Distance Measures for Layout-Based Document Image Retrieval

Joost van Beusekom, Daniel Keysers, Faisal Shafait, Thomas M. Breuel

Image Understanding and Pattern Recognition (IUPR) Research Group
German Research Center for Artificial Intelligence (DFKI)
and Technical University of Kaiserslautern
D-67663 Kaiserslautern, Germany
{joost, keysers, faisal, tmb}@iupr.net

Abstract

Most methods for document image retrieval rely solely on text information to find similar documents. This paper describes a way to use layout information for document image retrieval instead. A new class of distance measures is introduced for documents with Manhattan layouts, based on a two-step procedure: First, the distances between the blocks of two layouts are calculated. Then, the blocks of one layout are assigned to the blocks of the other layout in a matching step. Different block distances and matching methods are compared and evaluated using the publicly available MARG database. On this dataset, the layout type can be determined successfully in 92.6% of the cases using the best distance measure in a nearest neighbor classifier. The experiments show that the best distance measure for this task is the overlapping area combined with the Manhattan distance of the corner points as block distance together with the minimum weight edge cover matching.

1 Introduction

Most information that is currently available digitally — especially in libraries — is organized in form of documents, and those are typically stored in databases. The task of finding relevant information in such databases is a crucial problem of the information society. Many methods for document retrieval exist, but their success depends strongly on the format in which the documents are stored: the Google search engine does a very good job in document retrieval for WWW pages. The Windows operating system contains a search assistant that does a fast full text search in all MS Word or Excel documents on a PC's hard disk within seconds. But so far, no software system can reliably do content-based search in image or video files, let it be on the web or locally.

A problem of current querying methods is that they require a document to be present in text form, and their method to find similar documents is by comparing the textual contents.

For documents in image form, as they are produced e.g. by a scanner, this approach has some drawbacks, since the document has to be converted to text first by Optical Character Recognition (OCR) software. This process is computationally expensive, and it can also introduce errors that may prevent a document from ever being found again. A more fundamental problem is that the textual contents of the documents to be searched for can be unknown, e.g. when searching for all CD-covers on a home PC, or the text information is irrelevant or not sufficient to answer a query, e.g. when a user wants to search for all *IEEE*-style publications in an archive.

In this paper, we present a method to query document image databases by layout, in particular by measuring the similarity of different layouts in comparison to a reference or query document. The method works directly on the image data and does not require a costly OCR step. Depending on the application, it can either be the only search criterion used or act as an additional search feature for the user.

Distance measures for measuring the similarity of two layouts can be used in numerous ways. In this paper, we concentrate on their use for layout-based document retrieval. Other possible uses include the benchmarking of different layout analysis algorithms or tie-breaking in layout analysis system that are based on the combination of different layout analysis techniques. We also restrict ourselves to geometric layout information, i.e. how a page is split into different homogeneous regions like columns and paragraphs. We consider only Manhattan layouts because they represent the most general class of layouts found in practice, and they can be easily represented by a set of rectangular blocks. We do not study the logical partitioning of the page into semantic blocks like title, abstract, and author.

To find the similarity between two layouts, we need a distance measure. The desired properties of a distance measure are described in Section 3. Then, in Section 4, we investigate the possibility to use a benchmarking method for layout analysis algorithms as a similarity measure. Then, we propose our method consisting of a distance measure between the rectangular blocks of the two layouts, combined with a matching step from the blocks in one document image to the blocks in another document image. The details are described in Section 5. Our evaluation method is elaborated in Section 6 followed by results in Section 7 and conclusions in Section 8.

2 Related Work

Most of the work in the field of document image retrieval uses features computed on the document image such as font information, connected components, and texture. We disregard this type of information here in order to analyze to what extent geometric layout information alone can be used for document image retrieval. However, the features mentioned above can be combined with the layout similarity measures developed in this work to improve the performance of the retrieval system.

The layout similarity measures for document images have been scantily addressed in literature. Hu et al. [6] present a two step method for layout comparison. They use different methods to compute the distance between image rows after a segmentation into a grid of equal-sized cells. Each cell is identified as text cell if at least half of the cell is part of some text block. In the other case it is a white space cell. Document images are then compared using dynamic programming on the row-based representation of the documents. In [5] the use of clustering and a hidden Markov model for learning of prototypes is discussed in more detail. The test data used comprises five classes (1-column and 2-column letter, 1-column and 2-column journal, and magazine) and an average error rate of 21.4% is reported. Unfortunately, the data used is not available, so a direct comparison with the results is not possible.

Benchmarking methods for layout analysis algorithms have to compare the output of the algorithms to the ground truth. Because these measuring methods yield a quantitative description of the difference between two layouts, they can be used as a distance measure for the task of document image retrieval.

Mao and Kanungo present in [12] their page segmentation evaluation toolkit in which they use morphological operators to define the sets of missed, merged, split, and falsely detected text lines. The error types are weighted and a total metric is obtained by summing up the error types multiplied by their weight.

Liang et al. [11] use area overlap for finding the correspondences between the blocks in the ground-truth and the segmented images. Then, different errors are defined on the basis of correspondence analysis and the total error rate is defined as a weighted sum of the individual errors.

In [18] Yanikoglu and Vincent present a method that uses a method similar to the ones mentioned above, but instead of working with regions or blocks, the number of errors are counted on the basis of pixels. The total error is then again obtained by summing up the number of errors multiplied by the weighted cost.

3 Desired Properties of a Distance Measure

The distance measure we are interested in should primarily be used for document retrieval by layout comparison and the (dis)similarity will be based on the blocks that define the different regions in the documents. The following criteria may be used according to which we define our similarity:

- Position: If the positions of the blocks of the two layouts are similar then this will be in favor for the analyzed layouts.
- Width: If the width of the blocks is the same, the two layouts are likely to have the same column width.
- Area: If the two blocks of different layouts differ in area this might be an indicator that these blocks should not be matched to each other.

Furthermore, it would be useful if the distance measure was tolerant to some typical errors that occur during layout analysis, namely merge and split errors. It should also allow a few false alarms and missed errors. A merge error occurs when two blocks of the query layout are transformed into one block of the reference layout. A split error occurs in the inverse case. A false alarm occurs when one block from the query layout is not matched to any other block of the reference layout. A missed error occurs when a block from the reference layout is not matched by any block of the query layout. Another type of errors, spurious errors, are those errors that fit in none of the error categories mentioned above. Examples for these errors can be found in [11].

Another important feature of a distance measure used for queries is that it should be fast to compute. Since a query layout has to be matched against all layouts in the database, retrieval time becomes a critical issue as the size of the database grows. Hence, it should be a general goal to calculate the distance measure in an acceptable amount of time, e.g. less than a few seconds.

4 The Benchmarking Distance

Layout analysis algorithm benchmarking tries to find an objective measure for the performance of a layout analysis

algorithm. The benchmarking methods compare the output of a layout analysis algorithm to the given ground truth and report different errors made by the algorithm. Hence, the results of this comparison can be used as a similarity measure for document image retrieval. The layout of the query document is matched against all documents in the database and the document giving lowest error rate is retrieved.

One method for benchmarking layout analysis algorithms was presented by Liang et al. in [11]. It is based on the computation of the overlapping area of the matched blocks. But instead of using this area to build the distance measure, it uses the relations it gets from this overlapping area analysis to find out what different type of errors have been made by the segmentation algorithm. Then a weighted sum of these errors, normalized by the total number of blocks in the two layouts, is used as the final error metric.

The reason why we chose this algorithm is its simplicity. Furthermore, the method is fast to compute and so it fulfills one condition for layout based image retrieval.

5 Block Distance plus Matching

Since we consider only Manhattan layouts, a rectangular block can be considered as the basic entity of a document layout. Hence, it is self-evident that a distance measure for two layouts can be based on some distance measure for two blocks in order to obtain a global distance. The idea that emerged from this observation was to use a distance measure for blocks (in the following called "block distance") to obtain one global distance measure. Using one block distance, we can compute the distance between every pair of blocks in the two layouts.

The problem that arises now is to match the blocks from the query layout to the reference layout in order to minimize the total distance obtained by summing up the block distances of the blocks that are matched. This problem is referred to as the "matching problem".

Considering that we have different methods to compute the block distance and also three different methods to solve the matching problem, we may combine these two steps in many ways. Different combinations have been tested and evaluated. An example that illustrates such a matching is shown in Figure 2.

In the first part of this section we will present a number of block distances. The second part then presents the three different matching methods that have been analyzed. The third part is about the testing and the evaluation of these methods.

In the following, we use these definitions:

- layout: a layout L is a set of blocks $\mathbf{B}_i$: $L = \{\mathbf{B}_1, ..., \mathbf{B}_n\}$
- block: a block $\mathbf{B}$ is a pair of two points $\mathbf{p}_i$: $\mathbf{B} = (\mathbf{p}_1, \mathbf{p}_2)$, the lower left and the upper right corner.
- point: a point $\mathbf{p}$ is defined by a pair of coordinates x, y: $\mathbf{p} = (x, y)$

The coordinates can be computed relative to some reference point in order to allow the translational displacement of a page, for example by taking the center of gravity as reference point for each layout. However, in the experiments on the MARG dataset, we observed that the documents were scanned with low variation in their position on the scanner, so such a normalization was not necessary.

5.1 Block Distances

In the following we present different block distances that can be used to compute the distance between two blocks. This computation is the step that gives us the cost matrices, also called weight or block distance matrices, needed for block matching in the second step.

5.1.1 Manhattan Distance of Corner Points

One possible block distance is the sum of the Manhattan distances of the corner points of the two compared blocks. Let $\mathbf{B}_i = (\mathbf{p}_k, \mathbf{p}_l)$ and $\mathbf{B}_j = (\mathbf{p}_m, \mathbf{p}_n)$ be two blocks of two layouts. Then the block distance $D_{\mathrm{mh}}(\mathbf{B}_i, \mathbf{B}_j)$ is obtained by summing up the Manhattan distances of the corner points of blocks $\mathbf{B}_i$ and $\mathbf{B}_j$.

$$D_{\mathrm{mh}}(\mathbf{B}_i, \mathbf{B}_j) = d_{\mathrm{mh}}(\mathbf{p}_k, \mathbf{p}_m) + d_{\mathrm{mh}}(\mathbf{p}_l, \mathbf{p}_n) \quad (1)$$

where

$$d_{\mathrm{mh}}(\mathbf{p}_a, \mathbf{p}_b) = |x_a - x_b| + |y_a - y_b| \quad (2)$$

is the Manhattan distance between two points.

As in this distance both position and area take influence, this might be a reasonable block distance measure to use.

5.1.2 Overlapping area

This method computes the distance between two blocks by their overlapping area. The overlapping area is defined as the number of pixels that belong to the two blocks being compared and that have the same coordinates on the page.

For every pair of blocks $(\mathbf{B}_i, \mathbf{B}_j)$, where $\mathbf{B}_i$ and $\mathbf{B}_j$ are from two layouts, the following distance is computed:

$$D_{\mathrm{ov}}(\mathbf{B}_i, \mathbf{B}_j) = 1 - \frac{2 \times \mathrm{Ov}(\mathbf{B}_i, \mathbf{B}_j)}{\mathrm{area}(\mathbf{B}_i) + \mathrm{area}(\mathbf{B}_j)} \quad (3)$$

where $\mathrm{area}(\mathbf{B}_i)$ is the number of pixels (area) of block $\mathbf{B}_i$ and $\mathrm{Ov}(\mathbf{B}_i, \mathbf{B}_j)$ is the overlapping area of block $\mathbf{B}_i$ and

$\mathbf{B}_j$. For every pair of blocks a value between 0 and 1, where 0 is a perfect overlap and 1 is no overlap at all, is obtained.

This block distance incorporates position as well as area and aspect-ratio into one measure. Taking only the overlapping area as block distance has one major drawback: in case of non-overlap, the distance will be 1, no matter how far or how near the two blocks actually are.

Therefore, we combined the overlap with the normalized Manhattan distance of the corner points in case that there is no overlap. For a block having no overlap in common with any other block we get a distance between 1 and 2. The normalization of the Manhattan distance of the corner points is done by dividing the obtained distance of each corner point by twice the maximal Manhattan distance on the page, namely the sum of width and height.

5.1.3 Other simple block distances

It is clear that a lot of other block distances can be built. As the following block distances are almost self-explanatory, they are only defined here:

- Difference in width:

$$D_w(\mathbf{B}_i, \mathbf{B}_j) = |\mathrm{width}(\mathbf{B}_i) - \mathrm{width}(\mathbf{B}_j)| \quad (4)$$

- Difference in height:

$$D_h(\mathbf{B}_i, \mathbf{B}_j) = |\mathrm{height}(\mathbf{B}_i) - \mathrm{height}(\mathbf{B}_j)| \quad (5)$$

- Product of difference in width and difference in height:

$$D_p(\mathbf{B}_i, \mathbf{B}_j) = D_h(\mathbf{B}_i, \mathbf{B}_j) \times D_w(\mathbf{B}_i, \mathbf{B}_j) \quad (6)$$

- Distance of block centers:

$$D_{bc}(\mathbf{B}_i, \mathbf{B}_j) = d_{\mathrm{mh}}(\mathrm{center}(\mathbf{B}_i), \mathrm{center}(\mathbf{B}_j)) \quad (7)$$

Note that the block distances presented here only constitute different choices that can be made. It is not said that all these block distances are meaningful or give good results. The idea is to evaluate how well these simple measurements work and then to try to obtain a better distance measure by combining different simple block distances, each containing a different kind of information, to one new block distance. Examples for these simple block distances are shown in Figure 1.

5.1.4 Combination of different block distances

Instead of choosing one block distance to compute the distance between two blocks it might be useful to combine different block distances into one new block distance.

The difference in width, for example, contains a lot of information concerning the number of columns we have. In order to add position information of the blocks, we might add, for example, the distance of centers of the blocks.

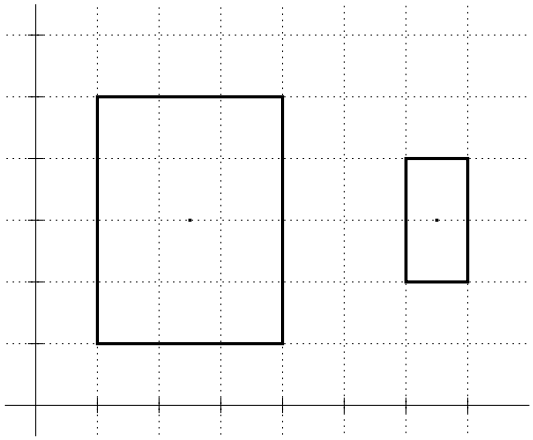


Figure 1. Example for simple block distances: the difference in width is 2 units, the difference in height is 2 units, the product of difference in width and difference in height equals 4. The number of overlapping pixels equals zero, so the combination of overlapping area distance and Manhattan distance will be 1.45 (the maximum Manhattan distance is 11, and the Manhattan distance of the corner points equals 10 units).

As one can see, we obtain a lot of different possibilities bringing some new problems:

- Scale: the different distance measures may have different scales, so they should all be converted to the same scale, e.g. if we want add area to Manhattan distance.
- Addition or multiplication: to combine the results of different distance measures, there are different methods, especially addition or multiplication: instead of adding different block distances they also could be multiplied. This is only meaningful in a few cases where one of the block distances has the ability of defining a perfect match, as e.g. for the overlap: if the overlap distance equals 0, than the position and the size and the aspect-ratio of the two blocks is the same, so we have a perfect match. Other ways of combining the simple block distances are also possible.
- Weighting: different block distances could be weighted differently according to their importance regarding our criteria, e.g. if we add the difference in width with the difference in height, it would be useful to give more importance to the difference in width, as for our purpose, column width is more important than paragraph height.

5.2 Matching

In the following, we present the different matching methods. The matching step matches the blocks of the query layout to the blocks of the reference layout trying to minimize the total distance. The total distance is obtained by summing up the costs of the matches, which are equal to the block distance between the two blocks that are matched. An example of a matching can be found in Figure 2.

In the literature, the matching problems are usually presented using the term "cost" instead of "block distance". We will use the term "cost" when discussing the general matching problem and when we are applying the method to our purposes we will use the term "block distance".

Three different matching methods will be discussed:

- Assignment Problem: each block is matched at most once.
- Minimum Weight Edge Cover Problem: each block is matched at least once.
- Earth Mover's Distance / Transportation Problem: each block is matched partially to at least one other block.

5.2.1 Matching by Solving the Assignment Problem

As mentioned above, the aim of this step is to match blocks from the query layout to the blocks of the reference layout by minimizing the total cost. The total cost is the sum of all matches between two blocks multiplied by their cost that in our case will be given by some block distance.

For the "Assignment Problem" each block is allowed to be matched at most once and every block that can be matched should be matched. So there will be exactly $\min(|L_a|, |L_b|)$ matches. It may happen that, if the number of blocks of the two layouts differ, some blocks are not matched. These are called "unmatched" blocks and in that case our assignment problem is called "non-quadratic" because of the obvious property of the cost matrix in that case.

This problem can be solved by the "Hungarian Algorithm", which is described in more detail in [10].

Handling non-quadratic problems In our application of the assignment problem it often happens that the problem is not quadratic. This is the case when the number of blocks in the two layouts differ. After the assignment there are a number of blocks that are not matched at all. We have to take care about handling these blocks appropriately. Simply ignoring them would give good similarities for layouts consisting of a few blocks only. Penalizing them by some value afterwards could be a possibility but also inserting dummy blocks with a certain penalty distance in order to get a quadratic problem could be a solution, which is the approach we followed (cp. Section 5.2.4).

5.2.2 Matching by Solving the Minimum Weight Edge Cover Problem

The minimum weight edge cover problem consists of finding matches for the same problem as the assignment problem, with the difference that every block of L_a is connected to at least one block of L_b and vice versa. This problem is again solved using the Hungarian algorithm. A description of this method can be found in [8].

5.2.3 Matching using the Earth Mover's Distance

One other interesting approach we analyzed is to use the Earth Mover's Distance (EMD) as matching method. Instead of matching entire blocks, here blocks can be divided into different basic units (in our case pixels) that are assigned to other blocks.

The EMD was used in [15] for image retrieval of color images. Simply spoken, the idea behind it is to calculate the cost to "transform" a generalized form of histogram of one image to the histogram of the other. In our case we want to move pixels so that the reference layout is built by moving pixels from the query layout to other blocks. As moving these pixels has a certain cost, a total cost can be computed to convert one layout into another. Blocks, as well as histograms are called in this concept "signatures", which is a more general concept.

A signature is defined as a set of feature clusters represented by their mean and by the fraction of pixels ("earth") that belongs to this cluster: signature $S = \{s_j = (m_j, w_j)\}$ where m_j is the mean value of the cluster and w_j the fraction of pixels that belong to cluster j. A page layout can also be considered as a signature: the blocks represent the clusters and the fraction is given by the area of the blocks.

In our case the blocks can be considered to be entirely black, so no text information is contained in the blocks. All the pixels in the block are considered as black pixels.

The computation of the EMD is based on the solution of the well known transportation problem. An algorithm for solving the transportation problem can be found in [7].

Coming back to the initial idea of "earth moving" the signatures S_m will be replaced by layouts that consist of blocks and a given block distance that gives the cost for transporting one pixel from block $\mathbf{B}_i$ to block $\mathbf{B}_j$.

By finding the solution for the transportation problem for two signatures we obtain the total cost of transforming signature S_m to signature S_n. This cost is divided by the total flow (the total number of transported units) and then defined as Earth Mover's Distance.

5.2.4 Implementation Details

For the assignment problem we chose to make the problem quadratic, in case the number of blocks in the two layouts differs by inserting dummy blocks. These dummy blocks need a certain penalty distance. Depending on this distance the algorithm tends to make different mistakes: if no penalty is assigned, layouts with few blocks will be too good a match. If the penalty value is too high, the matching method is very inflexible if the number of blocks in the two layouts differ (which may happen if e.g. splitting or merging errors occur). Various methods of defining this penalty have been analyzed, e.g. the mean block distance value, no penalty at all, half of the maximum block distance value, etc., and we opted for a penalty value that is given by the maximum block distance value that exists between two real blocks.

Normalizing the total distance by the number of blocks has also been tried but did not give better results, due to the fact that by dividing, we loose the information about the number of blocks per page.

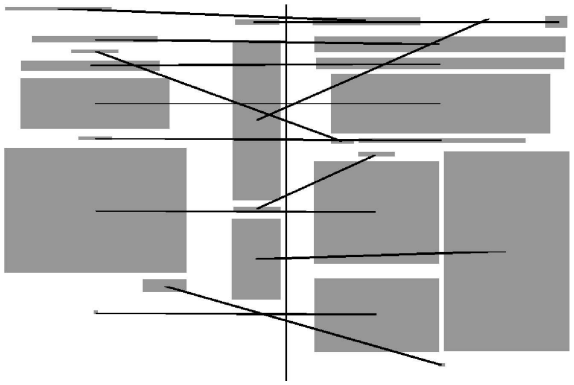
For the minimum weight edge cover method we did not need to specify any parameters.

Examples for the matching result for the assignment and the minimum weight edge cover problem are shown in Figure 2. The left side query document contains blocks on the right side that are not part of the page but parts of blocks from the neighboring page of the book. These artefacts come from the scanning process. The Voronoi layout analysis algorithm was used to extract these layouts. As one can see, the minimum weight edge cover method finds a correct match (the same journal), regardless of these "artefacts". The assignment problem based method returns a wrong layout as layouts with approximately the same number of blocks are preferred, due to the penalty value for unmatched blocks.

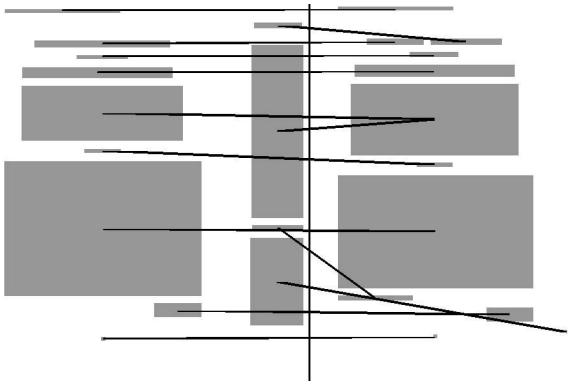
For the Earth Mover's Distance we had to find some solution for the case that the two layouts have different number of pixels (only pixels belonging to blocks are considered). As the area plays the role of demand and supply in the transportation problem, the initial transportation problem is unbalanced. So dummy blocks have to be inserted to solve the transportation problem. These dummy blocks don't have a position nor a size, they simply are pixel "producers" or pixel "consumers", in order to solve the transportation problem.

For the usual application of the transportation problem, this method works fine as there can only be transported as much as there is supplied and needed. The fact that the demand or the supply are too high will have no effect on our total cost.

This is not appropriate for our purpose. We want to take into account all pixels, even if the number of pixels in the blocks of the two layouts may differ. As for the assignment



(a) Best match for the left side query layout by assignment problem matching using as block distance the overlapping area.



(b) Best match for the left side query layout by minimum weight edge cover matching using as block distance the overlapping area.

Figure 2. Examples for the two matching methods. The lines indicate which blocks are matched to each other. The block distance used in both cases is the overlapping area, d_{ov}. The best match for the assignment matching method finds a wrong layout (different journal), whereas the minimum edge cover method finds a correct layout (same journal). The layouts were extracted using the Voronoi layout analysis algorithm.

problem, there are at least two possibilities: make the problem a balanced one or penalize the unmatched pixels afterwards. As the second solution needs some intelligent way of setting a penalty value and the first solution gave better results, we opted for the first solution: we normalize each layout to a size of one and each block does not have a fixed size in pixels but only a fraction of pixels of the layout that belong to it.

An example how the pixels from one layout are matched to the other layout for the two methods is shown in Figure 3.

6 Evaluation of Distance Measures

The problem with evaluating the distance measures is to define what a good result should be like. It should return for a given query layout a reference layout from a database that "looks" similar, but the problem is to provide a ground truth for that notion without too much manual labeling.

We decided to use a publicly available document image database that is annotated with the source of the document images (i.e. the journal or magazine it originated from). As for one specific journal the layout should look very similar for different articles, a query with a document from one given journal should return a document of the same journal, which is the definition of correctness we used in the experiments.

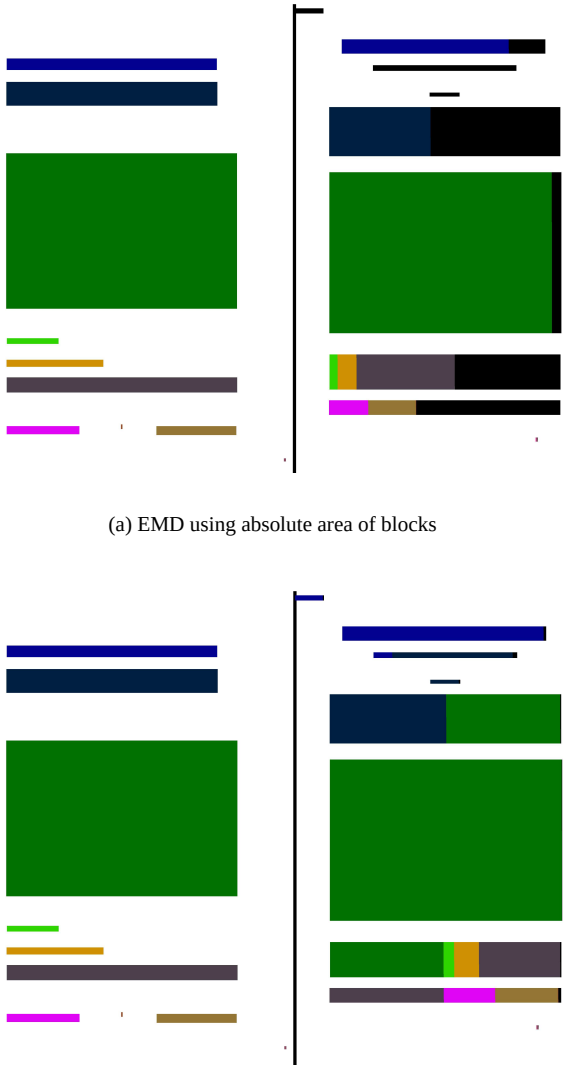
The database we chose for this task is the MARG (Medical Article Records Ground-truth) database. It contains 815 scanned documents of first pages of medical journals, sorted by type (9 different types) and journal (161 different journals). Further details about the MARG database[1] can be found in [4].

As different journals in this database are published by the same publisher and publishers often have similar layouts for their journals, we additionally manually sorted the MARG database according to the publisher. After sorting we obtained 59 different publishers.

Because the database is not divided into separate training and test sets and it does not contain a large number of samples, we decided to use the leave-one-out error rate as the evaluation criterion. As a classifier we used the simple nearest neighbor classifier, because our goal is primarily to evaluate the usefulness of the distance measures. That is, we queried the system with each document image in turn and determined the closest reference document image according to the distance measure evaluated (not including the query document). If the returned best match was from the same class (i.e. journal, type, or publisher depending on the setup), this was counted as a correct classification, otherwise as an error.

As layout information we first used the layout ground

[1] http://marg.nlm.nih.gov/index2.asp



(a) EMD using absolute area of blocks

(b) EMD using normalized area of blocks

Figure 3. (a) Result of the EMD without normalization. (b) Result of the EMD with the areas normalized to one. The left side shows the query image, the right side the best match. The colors indicate the corresponding pixels, where in (a) black is used to indicate unmatched pixels. The block distance used is the sum of the Manhattan Distance of the corner points.

truth of the MARG database. This ground truth only contains information about four special kinds of blocks, namely "author", "abstract", "affiliation" and "title". However, after a few tests, it became clear that this partial layout information was not a good basis to test the performance of the distance measures. Completely segmented pages are a better choice to run these tests.

We therefore chose to use known layout algorithms to extract the layout. This is also the case which is more relevant to a practical task, because in practice, it cannot be expected to have access to manually extracted layout information. Although the algorithms will not yield perfect segmentations, they should produce similar errors for similar layouts. The algorithms we used for extracting the layout information are: Voronoi algorithm [9], XY-Cut algorithm [13], Run length smearing algorithm [17], Docstrum algorithm [14], Whitespace algorithm by Baird [1] and Whitespace algorithm by Breuel [2].

7 Results

7.1 Error rates

There are three different error rates that were computed in order to evaluate the different distance measures:

- Journal: This is the error rate for finding the correct journal for a given query document out of a specific journal. This error rate is the one that should give us, according to the main idea of this evaluation method, a good overview about how good the distance measure works, as we expect to find for a given query document a document of the same journal. There are document images from 159 different journals.

- Type: This error rate gives the ratio of misclassifications of the document type. This error rate should be low, as the distance measure should be able to identify the right layout type. Furthermore, only nine different layout types have to be distinguished.

- Publisher: This error rate gives the ratio of misclassifications of the document publisher, e.g. if the query document is from publisher Elsevier but has been classified as publisher Springer. This error rate is less important because it may be that two journals from the same publisher have different layout, although the inverse case is quite frequent.

We are aware of the fact that the error rate is not always the most appropriate performance measure for retrieval. However, the error rate is strongly correlated with most other metrics that are normally applied (cp. [3]).

7.2 Evaluation of Matching Methods

As mentioned in Section 5.2, three different matching methods have been tested: the assignment problem, the minimum weight edge cover problem and the earth mover's distance based on the transportation problem.

These three methods have been tested with different block distances in order to find out whether one method has an overall advantage over the other or if the performance of the matching depends on the block distance, in a way that a good block distance may compensate a bad matching.

As the ground truth contains only incomplete page segmentations, we ran the major part of the tests on the output of the different layout analysis algorithms. As [16] stated that the Voronoi layout analysis algorithm is generally a reasonable choice, we chose this to illustrate our results in this part.

Table 1 shows the error rates obtained by applying the three different matching methods to different block distances. The results are representative for various tests with different layout analysis algorithms that have been performed: in short one can say that the minimum weight edge cover method performs acceptably, whereas the EMD and the assignment method perform badly, with a little advantage for the EMD. Liang et al.'s method, originally used for benchmarking, is not appropriate for our purposes because it is very sensitive to split and merge errors. Since for two layouts to be similar, it is not necessary for them to have same number and height of text segments (paragraphs). Hence, if the layouts of two different documents from the same journal are compared, they are likely to produce several vertical split, merge, and spurious errors. This is the main reason why the benchmarking method is not suitable for use as a similarity measure in layout-based document image retrieval. Another problem arises from the fact that not all necessary details for the implementation of Liang et al.'s method were available. The original code may perform better than ours.

The reason why the assignment problem performs that badly lies in the penalty for unmatched blocks: for unbalanced assignment problems (L_a has a different number of blocks as compared to L_b) unmatched blocks remain,

Distance Measure	JOUR	TYPE	PUB
Overlap / Edge Cover	32.8	8.2	7.6
Overlap / Assignment	52.0	22.9	25.4
Overlap / EMD	52.3	20.2	23.9
Liang et al.	97.2	80.0	93.5

Table 1. Comparison of different matching methods (error rates [%]).

which are penalized with the maximum occurring block distance value. This implies that layouts with similar number of blocks will be preferred, which is not always a wanted effect. Different other methods for penalizing have been tested, without big improvement (the error rate varies, but it stays far away from the minimum edge cover error rate). If we compare the EMD to the assignment method, we observe that the EMD performs slightly better.

If we interpret these results we can say that prohibiting the blocks from splitting up (as does the assignment problem) is not a good idea, as it penalizes very much the splitting and merging errors, errors that occur very frequently in document layout analysis. Splitting blocks up into very tiny parts (pixels) is not a good idea either. As one can see in Figure 3, pixels from one block may be spread everywhere on the page, a problem that is triggered by the transportation problem, but that is not necessarily wanted for document layout comparison. The minimum weight edge cover method is the most "natural" method for matching layout blocks: merging and splitting are not too expensive, as two or more blocks may be matched to the same block and based on a good block distance, blocks are not matched all over the page.

Another important result is the runtime: assignment and edge cover run in $O(n^3)$ whereas the transportation problem has worse complexity.

7.3 Evaluation of Block Distances

As shown in the preceding part, the best matching method of the three proposed methods is the minimum weight edge cover matching. In order to test the different block distances we used the minimum weight edge cover together with a few block distances and compared the results. The results were obtained with the blocks extracted by the Voronoi layout analysis algorithm on the MARG database. In Table 2 the different error rates for the various block distances can be found.

"Overlap" uses the overlapping area as block distance. So for every pair of blocks we obtain a value between 0 and 1. If two blocks have no common area at all, they will get the distance 1. In that case no conclusions can be made how similar these two blocks are. Therefore we used the block distance "Overlap + Manhattan", that, in case of non-overlap, adds the sum of the Manhattan distances of the corner points divided by twice the maximal possible distance on the page (sum of length and width of the image) to the 1 of the overlap distance. This way we obtain a value between 0 and 2 for every block and we have some information which block could be better or worse to match to if we have no overlap. "Manhattan Dist. of Corners" simply sums up the Manhattan distances of the corner points of the two blocks. "Euclidean Dist. of Corners" does the same but instead of the Manhattan distance it uses the Euclidean Distance to measure the distance between two points. "Manh. Dist. of Block Centers" computes the Manhattan distance of the center points of two blocks. "Eucl. Dist. of Block Centers" uses the Euclidean distance instead of the Manhattan distance. "Difference in Width" uses the difference in width of two blocks as block distance, so no explicit position information is used at all. "Diff. Height $\times$ Diff.Width" uses the product of the difference in width and the difference in height of two blocks. Instead of multiplying these two, one can also sum them up. This is done in "Diff. Height $+$ Diff.Width". "Difference in Area" uses the square root of the difference of the area of the two blocks. "Difference in Height" computes the difference of the height of two blocks.

Block Distance	JOUR	TYPE	PUB
Overlap + Manhattan	31.2	7.4	7.0
Overlap	32.8	8.2	7.6
Manhattan Dist. of Corners	39.7	11.3	10.5
Euclidean Dist. of Corners	40.7	11.9	11.5
Manh. Dist. of Centers	41.6	13.1	13.9
Eucl. Dist. of Centers	43.7	14.3	14.8
Difference in Width	47.4	19.4	20.4
Diff. Height + Diff. Width	49.6	17.2	18.4
Diff. Height $\times$ Diff. Width	50.7	13.2	14.6
Difference in Area	81.8	54.3	63.3
Difference in Height	88.1	60.1	70.9

Table 2. Comparison of the different block distances (error rates [%]) using the minimum weight edge cover method for matching.

As we can see, the overlapping area as block distance works quite well, compared to the other methods. This comes from the fact that the overlapping area depends on the position, size, and the aspect-ratio of the blocks, so a lot of information is contained within this single measurement. The attempt to improve this method by adding the Manhattan distance of the blocks in case of non-overlap did not improve noticeably the overall performance. This may be due to the observation that the best matches normally are made between blocks that are similar and the rest of the blocks are matched against some other block, although these are not necessarily similar. So it does not make a difference if we match them randomly as done for the first method or if we try to improve the matching by adding some additional feature in case of none matching.

Another conclusion that can be drawn is that the difference in width of two blocks contains information that is more appropriate for our purposes as the difference in height of the blocks. This is quite obvious as the width for

all the blocks in a column is the same, although the height of these blocks may vary. In addition, the column width for one journal is typically the same, so it is a better measure than the difference in height to identify the journal, but even a better one to identify the document type and the publisher.

Furthermore, it can be seen that the Manhattan distance has slightly better results as the Euclidean distance, although the difference is small.

Comparing "Diff. Height $\times$ Diff.Width" with "Diff. Height + Diff.Width", it can be seen that in this case the multiplication is the better operation to combine the two distances. Although the journal error rate is slightly higher, the type and the publisher error rates are significantly lower. As the difference in width is more meaningful than the difference in height (for our application), it is proximate that the multiplication gives better results, as the distance will be small when the width or the height is small. Using the sum to combine the two measures will lead to small distances only for blocks with approximately the same length and the same width.

Having a general look at the error rates, even the best of our block distances has an error rate of 31% on the Journal level. This seems to suggest that a lot of improvement should still be possible. However, it is unclear what the class overlap of this task is, i.e. how many journal pages cannot be distinguished by the layout alone. Recall also that the distance measures use *only* the layout information, i.e. the corner coordinates of the blocks. It is highly likely that better error rates can be achieved when including more information about the blocks, e.g. their texture, the distribution of bounding box sizes, or the output of a text/graphics classifier. Furthermore, the method is able to determine the correct layout type in 92.7% of the cases, which seems a reasonable basis for its use in document image retrieval.

8 Conclusion

We presented a method for document image retrieval by layout analysis. Different distance measures for this task have been analyzed: one method used for benchmarking layout analysis algorithms and a set of methods that result from a combination of layout block distances and a matching algorithm. Different block distances and three different matching methods have been implemented and tested. Furthermore, we presented a procedure to evaluate the performance of layout similarity measures based on a publicly available database of labeled document images and a nearest neighbor classifier.

The proposed distance measures are two step methods. First, the distance for every pair of blocks from the two layouts are computed using a block distance. In the second step a matching is done to minimize the total distance between the two layouts and thus assign blocks to each other. We evaluated various block distances and three matching algorithms, based on the assignment problem, the minimum weight edge cover problem, and the Earth Mover's Distance, which is based on solving the transportation problem.

For the block distances we found that the overlapping area is the best method of the many we analyzed. It integrates a lot of features as size, position and aspect-ratio of the blocks.

From the three matching methods, the one based on the solutions for the minimum weight edge cover problem performed best. We concluded, that apart from the problem of penalizing unmatched blocks for the assignment problem and handling unbalanced transportation problems for the Earth Mover's Distance, the minimum weight edge cover method is the most natural way of matching for layouts as it allows splitting and merging errors without penalizing the total result too much.

The overall error rate for our system on the MARG database is 7.4% for the determination of the correct layout type with a choice among nine classes. This compares favorably to the average error rate of 21.4% for the document image classification system presented in [6], where a data set with only five classes was used.

References

[1] H.S. Baird: Background Structure in Document Images. Bunke, H. and Wang, P. S. P. and Baird, H. S. (Eds.), *Document Image Analysis*, World Scientific, Singapore, pp. 17–34, 1994.

[2] T.M. Breuel: Two Geometric Algorithms for Layout Analysis. In *DAS '02: Proceedings of the 5th International Workshop on Document Analysis Systems V*, Springer-Verlag, London, UK, pp. 188–199, 2002.

[3] T. Deselaers, D. Keysers, H. Ney: Classification Error Rate for Quantitative Evaluation of Content-based Image Retrieval Systems. In *ICPR 2004, 17th Int. Conf. on Pattern Recognition*, volume II, Cambridge, UK, pp. 505–508, August 2004.

[4] G. Ford, G.R. Thoma: Ground Truth Data for Document Image Analysis. In *Proceedings of the 2003 Symposium on Document Image Understanding and Technology*, pp. 199–205, April 2003.

[5] J. Hu, R. Kashi, G.T. Wilfong: Document Classification Using Layout Analysis. In *DEXA Workshop*, pp. 556–560, September 1999.

[6] J. Hu, R. Kashi, G.T. Wilfong: Document Image Layout Comparison and Classification. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR'99)*, pp. 285–288, September 1999.

[7] J.P. Ignizio, T.M. Cavalier: *Linear Programming*. Prentice Hall, 1993.

[8] D. Keysers, T. Deselaers, H. Ney: Pixel-to-Pixel Matching for Image Recognition using Hungarian Graph Matching. In *DAGM 2004, Pattern Recognition, 26th DAGM Symposium*, volume 3175 of *Lecture Notes in Computer Science*, Tübingen, Germany, pp. 154–162, August 2004.

[9] K. Kise, A. Sato, M. Iwata: Segmentation of Page Images using the Area Voronoi Diagram. *Comput. Vis. Image Underst.*, 70(3):370–382, 1998.

[10] D.E. Knuth: *The Stanford GraphBase: A Platform for Combinatorial Computing*. Addison-Wesley, Reading, MA, 1994.

[11] J. Liang, I.T. Phillips, R.M. Haralick: Performance Evaluation of Document Structure Extraction Algorithms. *Computer Vision and Image Understanding*, pp. 144–159, 2001.

[12] S. Mao, T. Kanungo: Empirical Performance Evaluation Methodology and its Application to Page Segmentation Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):242–256, March 2001.

[13] G. Nagy, S. Seth, M. Viswanathan: A Prototype Document Image Analysis System for Technical Journals. *Computer*, 7(25):10–22, 1992.

[14] L. O'Gorman: The Document Spectrum for Page Layout Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11):1162–1173, 1993.

[15] Y. Rubner, L. Guibas, C. Tomassi: The Earth Mover's Distance, Multi-Dimensional Scaling, and Color-Based Image Retrieval. *Proceedings of the ARPA Image Understanding Workshop*, pp. 661–668, May 1997.

[16] F. Shafait, D. Keysers, T.M. Breuel: Performance Comparison of Six Algorithms for Page Segmentation. In *Proceedings of the 7th IAPR Workshop on Document Analysis Systems*, Nelson, New Zealand, February 2006.

[17] K.Y. Wong, R.G. Casey, F.M. Wahl: Document Analysis System. *IBM Journal of Research and Development*, 26(6):647–656, November 1982.

[18] B.A. Yanikoglu, L. Vincent: Pink Panther: A Complete Environment For Ground-Truthing and Benchmarking Document Page Segmentation. *Pattern Recognition*, 31(9):1191–1204, 1998.