IJDAR (2002) 4: 183–190

International Journal on Document Analysis and Recognition

An empirical measure of the performance of a document image segmentation algorithm

Amit Kumar Das[1], **Sanjoy Kumar Saha**[1], **Bhabatosh Chanda**[2]

[1] Computer Science & Technology Department, Bengal Engineering College (DU), Sibpore, Howrah 711 103, India; e-mail: {amit,sks}@becs.ac.in
[2] Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta 700 035, India; e-mail: chanda@isical.ac.in

Received July 14, 2000 / Revised June 12, 2001

Abstract. Document image segmentation is the first step in document image analysis and understanding. One major problem centres on the performance analysis of the evolving segmentation algorithms. The use of a standard document database maintained at the Universities/Research Laboratories helps to solve the problem of getting authentic data sources and other information, but some methodologies have to be used for performance analysis of the segmentation. We describe a new document model in terms of a bounding box representation of its constituent parts and suggest an empirical measure of performance of a segmentation algorithm based on this new graph-like model of the document. Besides the global error measures, the proposed method also produces segment-wise details of common segmentation problems such as horizontal and vertical split and merge as well as invalid and mismatched regions.

Keywords: Document image analysis – Segmentation – Document image database – Document model – Performance analysis

1 Introduction

Document image segmentation is a part of document image analysis (DIA) which is concerned with the automatic interpretation of printed and handwritten documents including text, drawings, maps, etc. Many methodologies, using different approaches, exist for segmentation. Although fully automatic segmentation is yet to be achieved, the techniques are mature enough to support commercial OCR systems. In this evolving field new algorithms for segmentation are continuously being proposed by researchers. Systematic performance analysis is a must for any computer vision task [5] and a variety of methods for performance analysis related to DIA can be found in the literature [11,9,2,1]. However, considering the diversity of the methods and wide variety of goals the performance analysis of the segmentation algorithms is not easy. The effectiveness of the segmentation algorithm is typically assessed by executing the algorithm over a large number of document pages whose ground-truth should either be available or created. Benchmarking complexity increases as the segmentation errors may be handled in various ways. If the segmentation errors are to be corrected manually we may evaluate the performance in terms of the editing cost of corrective measures such as block move, and insertion and deletion operations [7, 9]. For feeding the segmentation output directly to the OCR systems, the measure should be in terms of type mismatch, i.e., the amount of images that are classified as text or graphics or vice versa [12,2].

Normally segmentation algorithms are tested on sample images on an ad hoc basis and in most cases performance drastically varies with the variation in the test images. To alleviate this problem some image database created and maintained in Universities/Research Laboratories [8] may be used as a standard agreed-upon sample set for testing. We assume that such a document database will be used as a sample data set to circumvent the problem of getting a wide variety in the input data and to substantiate the claims of the performance of the segmentation algorithms. Even with the use of such a database where ground-truthed information is available for verification, the performance analysis is still a problem due to the lack of:

- A suitable document page representation scheme.
- Appropriate global error measures.

As a result it is difficult to analyse the performance of the segmentation algorithm. Our objective is to evaluate the performance of document image segmentation algorithms by providing a new document model and a method that may be used to compute global errors. We confine our domain of application to well-structured document pages such as technical journals and reports where entities (text, graphics, etc.) are bounded by rectangular zones.

Correspondence to: A.K. Das

1.1 Review of past work

Two basic approaches are adopted for automatic evaluation of document image segmentation algorithms: text-based and region-based. Among these, we briefly discuss only two widely known methods – one from each approach.

Text-based method [7]: In the text-based method, page segmentation of the OCR system is applied to the document and the result is available as an ASCII string. A string matching algorithm is then used to compute the number of insertions, deletions, and block moves necessary to convert this output into ideal ground truth string with a minimum cost. The total cost (TC) is usually the weighted sum of the operations as given by:

$$TC = BM \times W_{bm} + IN \times W_{in} + DL \times W_{dl}$$

where BM, IN, and DL are the number of block moves, insertions, and deletions operations, respectively, for the page, and W_{bm}, W_{in}, and W_{dl} are the costs associated with each operations.

Except for the string matching algorithm, the text-based approach is quite simple. However, there are some problems as identified below:

- The segmentation (or zoning) algorithm should be a part of the OCR system; so stand-alone algorithms for performance analysis are not suited for benchmarking.
- Measured accuracy is given in terms of text regions only; a split in the half-tone or a merge of graphics and tables is not considered at all.
- The output only gives a numerical account of moves, inserts, etc., required to convert the extracted string into the equivalent to the original text, and provides no information regarding the types of actual mistakes; namely, split, merge, region mismatch, etc.

For these reasons the text-based approaches are no longer popular with the OCR community.

Region-based method [12]: Here the performance of segmentation is calculated by comparing the segmentation output and the corresponding ground truth which are both ASCII files describing the regions on the page. Logically, the method proceeds by examining the segmented windows (regions) through ground-truthed windows (regions) to compare and compute match, overlap, split, and merge.

A region map is used to evaluate segmentation performance. Each region map is a reduced-resolution representation of the original document image in which each pixel is tagged to indicate the region it belongs to. Region correspondence is then computed by using region maps for each pair of $S \times G$ (i.e., Segmented region and ground-truthed region, respectively) whose bounding boxes overlap.

Region-based approaches are more appropriate for performance evaluation of stand-alone segmentation algorithms. However, the method proposed by Yanikoglu and Vincent [12] is based on an ad hoc approach involving the direct overlap of ground-truth images and segmented output and has the following problems:

- It produces no global error measures.
- Generation of region maps is cumbersome and time consuming.
- No document model is used. As a result, the region overlap computation blindly compares each segmented region and every other ground-truth region.
- It also ignores region type mismatch error which is important when segmentation output is directly fed to OCR systems.

The method we propose here is also a region-based approach which assumes that a document page is composed of various unique entities such as text, graphics, tables, half-tone, headings, etc. [6] and that each unique entity can be enclosed by an upright rectangle or bounding box [3]. This confines its applicability to well-structured document page images, similar to those available in UW-I and UW-II. In the proposed algorithm we improve upon all the drawbacks of [12] as stated above; moreover, our algorithm is faster because we the compute block area of a region instead of counting individual common on-pixels.

As stated earlier, schemes to represent regions in a document page are not standardised. They are usually represented as rectangles, piecewise rectangles, polygons, and nested rectangles. Regions also have attributes such as types, subtypes, parent zone, etc. In order to standardise, we first propose a new graph-like model of the document image based on the bounding box representation of the constituent parts of the document preserving their physical layout order.

1.2 Document page representation

A document page D may be represented as a 6 tuple

$$D = (E_1,\ E_2,\ E_3,\ E_4,\ E_5,\ E_6)$$

where E_i's are entities such as text, tables, headings, graphics, half-tones, and displayed math zones. Each entity, in turn, represents a collection of rectangular regions. Suppose, $E_1 = \{t_i,\ \phi\}$, $E_2 = \{T_i,\ \phi\}$, $E_3 = \{H_i,\ \phi\}$, $E_4 = \{g_i,\ \phi\}$, $E_5 = \{h_i,\ \phi\}$, and $E_6 = \{m_i,\ \phi\}$, where t_i, T_i, H_i, g_i, h_i, and m_i are rectangular regions containing only text, tables, headings, graphics, half-tones, and math regions, respectively. Thus each entity has a unique property denoted by Prop.(E_i) for $i = 1, \ldots, 6$. Therefore, the task of document page image segmentation is to extract a set of, say, n bounding boxes Bb_j $(j = 1, \ldots, n)$ from the image domain X such that:

(i) $\bigcup_{j=1}^{n} Bb_j \subseteq X$.
(ii) $Bb_j \bigcap Bb_k = \phi$ where $j \neq k$.
(iii) For every j there exists one and only one i such that Prop.(Bb_j) = Prop.(E_i).
(iv) $B = X \setminus \bigcup_{j=1}^{n} Bb_j$ is called background and Prop.(B) $\neq$ Prop.(E_i) for $i = 1, \ldots, 6$.

Throughout our discussion by the term bounding box we mean an upright rectangle containing a single entity, and property means the content type (e.g., Text) of a bounding box.

Here we suggest a graphical model of the document image that specifies how the constituent parts are arranged in the document, their general characteristics and inter-relationship. The model primarily describes an ideal, noise-free document image.

We summarize the advantages of the proposed approach as follows:

(i) Graphical representation of document page image.
(ii) No need to create cumbersome region maps and an on-pixel count as done in [12].
(iii) Minimum recursion/iterations to find the match as the graph representation preserves the layout order.
(iv) Detailed benchmarking output including global error measures and segment-wise details of all sorts of errors such as merge, split, etc.

This paper is organised as follows: Section 2 gives an example document image database based on which the proposed performance evaluation method is devised. The proposed model and error measures are given in Sects. 3 and 4, respectively. Experimental results as well as concluding remarks are cited in Sect. 5.

2 An example image database: UW-I and UW-II

We assume that the document image segmentation algorithms will be tested using one of the existing image databases available for that purpose. The database developed at the Intelligent Systems Laboratory of the University of Washington [8] may be referred to as an example. The University of Washington's English Document Database (UW-I and UW-II) is intended for OCR and DIA applications with the following objective:

- To provide a substantially sized training and testing database for algorithm development.
- To provide an accurate ground-truthed document database for testing the algorithm.
- To estimate performance of vendor-proposed systems by the OCR customers.

The database consists of document pages from technical journals and contains both bi-level and gray images. In addition to the images the database consists of substantial qualitative and quantitative information on each document page.

Below we name some of the files and the information contained therein:

Page condition file: registers skew angle, presence of noise, etc.
Page attribute file: describes dominant font, figures present, etc.
Page bounding box file: contains the location and size of broad zones in a page such as *header*, *footer*, and *live matter*, etc.
Zone bounding box file: is a finer-level description of zones in terms of location, size, and contents (e.g., text, tables, half-tones, etc.).
Zone attribute file: contains the semantic meaning of each zone; dominant font in a zone, etc.
Ground truth files: are given for text zone; the contents of each zone are described in terms of ASCII text. In addition to this, UW-III contains ground-truth for the displayed math zone and graphics zone.

In short, the database has page images and a complete description of various zones to serve our intended objective.

3 Proposed document image model

In this section we specify a document image model based on the bounding boxes of its constituent parts for performance evaluation of segmentation algorithms. This model is a directed graph constructed using bounding box information available from the 'zone bounding box' and 'zone attribute' files.

Bounding boxes are usually represented by coordinates of diagonally opposite corner points (B_{i1} and B_{i2}). An example of such bounding boxes of five distinct zones is shown in Fig. 1a. First, we derive two different bounding boxes bb_i and BB_i (as shown in Fig. 1b) from the bounding box information available from the UW database. bb_i is the smallest bounding box containing one of the components such as texts, tables, graphics, headings, half-tones or math-zones, and BB_i is the largest bounding box containing bb_i and some part of the background (B) but not bb_j (where $i \neq j$). Thus, BB_i has the following properties:

(i) $BB_i \supseteq bb_i$.
(ii) $BB_i \bigcap bb_j = \phi$ for $i \neq j$.

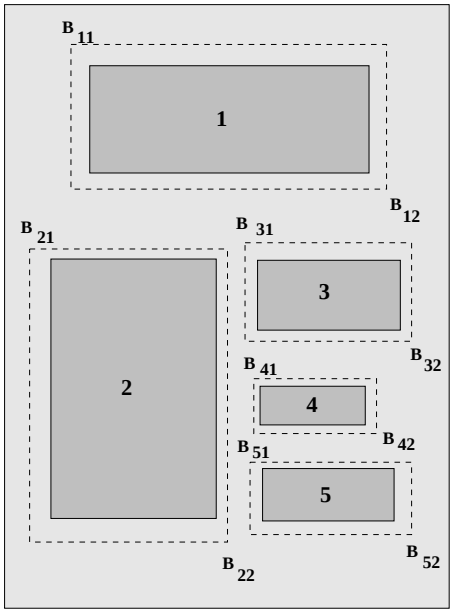
bb_{i1} and bb_{i2} represent top-left and bottom-right corners of bb_i. Similarly BB_{i1} and BB_{i2} represent top-left and bottom-right corners of BB_i [4].

The next thing we require to construct the graph is the relative location of a bounding box with respect to the others in a document page. With respect to a given bounding box (see Fig. 2a) we define two zones: the right zone and the bottom zone as follows:

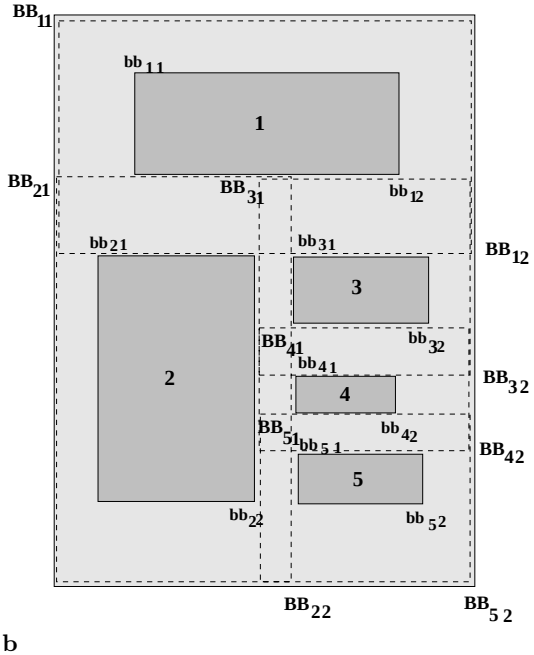
Definition: *Right zone* of a bounding box designated by corner points $\{(r_{min},\ c_{min}),\ (r_{max},\ c_{max})\}$ is a half-plane in $(r,\ c)$-space where $c > c_{max}$.

Definition: *Bottom zone* of a bounding box designated by corner points $\{(r_{min},\ c_{min}),\ (r_{max},\ c_{max})\}$ is a half-plane in $(r,\ c)$-space where $r > r_{max}$.

Note that the intersection of the right zone and bottom zone is non-empty. However, a bounding box, being convex, must be contained completely within either the bottom zone or the right zone of another bounding box, if any exists. We call the latter one as "top-of" or "left-of" the former one providing no other bounding box lies between them.



a



b

Fig. 1a,b. Bounding box representation of constituent parts of a document page. **a** Bounding box information usually available in the UW document image database; **b** the smallest and largest bounding boxes derived from B_{i1} and B_{i2}

Suppose there are three bounding boxes B_1, B_2 and B_3 designated by their corner points and the second and third bounding boxes are in the bottom zone of the first bounding as shown in Fig. 2b. Then the third box is said to lie between the first and second if

(i) $[c_{1min}, c_{1max}] \bigcap [c_{2min}, c_{2max}] \bigcap [c_{3min}, c_{3max}] \neq \phi$ and

(ii) $[r_{1max}, r_{2min}] \bigcap [r_{3min}, r_{3max}] \neq \phi$

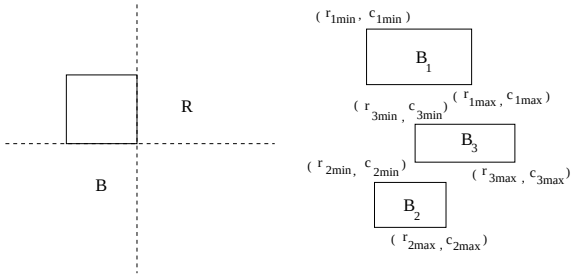


Fig. 2a,b. Relative location of bounding boxes. **a** Right and bottom zone; **b** relative position of a bounding box with respect to others

If the second and third bounding boxes are in right zone of the first one, the corresponding conditions are:

(i) $[r_{1min}, r_{1max}] \bigcap [r_{2min}, r_{2max}] \bigcap [r_{3min}, r_{3max}] \neq \phi$ and

(ii) $[c_{1max}, c_{2min}] \bigcap [c_{3min}, c_{3max}] \neq \phi$

The layout features of document page image can thus be represented by a directed graph whose nodes represent various zones, and edges represent their relative locations. Figure 3a shows the graph representation of the example page shown in Fig. 1. Suppose we call it the model graph of the page. Each node M_i of the graph has the following attributes:

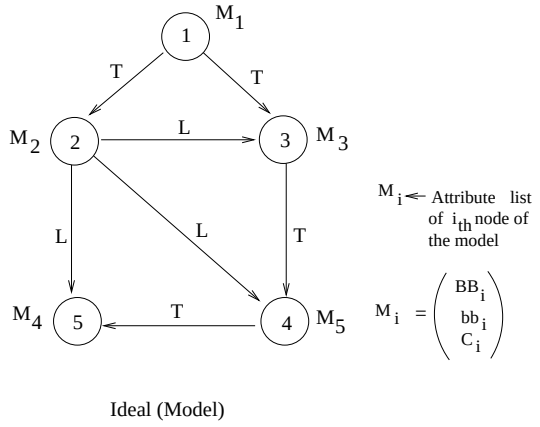
- Smallest bounding box (denoted by bb_i).
- Largest bounding box (denoted by BB_i).
- Type of the content (denoted by C_i).

Note that the edges connecting the nodes also carry relative location information. The notations T and L indicate that a node is either sitting at the top-of or left-of another node joined by that particular directed edge. Note that we assume the scanning direction to be from top to bottom and left to right.

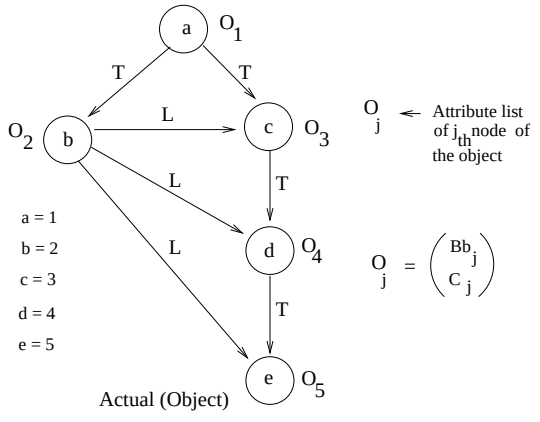
A similar graph can be generated from the segmented image. Let us call this graph the actual or object graph where the nodes are denoted by O_j. Attributes are denoted by Bb_j, the bounding box, and C_j, the content type. An example of such a graph is shown in Fig. 3b. These two attributed graphs, i.e., model and object, are compared to evaluate the segmentation algorithm as presented in the next section.

4 Proposed measure of performance

We propose a performance measure of the segmentation algorithms by using a graph representation of the document page, in terms of the bounding boxes. For performance analysis, two graphs may be matched based on node types and relative location associated with edges using graph isomorphism algorithms. As the number of nodes is very small (in most cases the number is less than ten), we implement the following simple algorithm to match the nodes explicitly.



a



b

Fig. 3a,b. Graphical representation of the page shown in Fig. 1 in the bounding box form. **a** Model graph of the page and **b** actual (object) graph of the page obtained after segmentation

Algorithm for node matching

Assumptions: both model and object graphs are directed acyclic graph. Let us represent the model graph by

$$G_M = (V_M,\ E_M)$$

where $V_M = \{M_0,\ M_1,\ M_2, \ldots,\ M_n\}$ represents the set of nodes or vertices, and E_M represents an edge set defining a binary relation on V. Thus, $E_M : (M_i,\ M_j) = W_{ij}$ where $W_{ij} \in \{L, T, \phi\}$. An attribute list $(BB_i,\ bb_i,\ C_i)$ is attached to each M_i. Similarly, the object graph may be represented by

$$G_O = (V_O,\ E_O)$$

where $V_O = \{O_0,\ O_1,\ O_2, \ldots,\ O_k\}$ and $E_O : (O_i,\ O_j) = W'_{ij}$; where $W'_{ij} \in \{L, T, \phi\}$. An attribute list $(Bb_i,\ C_i)$ is attached to each O_i. Suppose for both the graph the topmost node is the root node whose in degree is zero.

Algorithm:

```
for i = 0, 1, 2, ..., k {
    for j = 0, 1, 2, ..., n {
        if #(Bbi) = #(Bbi ∩ BBj)
        { /* #(A) indicates area of A */
          store (Oi ≡ Mj);
          break;
        }
        else if #(Bbi ∩ bbj) > 0 {
          store (Oi ≡ Mj);
          form a stack with the nodes Mj'
          such that Wjj' ≠ φ ;
          while (stack != null) {
                pop a member Ml from stack;
                if #(Bbi ∩ bbl) > 0 {
                  store (Oi ≡ Ml);
                  push the nodes Ml' such that Wll' ≠ φ;
                }
          }
          break;
        }
    }
}
```

Once the graphs are matched to the best possible extent, the similarity of the extracted regions to that of the model is computed. It should be noted that four different situations may occur as stated below:

i) The object node does not match with any node of the model – clearly an invalid segment and it is deleted immediately.
ii) One node of the object graph matches with one and only one node of the model graph (exact matching).
iii) More than one node of the object graph matches with one node of the model graph (splitting).
iv) One node of the object graph matches with more than one node of the model graph (merging).

However, for each pair of objects and model nodes that are matched we need to calculate the error as follows.

Suppose, due to matching of the graphs, node M_i of the model graph corresponds to node O_j of the object graph. The error measure due to this matching takes into account the commonly used metric used in pattern recognition such as:

(i) Correct classification $cc_{ji} = \#(Bb_j \bigcap BB_i)$.
(ii) False alarm $ef_j = \#(Bb_j \setminus BB_i)$.
(iii) Mis-classification $em_j = \#(bb_i \setminus Bb_j)$.

An example illustrating these measures is shown in Fig. 4. These are computed for all the object nodes O_j. Note that in case of merging $BB_i = BB_k \bigcup BB_l \bigcup \ldots$ and $bb_i = bb_k \bigcup bb_l \bigcup \ldots$, where M_k, M_l, ..., etc. merge to O_j and $C_j = C_k = C_l \ldots$. The correctness of segmented region corresponding to the node O_j, denoted by S_j, may be defined as

$$S_j = \begin{cases} cc_{ji} - ef_j - em_j & ; \text{ if } C_i = C_j \\ 0 & ; \text{ otherwise} \end{cases}$$

Note that value of correctness is non-negative only when types of corresponding regions are the same. Hence,

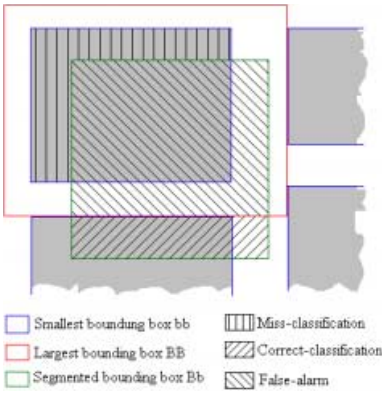


Fig. 4. Correctly classified area and errors in segmentation

the correctness of the segmentation algorithm is given by

$$S = \sum_j S_j$$

and finally, the overall %error for a given page is defined as:

$$E = \left\{ 1 - S / \sum_j \#(Bb_j) \right\} \times 100\%$$

Proper benchmarking needs this error be computed for a large data set and an average would reflect the performance of the segmentation algorithm. It is evident that the percentage error measure for individual items such as correct classification, false alarms, and mis-classification can also be obtained by a slight modification of the above equation. Thus, the respective expressions are omitted in this presentation. For a more in-depth analysis of the system performance, especially during the development and tuning, one needs to consider which entity is classified or confused as which entity. This kind of information can also be derived from the proposed model.

Suppose we would like to evaluate the % error incurred due to the segmentation of 'table' regions as 'Text' regions. Let us define

$$A_i = \begin{cases} \#(Bb_j \bigcup bb_i) & ; \quad \text{if } C_i = T \text{ and } C_j = t \\ 0 & ; \quad \text{otherwise} \end{cases}$$

and

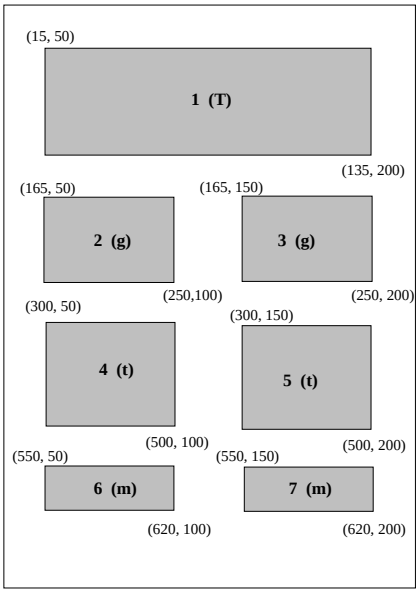
$$R_i = \begin{cases} \#(Bb_i) & ; \quad \text{if } C_i = T \\ 0 & ; \quad \text{otherwise} \end{cases}$$

Finally, the measure is

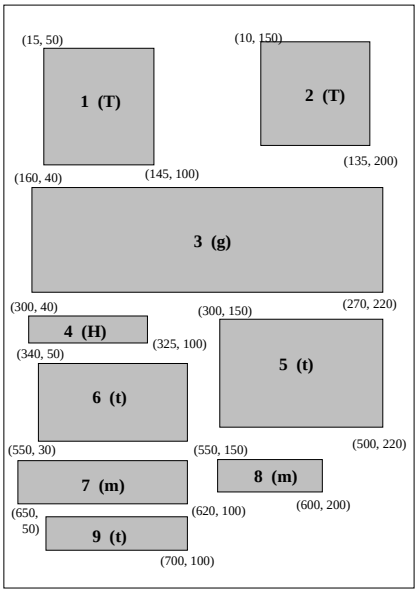
$$E(T,t) = \sum_i A_i / \sum_i R_i$$

This may be defined for other similar or different types of entities.

The bounding box and type information for the model graph is available from the zone bounding box file and zone attribute file for each image in the database [8,10]. Similar information for the object graph is extracted from the result of the document page segmentation algorithm.



a



b

Fig. 5. Bounding box representation of constituent parts of a document page

Segment-wise results: the proposed method can produce a detailed report on each segmented zone regarding the common segmentation problems such as split, merge, etc. This is shown with the help of Fig. 5a and 5b which are ground-truth regions and segmented regions of an example document page. The tabular output (Table 1) shows the region-wise details of the segmentation including invalid segment, matched regions as well as split and merge.

Table 1. Region-wise matching data

Ground-truth region no.	Invalid Segmented Region No.
None	9 (650, 50 – 700, 100)

Ground-truth Region No.	Matched portion (Segmented region)	Type Mismatch	Missed portion
1	1 (15, 50 – 135, 100) 2 (15, 150 – 135, 200)	No	(15, 101 – 135, 149)
2	3 (165, 50 – 250, 100)	No	None
3	3 (165, 150 – 250, 200)	No	None
4	4 (300, 50 – 325, 100) 6 (340, 50 – 500, 100)	No Yes	(326, 50 – 339, 100) No
5	5 (300, 150 – 500, 200)	No	No
6	7 (550, 50 – 620, 100)	No	No
7	8 (550, 150 – 600, 200)	No	(601, 150 – 620, 200)

Ground-truth Region No.	Split Regions (In terms of segmented regions)
1	Horizontal Split: 1, 2
4	Vertical Split: 4, 6

Segmented Region No.	Merge Regions (In terms of Ground-truth regions)
3	Horizontal Merge: 2, 3

5 Experimental results and conclusion

We have used the proposed performance evaluation method on the results obtained by the segmentation algorithm described in [3]. The test data consist of 118 document pages mostly from the UW-I and UW-II image database and the rest were scanned by us. The document page segmentation algorithm mentioned here considers only five different entities: text, graphics, tables, headings, and half-tone. Math-zones, if any, are segmented as text zones. The result is summarized as a confusion matrix $[C_{i,j}]$ as shown in Table 2. The matrix C has six rows and five columns, where columns indicate the actual (i.e., ground-truthed entities) and the rows the segmented (i.e., identified) ones. Hence, $C_{i,i}$ $(i = 0, \ldots, 4)$ denotes correct classification and $C_{i,j}$ $(i = 0, \ldots, 4$ and $j = 0, \ldots, 4$, but $i \neq j)$ denotes a false alarm. Mis-classification is denoted by $C_{5,j}$ for $j = 0, \ldots, 4$.

Not all the entities are equi-probable in the data set. Based on the ground-truth their probability of occupying a unit area of the document page are computed. These values of P_js are given in Table 3.

Using Table 2 and Table 3, different measures are given by

$$\text{Correct classification} = \sum_{j=0}^{4} P_j C_{j,j}$$

$$\text{False alarm} = \sum_{i=0}^{4} \sum_{j=0}^{4} \sum_{j \neq i} P_j C_{i,j}$$

Table 2. Performance (in %) of segmentation algorithm for different entities

		Original (Ground-truth)				
		Text	Graphics	Table	Heading	Half-tone
S	Text	93.05	1.16	3.44	1.42	
e	Graphics	1.11	88.43	4.09	2.73	3.62
g	Table	3.70	2.75	90.28	2.28	
m	Heading	2.14			86.35	
e	Half-tone		4.94		7.22	95.17
n						
t	Mis-classifi-		2.72	2.19		1.21
e	cation					
d						

Table 3. Probabilities of entities in the test data set

	Text	Graphics	Table	Heading	Half-tone
Probability	0.623	0.158	0.026	0.019	0.174

Table 4. Measure (in %) of different components

Correct classification	92.49
False alarm	6.81
Mis-classification	0.70

$$\text{Mis} - \text{classification} = \sum_{j=0}^{4} P_j C_{5,j}$$

The values are shown in Table 4.

Hence, the overall performance in terms of error is given by E = 15.02 %.

In this paper we have presented a methodology for measuring the performance of document segmentation algorithms with the assumption that they would be tested with an image database such as [8,10] where the required ground-truth is available. The performance measure is based on a new graph-like model of a document page based on the relative locations of the constituent parts of the document image. The proposed measure of performance evaluation would solve the problem of evaluation of segmentation algorithms producing results in terms of rectangular bounding boxes. Right now it does not support segmentation in terms of bounding polygons suitable for complex documents. In a complex document every object cannot be encompassed by a rectangular bounding box though an upright bounding box is an ideal representation for different entities in the majority of these documents and the scheme may be extended as a piecewise rectangle can approximate arbitrary polygons. The graph matching could be carried out more elegantly and efficiently based on edge information. However, we did not place emphasis on this because the number of nodes are small.

References

1. Ecvnet benchmarking and performnace evaluation website: http://pandora.imag.fr/ecvnet/benchmarking.html
2. S. Chen, S. Subramanium, R.M. Haralick, I.T. Phillips: Performance evaluation of two OCR systems. In: 3rd Ann. Symp. on Document Analysis and Inf. Retrieval, pp. 299–317, Univ. of Nevada, Las Vegas, USA, April 11–13, (1994)
3. A.K. Das: Document image segmentation : a morphological approach. PhD thesis, Computer Science and Technology Department, Bengal Engineering College (D.U.), Sibpore, Howrah, India, (1998)
4. A.K. Das, B. Chanda: A new document model for performance analysis of segmentation. In: 4th Int. Conf. on advances in pattern recognition and digital techniques (ICPARDT-1999), Dec. 27–29, Calcutta, India, pp. 210–214, (1999)
5. R.M. Haralick: Performance assessment of near perfect machines. Mach. Vision Appl. 2:1–16, (1989)
6. A.K. Jain, Bin Yu: Page segmentation using document model. In: Proc. ICDAR, Ulm, Germany, pp. 34–38, August (1997)
7. J. Kanai, T.A. Nartker, S.V. Rice, G. Nagy: Automatic evaluation of OCR zoning. IEEE Trans. PAMI, 17(1):86–90, (1995)
8. I.T. Phillips, S. Chen, R.M. Haralick: CD-ROM document database standard. In: ICDAR93, pp. 478–483, (1993)
9. I.T. Phillips, J. Liang, A.K. Chhabra, R.M. Haralick: A performance evaluation protocol for graphics recognition systems, In: K. Tombre, A. K. Chhabra (Eds.) Graphics recognition: algorithms and systems, Lecture Notes in Computer Science, vol. 1389. Springer, Berlin Heidelberg New York, 1998, pp. 372–389
10. R.P. Rogers, I.T. Phillips, R.M. Haralick: Semiautomatic production of highly accurate word bounding box ground truth. In: DAS, pp. 375–387, (1996)
11. M. Thulke, V. Mergner, A. Dengel: Quality evaluation of document segmentation results. In: ICDAR'99, p. 450, Sept. 20–22, Bangalore, India (1999)
12. B.A. Yanikoglu, L. Vincent: Pink panther: a complete environment for ground-truthing and benchmarking document page segmentation. Pattern Recognition, 31:1191–1204, (1998)

Amit Kumar Das (1957) was born in Calcutta and completed his Masters and PhD degree at Calcutta University and Bengal Engineering College, respectively. He is presently an Assistant Professor in the Computer Science and Technology Department, Bengal Engineering College (D.U). His field of interest includes embedded system design and document image processing.

Sanjoy Kumar Saha (1968) receives his B.E. and M.E. Degree in Electronics and Telecommunication from Jadavpur University, Calcutta in the years 1990 and 1992, respectively. He worked in different positions in a number of Industries/Institutes and finally joined the Computer Science Department, Bengal Engineering College (DU) in 1998 as a lecturer. He has published some papers and his research interest is in Image Processing including document image analysis and content-based image retrieval.

Bhabatosh Chanda born in 1957. Received B.E. in Electronics and Telecommunication Engineering and PhD in Electrical Engineering from the University of Calcutta in 1979 and 1988, respectively. Received "Young Scientist Medal" of Indian National Science Academy in 1989 and "Computer Engineering Division Medal" of the Institution of Engineers (India) in 1998. He has also been the recipient of a UN fellowship, UNESCO-INRIA fellowship, and fellowship of National Academy of Science, India during his career. He worked at the Intelligent System lab, University of Washington, Seattle, USA as a visiting faculty member from 1995 to 1996. He has published more than 50 technical articles and a textbook on image processing. His research interest includes Image Processing, Pattern Recognition, Computer Vision, and Mathematical Morphology. Currently working as Professor in Indian Statistical Institute, Calcutta, India.