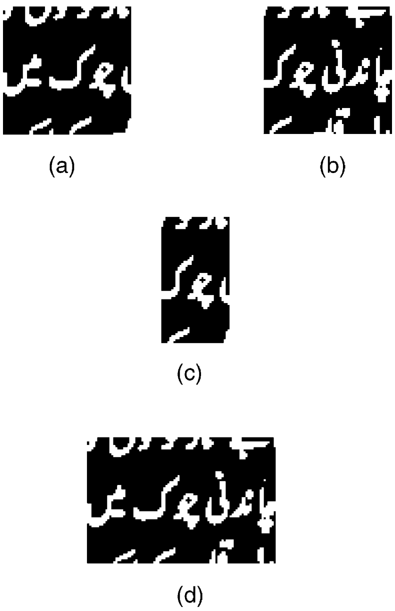

Fig. 7. Document containing Urdu language text lines.

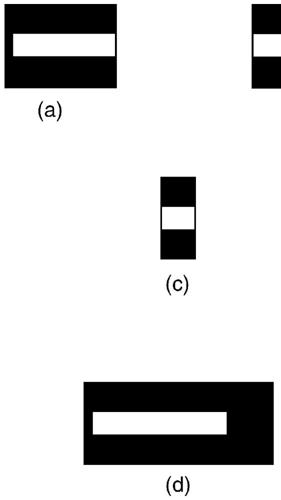

Fig. 8. Synthetic image which shows the algorithm 1 fails when the split image 1 (a) contains more overlapping region than split image 2 (b).

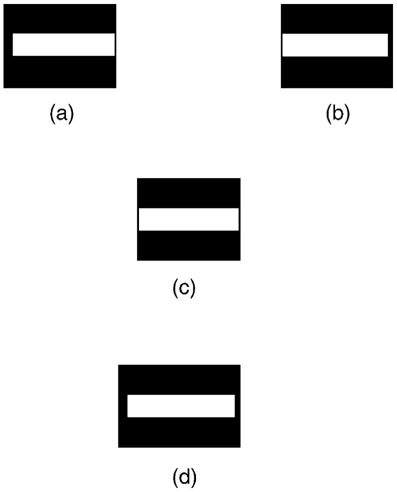

Fig. 9. Synthetic image which shows the algorithm 1 works when the split image 2 (b) contains more overlapping region than split image 1 (a).

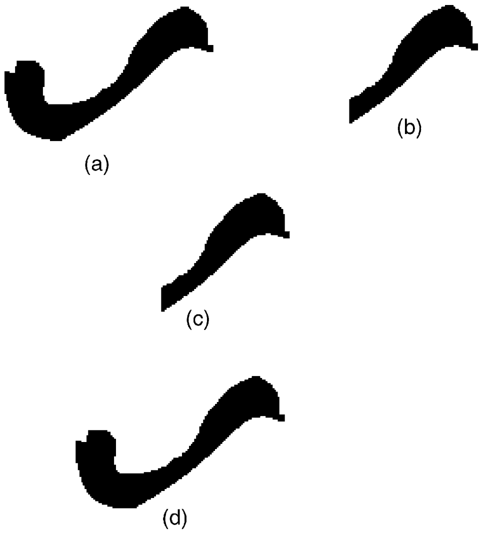

Fig. 10. Example 1 for irregular shaped images.

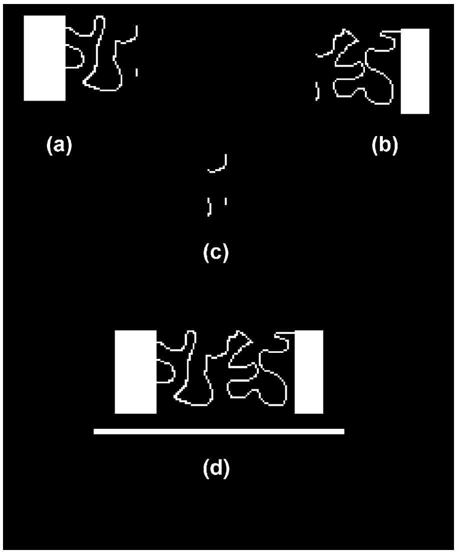

Fig. 11. Example 2 for irregular shaped images.

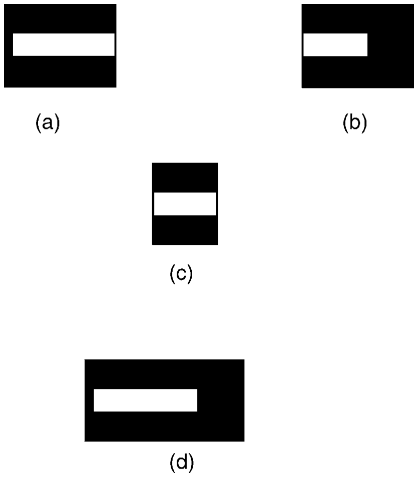

Fig. 12. Synthetic image which shows the algorithm 2 works when the split image 1 (a) contains more overlapping region than split image 2 (b).

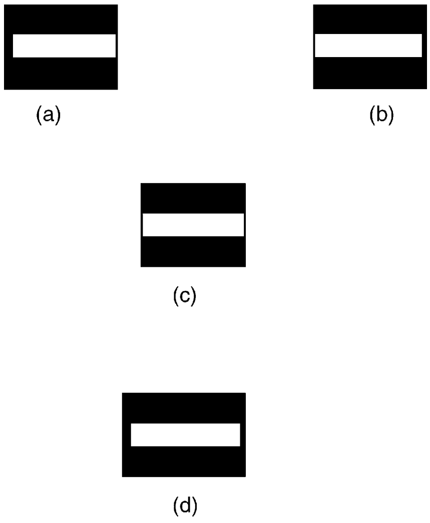

Fig. 13. Synthetic image which shows the algorithm 2 works when the split image 2 (b) contains more overlapping region than split image 1 (a).

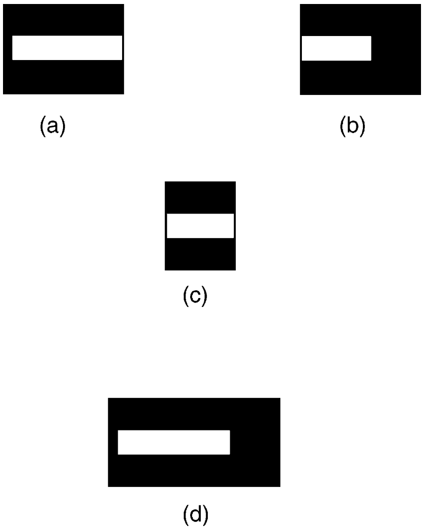

Fig. 14. Synthetic image which shows the algorithm 3 works when the split image 1 (a) contains more overlapping region than split image 2 (b).

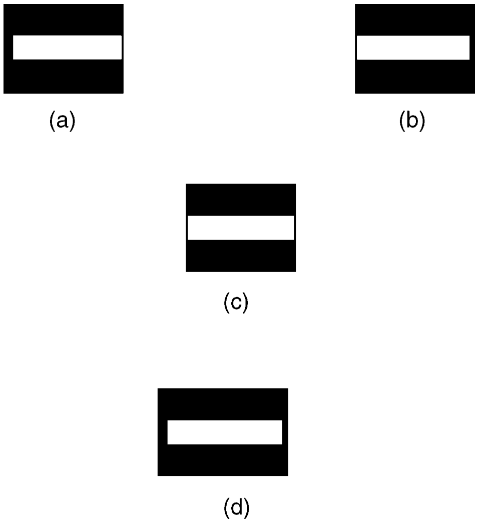

Fig. 15. Synthetic image which shows the algorithm 3 works when the split image 2 (b) contains more overlapping region than split image 1 (a).

5. Conclusion

In this paper, a new and simple approach for mosaicing of two split images of a large document to produce single and large document is presented. The proposed methods work based on sliding window concept which gives high level features instead pixel level features. The complexity of each algorithm is discussed. The proposed approach assumes that the overlapping region is present at the right and the left ends of the split image 1 and split image 2, respectively. In addition to this the method demands overlapping region at least size of window in the split images. However, the method works fine for all types of documents but it consumes time and fails if the sequence is missed. This is considered as our future work.

Acknowledgements

One of the authors (Mr Shivakumara P.) wishes to place on record his gratitude the fellowship sponsoring agency AICTE, vide sanction number F.No/8020/RID/R&D-50.2001-01, New Delhi, in pursuing his work.

References

[1] A.R. Zappala, A.H. Gee, M.J. Taylor, Document mosaicing. in: Proceedings of the British Machine Vision Conference, vol. 2, pp. 600-609, Colchester, 1997.

[2] S. Peleg, A. Gee, Haifa Research Laboratory, Virtual Cameras using Image mosaicing, Herbrew University, October 1997.

[3] A.P. Whichello, H. Yan, Document Image Mosaicing, Imaging Science and Engineering laboratory, Department of Electrical Engineering, University of Sydney, NSW 2006, (1997).

[4] P. Shivakumara, D.S. Guru, G. Hemantha Kumar, P. Nagabhushan, Document image mosaicing: a novel technique based on pattern matching approach, Proceedings of the National Conference on Recent Trends in Advanced Computing (NCRTAC-2001), Tamil Nadu, Feb 9–10, 2001, pp. 01–08.

[5] D.S. Shivakumara Guru, G. Hemantha Kumar, P. Nagabhushan, Pattern matching approach based image sequencing useful for document image mosaicing, Proceedings of the National Conference on Document Analysis and Recognition (NCDAR-2001), Mandya, Karnataka, July 13–14, 2001.

[6] D.S. Shivakumara Guru, G. Hemantha Kumar, P. Nagabhushan, Mosaicing of color documents: a technique based on pattern matching approach, Proceedings of National Conference on NCCIT, Kilakarai, Tamilnadu, 24th and 25th September, 2001,pp. 69–74.

[7] D.S. Shivakumara Guru, G. Hemantha Kumar, P. Nagabhushan, Mosaicing of scrolled split images based on pattern matching approach, Proceedings of Third National Conference on Recent Trends in Advanced Computing (NCRTAC - 2002), Tamil Nadu, Feb 13–15, 2002.

[8] P. Shivakumara, G. Hemantha Kumara, D.S. Guru, P. Nagabhushan, Document image mosaicing: a novel approach based on column block matching, Proceedings of National Workshop on Machine Intelligence using Soft Computing NWMISC—2003, Department of Computer Applications, RMK Engineering College, Kavaraipettai, Tamil Nadu, August 9th, 2003, p.16.

[9] Schutte K and Vossepoel A, Accurate mosaicing of scanned maps or how to generate a virtual A0 scanner, Proceedings of the first annual conference of the advanced school for computing and imaging, Heijen, the Netherlands, pp 353–359.