Document Ranking by Layout Relevance

May Huang, Daniel DeMenthon, David Doermann
Language and Media Processing Laboratory (LAMP)
University of Maryland
College Park, MD 20742

Lynn Golebiowski
Booz Allen Hamilton
134 National Business Parkway,
Annapolis Junction, MD 20701

Abstract

This paper describes the development of a new document ranking system based on layout similarity. The user has a need represented by a set of "wanted" documents, and the system ranks documents in the collection according to this need. Rather than performing complete document analysis, the system extracts text lines, and models layouts as relationships between pairs of these lines. This paper explores three novel feature sets to support scoring in large document collections. First, pairs of lines are used to form quadrilaterals, which are represented by their turning functions. A non-Euclidean distance is used to measure similarity. Second, the quadrilaterals are represented by 5D Euclidean vectors, and third, each line is represented by a 5D Euclidean vector. We compare the classification performance and computation speed of these three feature sets using a large database of diverse documents including forms, academic papers and handwritten pages in English and Arabic. The approach using quadrilaterals and turning functions produces slightly better results, but the approach using vectors to represent text lines is much faster for large document databases.

1. Introduction

Searching through very large collections of scanned material is an immediate need both in the government and commercial sectors. For example, in the discovery phase of litigation cases, companies are often asked to provide all the documents they produced in the course of decades. Tons of boxes of documents are shipped and scanned, and the resulting databases can be huge. For instance, the Minnesota Tobacco Document Depository contains over 35 million pages of material discovered in the course of the litigation against the tobacco industry. Although content search is useful and can be provided with the help of OCR, there is also a need in these situations for search based on document layout. Similar layouts are often indicative of a particular source — deposit forms for a given bank, letterhead from a given company, etc. While using other search capabilities, or in the process of examining the collection, a user may find a document (form, invoice, letter, etc) with a layout of particular interest. The user would show the system one or more examples of documents with the layout of interest, and from these examples the system scores the documents and returns a list of document images with the most relevant on top, as would an internet search engine. A system based on layout is robust even when OCR fails to decode the content, as is the case with handwritten documents or old photocopies.

1.1. Related Work

There has been limited work on retrieval using document layout similarity. Most of the work has focused on logical labeling (e.g. [3, 8]) or on genre specific classification [11]. In [4], a strategy is described which focuses on feature selection to optimize the separation of different classes for routing, but the feature sets used are not described. In [7], interval encoding is proposed to classify structured documents using segmented text blocks. The approach will have limited success for highly varying or noisy documents. [5] contains a survey of the literature on document image retrieval that highlights some non-traditional approaches to document image retrieval.

1.2. Approaches

This paper examines three algorithmic solutions for ranking documents and compares their performance. One of the solutions studied here builds upon prior work by one of the current authors [6]. This solution consists of detecting text lines, then considering as objects describing the page layout the quadrilaterals generated by all pairs of lines. In order to compare page layouts, quadrilaterals from new documents and documents in training sets need to be compared. Arkin's distance measure [1] was used. This distance was also used for clustering the quadrilaterals, so that the subsequent comparisons were only applied to cluster centers in order to increasing computation speed. This approach gives surprisingly good results (see Section 6). The rationale behind the idea of producing N^2 quadrilateral objects from the initial N text line objects is that in this expanded representation the configuration of every text line is expressed with respect to every other text line by the shape of a quadrilateral without the need for a frame of reference which may be difficult to find reliably when text scanning is skewed. The drawbacks are that one ends up with many more objects than one started with, and that efficient algorithms for

clustering, nearest neighbor and range search are more difficult to implement with Arkin's distance.

Therefore, the goals of the research described in this paper were to evaluate the performance of Euclidean descriptions of quadrilaterals, and also to find out if more concise layout representations by Euclidean descriptions of single text lines would provide competitive performance in spite of higher sensitivity to document skew and translation. As with our prior work [6], we quickly discard without further comparison potential matchings that have fonts of very different heights. One difference is that we focus on purely geometric aspects and do not use any text script information in the training and ranking procedures.

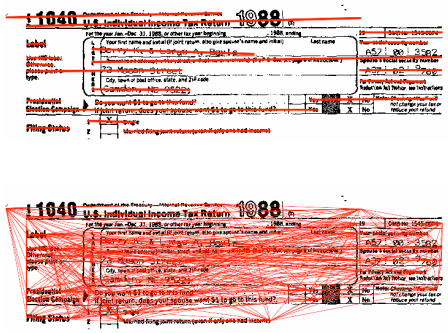

Figure 1: *Top*: Text lines detected in tax form by grouping connected components of black pixels. *B*ottom: Set of quadrilaterals formed by considering all pairs of text lines.

2. Ranking Procedure

Each document is subjected to the following processing steps:

1. Find text lines (by grouping connected components, see Figure 1, top) and de-skew the text.
2. Generate quadrilateral objects composed of all pairs of lines (Figure 1, bottom) or lines paired with the top edge of the bounding box (Section 3.3).
3. Cluster these objects and find cluster centers.

Then we apply the following steps to rank documents:

1. For each of the documents shown as examples of wanted documents, store objects that are cluster centers into a database with a "wanted" label.
2. For each of the documents shown in a training set of unwanted documents, store objects that are cluster centers into the database with an "unwanted" label.
3. For each document of the set that needs to be ranked, extract its lines and related objects, cluster them, and score each cluster center by looking at its neighbors in the database of wanted objects. Incorporate in the score the presence of neighbors in the database of unwanted objects. Then obtain a score for the document by combining the scores of each of its objects.
4. Present the documents as a ranked list.

3. Feature Representations

3.1. Line Pairs and Arkin's Distance

One can describe the shape of a polygon concisely by providing its *turning function* $\Theta(s)$, which measures the angle of the edge along the shape as a function of the normalized arc length [1]. The distance between two quadrilaterals can be computed as the minimum of an integral between their turning functions when the arc length origins are shifted with respect to each other. Arkin et al. [1] show that this distance is a metric. We applied the C implementation of this method written by Ressler [10]. Since this distance measure is scale independent, additional tests are needed in our application: distances are computed only if the areas of the two quadrilaterals are within 90% of each other. We have evaluated two methods for clustering quadrilaterals within each document when distances were computed with Arkin's method:

Method 1: Randomly select an object that has not been clustered yet and make it the center of a new cluster. All the objects that are within a fixed given range of this center are taken as members of this new cluster.

Method 2: After finding the members of a new cluster using Method 1, (1) reassign the cluster center as the object that has the minimum sum of distances to all cluster members, then (2) reassign the cluster members as those objects that are within range of this new center. These two steps are repeated over several iterations (we applied two iterations).

Method 2 provided significant ranking improvements and therefore the experiments discussed in this paper were conducted only with Method 2. However, the computational cost of our implementation is of order $O(N^2)$ in the number of quadrilaterals in a page, and explains the long computation time shown at the bottom of the first table of Figure 5. We will explore the use of efficient range search in Arkin's metric space in future work.

3.2. Euclidean Quadrilaterals

Clearly, there is a penalty in implementation complexity with the representation of quadrilaterals by turning functions. Therefore, for comparison we represented quadrilateral shapes using Euclidean components. A representation where some of the components were lengths and others were angles (Figure 2, left) was abandoned because it was not competitive with the turning function representation. We think this is due to the difficulty in finding a universal weighting between the angle components and the length components. We then turned to a representation with five lengths used as components which does not have this problem (Figure 2, right). An Euclidean representation of quadrilateral shape provides significant advantages; for clustering quadrilaterals in each document image, we used k-means clustering which has a complexity $O(N)$ in the number of objects; also, we applied an efficient implementation of range search using k-D trees [2] for scoring

documents. This new approach is competitive with the representation using a turning function (see Section 6).

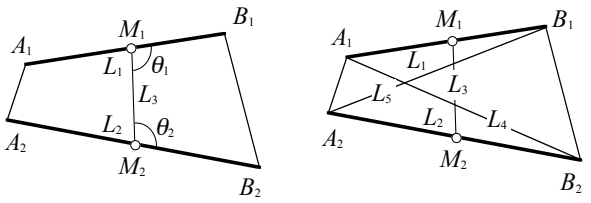

Figure 2: Two methods for defining five components that completely specify the shape of a quadrilateral and can be used for an Euclidean representation. Left: three lengths and two angles. Right: five lengths. This second method is much preferrable because all the components are of the same type and issues of appropriate weighting of components representing different concepts are avoided.

3.3. Documents as Line Sets

Even after achieving the great simplification of using Euclidean representations of quadrilaterals, there may still be an advantage to *not* transforming the M text lines into M^2 quadrilaterals, and instead representing documents with text lines. To find out, we turned to Euclidean representations of text lines with respect to the de-skewed text bounding box. We represented each individual text line by the quadrilateral formed by the top edge of the text bounding box and the text line. We described each quadrilateral with the same five length components described in Section 3.2. As we will see, this provided a factor of 50 speed increase while ranking performance remained competitive for our test set.

4. Document Scoring

To summarize, we represent document layout either using pairs of text lines described by turning functions, pairs of text lines described by vectors with five components, or text lines paired with the top edge of the text bounding box, also described by vectors of five length components. We then applied the same scoring principles using the cluster centers of each of these three representations as objects. Clearly, the score of a document should be some increasing function of the scores of its objects. We define the document score as the sum of the scores of its objects. The score of an object should depend on the distribution in its neighborhood of objects that were labeled as wanted or unwanted in the training set (Section 2). An object that has a lot of wanted objects in its neighborhood and few unwanted objects should receive a higher score as it is more likely to represent a feature that appears in wanted documents. To account for unwanted objects, we can preprocess the training set in order to give each wanted object a *uniqueness weight*, which indicates how unique to the wanted set an object really is or whether it is similar to objects that appear in the unwanted set. The uniqueness weight for a wanted cluster center is computed as the ratio $N_i/(N_i+U_i)$, where N_i is the number of objects in the wanted cluster, and U_i is the number of objects inside all the unwanted clusters in the neighborhood of the wanted cluster. Note the relationship of this weight to *term frequency* concepts in text retrieval. Once we have computed uniqueness weights, we can discard all the unwanted objects from the database, since their presence in the neighborhoods of the wanted objects is now *summarized* in these weights. We define the score of an individual cluster as the product of the uniqueness weight of the nearest neighbor from the wanted set (within a threshold range) by the number of objects in the individual cluster (larger clusters are given more weight). The document score is the sum of the scores of its clusters, normalized by the total number of objects in the document (without this normalization, documents containing more lines would tend to receive a higher score). With the quadrilateral representation using Arkin's distance, we use a brute force search of neighbors from the wanted set. For the Euclidean representations, we store the set of wanted objects in a tree structure to perform an efficient range search [2].

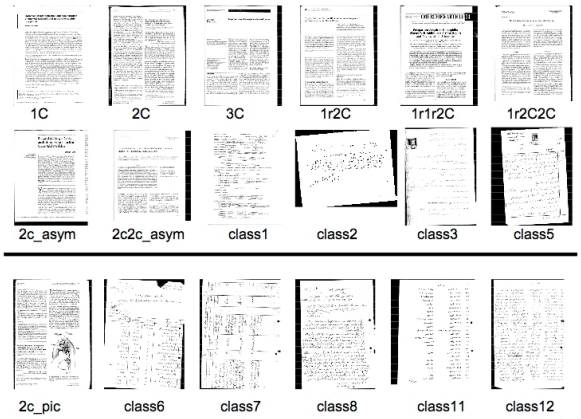

Figure 3: *Top rows*: Document samples for the twelve classes used in our experiments. The documents of each of these classes were taken in turn as "wanted documents" while those in the other eleven classes and six more classes shown at the bottom row were considered unwanted. The first eight classes group papers printed in English, where the differentiating visual features are the numbers of text columns, their symmetry, the presence of a single or double-column abstract in a double-column document, the presence of a title, etc. The last four classes contain handwritten Arabic text, and respectively contain (1) forms, (2) notes, (3) résumés, and (4) formal letters with headers. *Bottom row*: In addition, an additional set of six classes remained in the set of unwanted documents for all the experiments. Samples of these classes are shown in the bottom row. These six classes contributed around 10% of the document set which contained more than 2500 documents.

5. Performance Measures

We used two performance measures for the three layout representations we tested, the Mean Average Precision (MAP) for 100 documents, a familiar measure applied extensively in the TREC and TRECVID performance evaluations [12], and the Mean Average Normalized Rank (MANR). While average precision at 100 evaluates the ranking quality for the 100 top ranked documents, average normalized rank (ANR) [9] describes the quality of the whole ranked list of documents. Its definition is

$$ANR = \frac{1}{NN_w} \sum_{i=1}^{i=N_w} (R_i - \frac{N_w + 1}{2}),$$

where N is the number of documents in the set, N_w is the number of wanted documents in the set, and R_i is the rank of each wanted document in the set. ANR has a value of 0 when the wanted items have all been sorted on top, a value close to 0.5 where they are randomly shuffled in the list, and a value close to 1 when they are all sorted at the bottom while we want them at the top. Since the ANR's are normalized over similar ranges, it is appropriate to average them over experiments performed with various document classes in order to obtain a measure of the intrinsic quality of each approach. We call the resulting measure the Mean Average Normalized Rank (MANR).

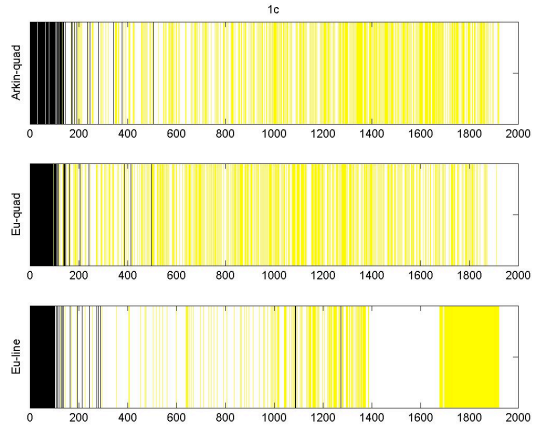

Figure 4: Ranking of printed documents belonging to Class 1c (single column and title) with our three representations, from highest score (left) to lowest score (right). The black vertical tick marks correspond to the wanted documents. The lighter tick marks correspond to the unwanted documents from six classes that remained unwanted in our experiments. The white regions correspond to unwanted documents from the 11 other classes.

6. Experiments

We organized by hand 2555 documents into 18 classes. Sample documents for each of these classes are shown in Figure 3, and a description of these classes is given in the caption. The ranking results can be displayed visually as "barcode graphs" shown in Figure 4. The graphs produced by the three layout representations are respectively labeled *Arkin-quad* (quadrilaterals using Arkin's distance), *Eu-quad* (line pairs and Euclidean distances) and *Eu-line* (lines and Euclidean distances). The black vertical tick marks correspond to actually wanted documents. Most of them are correctly ranked on top of the list, i.e. to the left of the Figure. The lighter tick marks correspond to the unwanted documents from the six classes that remained unwanted in our experiments. The white regions correspond to unwanted documents from the 11 other classes. For this class (class 1c) the results of *Eu-quad* and *Eu-line* are slightly better than *Arkin-quad*.

The overall ranking results are presented in the two tables of Figure 5. In the top table, the first two columns show how many documents were used in the training set and the testing set when each class was taken as wanted against the documents of the 17 other classes. The last four columns are the results for the three layout representations. To obtain *Eu-line2* data we only doubled the range used for *Eu-line* in the neighborhood range search used in score cal-

Average Normalized Rank (ANR)

Layout Name	Training size	Testing size	Arkin-quad	Eu-quad	Eu-line	Eu-line2
1c	46	113	0.013	0.008	0.024	0.099
2c	9	10	0.011	0.045	0.064	0.105
3c	20	23	0.0002	0.000	0.000	0.0002
1r2c	112	144	0.072	0.158	0.158	0.097
1r1r2c	67	431	0.012	0.021	0.085	0.075
1r2c2c	116	362	0.077	0.155	0.167	0.081
2c_asym	3	6	0.028	0.014	0.032	0.032
2c2c_asym	10	45	0.0002	0.0002	0.020	0.064
class1	50	62	0.002	0.006	0.011	0.012
class2	100	264	0.013	0.058	0.010	0.011
class3	49	121	0.030	0.146	0.033	0.070
class5	60	95	0.065	0.100	0.133	0.295
Mean			*0.027*	*0.048*	*0.061*	*0.197*
Time cost			*4 days*	*13.5 hrs*	*2 hrs*	*2 hrs*

Average Precision for N = 100 and MAP

Layout Name	Arkin-quad	Eu-quad	Eu-line	Eu-line2
1c	0.932	0.997	0.991	0.988
2c	0.415	0.251	0.057	0.071
3c	0.987	1.000	1.000	0.989
1r2c	0.769	0.506	0.528	0.577
1r1r2c	1.000	0.986	0.901	0.972
1r2c2c	0.996	0.841	0.578	0.896
2c_asym	0.805	0.552	1.000	1.000
2c2c_asym	0.993	0.990	0.996	0.997
class1	1.000	0.998	1.000	0.998
class2	0.993	0.938	0.996	0.994
class3	0.698	0.347	0.755	0.897
class5	0.524	0.247	0.641	0.720
MAP	***0.843***	***0.721***	***0.787***	***0.854***

Figure 5: *Top*: Training set size, testing set size for 12 document classes, and results measured with Average Normalized Rank for the three document ranking methods. The fourth column was produced by using a larger range radius in the third method. The last two rows show the Mean Average Normalized Rank and the time spent for computing both tables. *Bottom*: Average precision calculated for the top 100 documents when each class is taken in turn as wanted class, and Mean Average Precision (botttom row).

culation ($r = 0.1$ instead of $r = 0.05$). *Arkin-quad* produces the best Mean Average Normalized Rank (top table; smaller is better). It is a measure that reflects the ranking quality of the whole ranked list of documents. However, Mean Average Precision over the 100 top ranked documents may be a better measure to evaluate a system in which the user only looks at documents placed on top of the ranked list. When MAP is used (bottom table; larger is better), the three methods are scored very close and perform well, with a MAP between 0.7 and 0.9; this means that the user will typically find only one out of four documents to be unwanted at the top of the ranked lists returned by the system; this is an encouraging result. *Eu-line2* produces a slightly better ranking than the others. Compare, however, the computation times of the three methods for the whole set of experiments (each class taken in turn as wanted against all the other documents taken as unwanted). *Eu-line* and *Eu-line2* are 50 times faster than *Arkin-quad*. This is partly due to the fact that we did not implement efficient clustering and range searching algorithms for *Arkin-quad*, while we were able to use such techniques for Euclidean representations. Even if we do implement them, while *Eu-line* handles N text line as objects, *Arkin-quad* will still have to handle around N^2 quadrilateral objects, so that the ratio of completion time performance between the two methods is bound to increase linearly with the number of documents in the collection. It is possible, however, that for highly degraded document images (carbon copies, old photocopies), the performance of *Eu-line* would deteriorate so much compared to the methods using line pairs that one would be willing to accept more costly processing. We need to assemble a database of such documents to reach further conclusions.

7. Conclusions

Our main contributions are the following:

1. We propose effective representations of document layout that rely on describing quadrilaterals by their turning function or as 5D Euclidean vector where their components are distances.
2. Instead of including objects from unwanted documents explicitly in the database generated during the training phase, we include only objects from wanted objects, with weights that implicitly describe the number of unwanted objects in their neighborhoods. This greatly reduces the number of objects in this database and speeds up the range search required by the document scoring process.
3. Scores for documents can be computed as normalized sums of object scores, where object scores are proportional to the weights of the nearest neighbor objects in the wanted documents.
4. Individual lines can be represented by Euclidean vectors found by considering the quadrilaterals they form with the top of the de-skewed bounding box. For document collections with moderate noise level, they provide a Mean Average Precision that is comparable to the much more expensive descriptions using pairs of feature lines. However, for degraded document collections in which it is not possible to find a text bounding box reliably, the pairwise quadrilaterals should be used.

A key component of the proposed approach is its adaptability, for document collections where the wide range of layouts prohibits formal document analysis. In future work, we will explore the inclusion of lines other than text lines, for example zone boundaries and separating lines in forms and tables.

Acknowledgments

The partial support of this research under DOD contract MDA90402C0406 is gratefully acknowledged.

References

[1] E.M. Arkin, L.P. Chew, D.P. Huttenlocher, K. Kedem and J.L. Mitchell, "An Efficiently Computable Metric for Comparing Polygonal Shapes", IEEE Trans. PAMI, vol. 13, no. 3, pp. 209–216, 1991.

[2] S. Arya and D. Mount, "Approximate Range Searching", Computational Geometry: Theory and Applications, vol. 17, pp. 135–163, 2000.

[3] O. Altamura, F. Esposito, D. Malerba: "WISDOM++: An Interactive and Adaptive Document Analysis System", in Proc. ICDAR, pp. 366–369, Sept. 1999.

[4] F. Carmagnac, P. Héroux, E. Trupin, "Distance Based Strategy for Supervised Document Image Classification", Int. Workshop SPR 2004, August 2004.

[5] D. Doermann, "The Indexing and Retrieval of Document Images: A Survey", Computer Vision and Image Understanding, vol. 70, no. 3, pp. 287–298, 1998.

[6] L. Golebiowski, "Automatic Layout Recognition", Proc. 1st ACM Hardcopy Document Processing Workshop (HDP 2004), Washington, D.C., Nov. 2004.

[7] J. Hu, R. Kashi, and G. Wilfong, "Document Image Layout Comparison and Classification", Proc. ICDAR, 1999.

[8] J. Liang and D. Doermann, "Logical Labeling of Document Images Using Layout Graph Matching with Adaptive Learning", Proc. DAS, pp. 212–223, 2002.

[9] H. Müller, S. Marchand-Maillet, and T. Pun, "The Truth about Corel–Evaluation in Image Retrieval", Proc. CIVR, 2002.

[10] G. Ressler, Implementation of "An Efficiently Computable Metric for Comparing Polygonal Shapes," by Arkin et al., http://www.cs.sunysb.edu/∼algorith/implement/turn/distrib/sim.c

[11] C. Shin, D. Doermann, and A. Rosenfeld, "Classification of Document Pages Using Structure-Based Features", IJDAR, vol. 3 no. 4, pp. 232–247, May 2001.

[12] TREC-10 Proceedings Appendix on Common Evaluation Measures, http://trec.nist.gov/pubs/trec10/appendices/measures.pdf