GROUP 4 COMPRESSED DOCUMENT MATCHING

DAR-SHYANG LEE, JONATHAN J. HULL
Ricoh California Research Center
2882 Sand Hill Road, Suite 115
Menlo Park, CA 94025
Email: {dsl,hull}@crc.ricoh.com

Numerous approaches, including textual, structural and featural, for detecting duplicate documents have been investigated. Considering document images are usually stored and transmitted in compressed forms, it is advantageous to perform document matching directly on the compressed data. A two-stage process for matching Group 4 compressed document images is presented. In the coarse matching stage, ranked hypotheses are generated based on compression bit profile correlations. These candidates are further evaluated using a feature set similar to the pass codes. Multiple descriptors based on local arrangement of the feature points are constructed for efficient indexing into the database. Performance of the algorithm on the UW database is discussed.

1 Introduction

As electronic document images become prevalent, efficient retrieval methods become increasingly more important. A common solution is to perform OCR followed by a text based search. Recently, alternatives to the text-based approach have been developed by extracting features directly from images, with the goal of achieving efficiency and robustness over OCR. An example of such a feature is word length. Using sequences of word lengths in documents as indexes, Hull identifies similar documents by comparing the number of hits in each image generated by the query[4]. Spitz maps alphabetic characters to a small set of character shape codes (CSC) which can be used to compile search keys for ASCII text retrieval[10]. CSC's can also be obtained from text images based on the relative positions of connected components to baselines and x-height lines, as used by Spitz for word spotting in document images[9]. Doermann, et. al. extend the application of CSC's to document duplicate detection by constructing multiple indexes using short sequences of CSC's extracted from the first line of text of sufficient length[2]. All of these methods are inherently text-line based. Line, word or even character segmentation need to be performed. The duplicate detection mechanism in DocBrowse is based on horizontal projection profiles[1]. The distance between wavelet coefficient vectors of the profiles represents document similarity. It is noted that this method out-performs the text-based approach on degraded documents and documents with small amounts of text.

Proceedings of the 3rd IAPR Workshop on Document Analysis Systems, Nov.4-6, 1998, Nagano, Japan, pp. 29-38.

Since the majority of document images in databases are stored in compressed formats, it is advantageous to perform document matching on compressed files. Not only does this eliminate the need for decompression and recompression, the reduced memory requirement makes commercialization more feasible. Matching compressed files of course presents additional challenges. For CCITT Group 4 compressed files, *pass codes* have been shown to contain critical information in identifying similar documents. In Hull's work, pass codes extracted from a small text region are used with the Hausdorff distance metric to correctly identify 92.5% of duplicate documents[3]. However, calculation of the Hausdorff distance is computationally intensive and the number of distance calculations scales linearly with the size of database. The computational cost can be reduced by measuring global similarities of pass code distributions. It has been shown that the number of pass codes inside the cells of a fixed grid can effectively retrieve visually similar documents, and can be used as a preprocessing step for the Hausdorff measure[5].

In this paper, we present a two-stage algorithm for duplicate detection of Group 4 (G4) compressed documents. The first stage performs coarse matching based on document profile correlation. Global statistics such as line spacing and text height are used to confine the search space. If multiple candidates are generated, a set of *endpoint* features is extracted from the query document for detailed matching. Similar to the pass codes, the endpoint features contain sufficient information for various levels of processing, including page skew and orientation estimation. In addition, endpoint features are stable, symmetric and easily computable from Group 4 compressed files.

The rest of the paper is organized as follows. Details of the coarse matching processing, including profile extraction, global statistics calculation and feature robustness are discussed in Section 2. Section 3 describes the detailed matching procedure which includes endpoint feature extraction and generation of local descriptors. Section 4 discusses experimental results and suggests further improvements, followed by conclusions in Section 5.

2 Coarse Matching Algorithm

The coarse matching process retrieves documents based on their profile similarities. Although a horizontal profile does not always contain sufficient information to uniquely identify a document, it is reasonable to assume that duplicate documents should have similar profiles. First, the *compression bit profile* is computed from the G4 compressed query image. Spectral analysis techniques are then applied on the bit

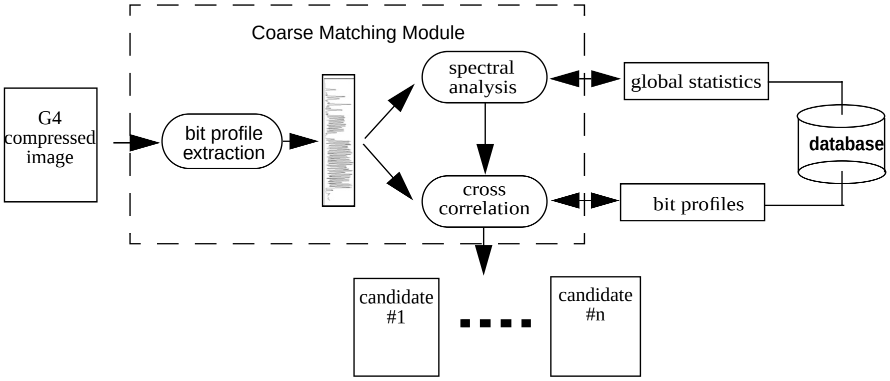


Figure 1: An overview of the coarse matching algorithm. Spectral analysis is first performed on the bit profile to get global statistics for indexing. Profile cross correlations against the selected set of images are used for hypothesis generation.

profile to generate robust global statistics for database indexing. The precomputed bit profiles of the selected candidates are correlated against the profile of the query image to produce a set of ranked hypotheses. Further processing may be avoided if a highly confident match is found by correlation. Figure 1 summarizes the coarse matching process.

The deterministic nature of G4 encoding leads to the expectation that the same image pattern will produce a similar compression ratio regardless of its location. In general, halftones require the most number of bits for encoding; texts require fewer bits, and background even fewer. For images which are text-dominant and oriented horizontally, the bit profiles should show peaks and valleys corresponding to text lines. For a set of point sizes commonly occurring in documents, the compression ratio for full page-width text lines is quite consistent, making them distinguishable from halftones, despite the fact that their ink densities may be similar. In contrast to the horizontal projection of ink density, the bit profile shows where the information really is. Large black regions often encountered at edges of photocopied documents will have almost no effect on the bit profile, whereas large peaks will be produced in an ink density profile. In fact, the bit profile will not look much different if the page is in reverse video. Obviously, this would be a serious disadvantage if large black regions carry significant meanings as in, for example, tables and forms.

Cross correlation is used to measure similarities between profiles. Global characteristics of profiles can be used to reduce the number of distance calculations. The periodic nature of bit profiles suggests spectral properties will be more useful than statistical moments. Intuitively, the dominant line spacing, the number of text lines and the location of text provide a good first-level characterization of a document. The dominant line spacing can be directly calculated from the highest peak in the power spectrum density. Although spectral analysis does not provide a quantitative measure of the number of text lines, the energy under the peak frequency is a good indication of the amount of text on the page. To estimate the location of the text lines, we apply a bandpass filter, centered at the dominant line spacing frequency, to the profile. The filtered signal will have large amplitude at text locations. Sections of the profile which are linear in phase correspond well to text blocks. We use the centroid of this text energy profile and the width of the 90% energy span as an estimation for text location and concentration. These two numbers, along with peak frequency and total text energy, are used to define a search window in the space of database images.

3 Detailed Matching

Since visually different documents can have similar compression profiles, a second stage process may be necessary to resolve any ambiguities. To obtain more information, a set of *endpoint features* similar to pass codes is extracted from the G4 image. After feature analysis, a subset of these features are identified as markers. Descriptors based on the positions of these markers are generated for document indexing. Cross validation is carried out against document candidates provided by the coarse matching procedure. This process is illustrated in Figure 2.

3.1 Endpoint extraction

In the Group 4 compression format[6], each scan line is encoded with respect to the line above. The starting points for two consecutive runs, referred to as *changing elements,* on both lines are identified at any time with respect to the current encoding point, a_0. Based on the relative positions among these *changing elements*, one of three possible modes, *horizontal*, *vertical* or *pass* mode, is selected for encoding. After encoding, a_0 is moved forward and the process is repeated. The process is reversed during decoding. Therefore, the *mode* information is decoded first, but positions of the changing elements are also maintained at all times.

It has been noticed that pass codes occur at locations corresponding to “bottoms of strokes” (*white pass*) or “bottoms of holes” (*black pass*). For Roman alphabets,

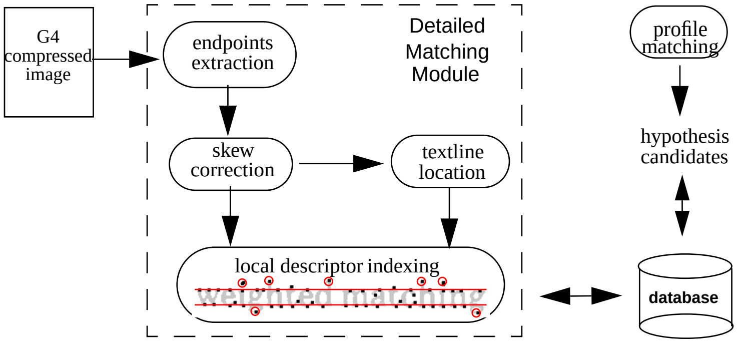


Figure 2: Summary of the detailed matching process. Skew corrected endpoint features in the ascender and descender regions are used for document indexing.

these feature points occur at the end of a downward vertical stroke or the bottom of a curved stroke. The alignment of these pass codes near baselines and the structural information they carry make them very useful in a variety of tasks such as skew estimation[8] and text matching[3]. Equally important is the fact that they can be extracted easily from a G4 compressed file.

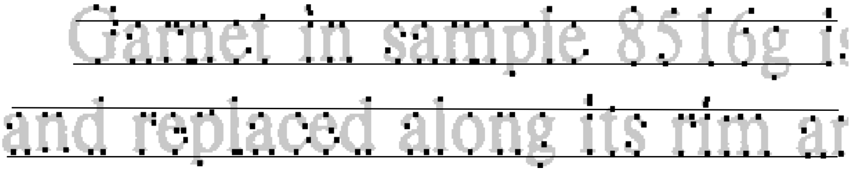
While pass codes are very informative, they have some drawbacks. First of all, they are unstable in the sense that while all *white pass* codes correspond to *bottoms of strokes*, not all *bottoms of strokes* will be represented by pass codes. As a result of the context-dependent nature of G4 encoding *modes*, identical local patterns of changing elements can be encoded as a pass code or part of a horizontal mode. Another limitation of pass codes is that they are asymmetric. While the bottom of a stroke or a hole is captured, there is no information about the top of the stroke or hole. For example, the bottom of a “d” often contains two pass codes, one white and one black, while no feature point on the top of the character is captured.

Based on these observations, we propose extraction of *endpoint* features directly from the changing elements. There are two sets of endpoints: *up* and *down* endpoints. *Down* endpoints are bottoms of strokes, the same as what white pass codes capture. However, they are extracted by directly comparing the positions of changing elements, eliminating the possibility of obscurity by horizontal encoding. All bottoms of strokes are *down* endpoints and vice versa. The tops of strokes are similarly extracted as *up* endpoints. Since endpoints are detected based on relative positions of changing elements, they are as easy to calculate as pass codes. Figure 3 shows an image segment and its corresponding endpoints. It is apparent that *down* endpoints align at the baseline while *up* endpoints align at the x-height line. This allows for information such as text height, page orientation and ascenders to be extracted. The symmetric nature of the *up* and *down* endpoints is also beneficial in dealing with inverted pages. If the page is inverted, the endpoints for the correctly oriented page can be obtained by switching the *up* and *down* endpoints followed by a simple coordinate remapping. There is no need for rescanning the compressed document.

3.2 Document Indexing

Following feature extraction, we convert the two dimensional endpoint information to a one dimensional representation for efficient indexing. Several simple steps are involved in this process. First, page skew is estimated and corrected using a process similar to that described by Spitz[8]. The smoothed horizontal projection profiles for the skew corrected *up* and *down* endpoints, which will be referred to as *U* profile and *D* profile, are used to locate text lines. Since x-height lines must be above their corresponding base lines, the *D* profile must lag behind the *U* profile. We calculate the maximum correlation between the U profile and D profile within an offset constrained by the dominant line spacing, which is obtained from spectral analysis of the profiles. In the correlated profile, wherever a local maximum in the *U* profile matches up with a local maximum in the *D* profile, separated by a distance equal to text height, there is a good possibility that a text line is located.

Given a set of text line locations, the endpoints within each textline zone are extracted. Since we also have the x-height line and baseline location, we can define the ascender and descender zones. With well-defined reference lines, there are several possibilities to encode endpoints as sequences. We observed that endpoints occurring inside the x-height zone are more susceptible to noise due to touching, fragmentation, serifs and font style variations. Therefore, endpoints in the middle zones are ignored. Only up endpoints above the x-height line and down endpoints below the baseline are used as markers. We use sequences of quantized distances between consecutive markers as descriptors. Negative values are used for distances between down endpoint markers to distinguish them from those of up endpoint markers. The left-most endpoint in each text line region is used as a reference point. To maintain the two dimensional structure, descriptors across text lines are concatenated, separated by a 0. Hence, a string of positive and negative values will be generated for given lines of text, as shown at the bottom of Figure 3. Alternatively, marker distances in the ascender and descender zones can be interleaved in strictly left to right order.



1, 11, 13, 4, 2, 2, 2, 4, 0, -39,
0,
5, 7, 7, 4, 8, 6, 0, -11

Figure 3: An example of endpoints within a two text line region. Endpoints in the ascender and descender zones are used as markers for index generation.

However, similar performances are observed.

Each document in the database is reverse indexed by sequences of *n* consecutive distances. Similarly, *k* sequences of *n* consecutive distances are formed during a test query. The weight for each descriptor is inversely proportional to the number of documents it indexes. Suppose *n* is 5 in the example of Figure 3, then *k*=15 sequences S_1=(1, 11, 13, 4, 2), S_2=(11, 13, 4, 2, 2), S_3=(13, 4, 2, 2, 2) ... S_{15}=(4, 8, 6, 0, -11) will be generated. Each of the *k* sequences, S_i, contributes a score of $1/(k*m_i)$ to every one of the m_i documents that S_i indexes. Documents that receive scores greater than a threshold are returned. Clearly, large *n* values will produce fewer, more unique descriptors. However, longer sequences are also more susceptible to disruption by noise.

4 Results and Discussions

Experiments are conducted on a set of 979 document images from the University of Washington (UW) database[7]. Of the 979 images, 292 images (146 pairs) have a matching counterpart. Each of the 292 images is used as a query for retrieving its counterpart from the remaining 978 images. The coarse and detailed matching procedures were tested independently as well as in combination. Results on each experiment will be presented.

In our implementation of the coarse matching algorithm, the original bit profile obtained at the vertical image resolution is down sampled by averaging to 36 dpi. Thus we produce 396 bytes (11 inch x 36 dpi x 8 bits) of data for a typical 8.5”x11" page. Cross correlation of the bit profiles produced 86.0% correct on top choice, and 95.2% correct on top 20 choices. Using the global statistics for indexing, the average number of candidates for cross correlation calculation is reduced by 90% without any loss in the recall rate. The Discrete Fourier Transforms of the bit profiles for images in the database are precomputed and stored, so cross correlation can be calculated by a vector product. Therefore, each image query involves extracting the bit profile, filtering by global statistics, followed by approximately 100 vector products of dimension 396.

In the detailed matching experiment, endpoints were extracted from a 1.5 by 1 inch region from the first body of text in the image using the ground truth information. The text line location algorithm was then applied to detect endpoints in the ascender and descender zone. Although some of those regions contained non-text portions of the image, we relied on the line location algorithm to eliminate any feature points not belonging to textlines. Once the ascender and descender zones were defined, a sequence of distances between endpoint markers was generated for each patch. Taking every 3, 4, and 5 consecutive distances as an index, multiple descriptors were constructed for a database query. Using the weighting scheme described above, 92.5% of the duplicates are correctly detected. This performance is comparable to the computationally intensive Hausdorff distance based method[3]. In addition, the indexing approach has much greater scalability than the distance based strategy.

In the combined test, we return the result of the coarse match if the correlation score of the top choice is greater than 0.85 and the difference between the top and second choice score is more than 0.03. Otherwise, the top twenty choices are passed on for detailed matching. As a result, 70% of the images are accepted after coarse matching, and only 30% of the images require detailed matching. The overall correct rate for the system is 93.8%. Therefore, coarse matching by profile correlation not only improves execution efficiency but also eliminates candidates which otherwise would be confused by detailed matching alone. Clearly, different results will be achieved by modifying the decision rule. We have not taken into account the problem of false alarm in our performance analysis. The effectiveness of the combination rule is contingent upon the assumption that we expect exactly one duplicate for each query. Lacking such restrictions, as in most practical cases, the detailed matching should always be invoked to improve the reliability of detection.

Since realistic timing of our prototype, which consists of research code written in C, Perl and Splus, is difficult, we provide a speed analysis based on hypothetical situation of a single query into a database containing one million documents. At 8 minutes per 978 images for optimized C code running on a 70MHz Sparc20[3], one million Hausdorff distance calculation would require one week of CPU time. For the proposed method, assuming the same 90% reduction is achieved by global statistics of document profile, correlation on 100,000 images is required, which would take approximately 2 seconds on the Sparc20. Adding on the time required for endpoints extraction, skew correction and indexing for the query mage, we estimate the overall time to be around 5 seconds, roughly 5 orders of magnitude improvement in speed.

We analyzed the errors made by both stages of the system and found few surprises. Excluding errors introduced by scale differences and non-linear distortions, which can not be handled by the proposed method, the most common errors resulted from skewed images and misaligned columns. Page rotation has the effect of locally averaging horizontal projection profiles, making the peaks and valleys less prominent. Without any estimate or correction for page rotation, the coarse matching algorithm has no correction for document skew. Although text line location, based on horizontal profiles of skew corrected feature points, is skew tolerant, it has difficulty handling multicolumn pages. Non-colinear columns can lead to aliasing and incorrect line spacing estimation and text line location.

Several improvements, which are subject to further research, are possible. The skew corrected endpoint projection profile, which is used for text line location, displays similar characteristics to the bit profile, and it can be used for coarse matching. However, endpoint profiles are spiky and the results are sensitive to the smoothing parameters. To improve the robustness of text line location in documents with non-colinear columns, one solution is to use a vertical projection profile for column segmentation. Another solution is to perform text line location within vertical slices of the document, and use only the high confidence results in hope of avoiding column boundaries. Obviously, such analysis is still vulnerable to documents with complex layouts. Moreover, spurious feature points occurring beyond text line boundaries can generate false descriptors. Some measures for detecting the horizontal extent of text lines should be provided. Since the feature points have been skew corrected and the positions of the x-height lines and baselines are known, finding the ends to such line segments should be relatively trivial. Furthermore, the regions for descriptor generation should be automatically determined. In our experiment, we used ground truth information for identifying corresponding text regions in document images. This registration process should be replaced by an automatic region selection scheme. Generating descriptors for every located text line will increase the database size and reduce precision. Some criteria for identifying candidate regions should be investigated. One possibility is to base the selection on local feature point densities.

5 Conclusions

We described a two-stage process for detecting duplicate documents in Group 4 compressed images. Coarse matching generates ranked hypotheses based on profile similarities. Global statistics obtained from spectral analysis of profiles can be used to confine the search space. If no high confidence match is found, multiple candidates are further evaluated by a detailed matching process utilizing a set of endpoint features directly computable from the Group 4 decompression scheme. Descriptors based on sequences of distances between endpoint markers provide efficient indexing to the database. Experiments on the UW database showed 93.8% correct rate in detecting duplicates.

6 Acknowledgments

The authors are grateful to the reviewers for many thoughtful comments.

7 References

1. V. Chalana, A. Bruce, and T. Nguyen, “Duplicate document detection in DocBrowse”, SPIE Conference on Document Recognition V, pp. 169-178, 1998.
2. D. Doermann, H. Li, O. Kia and K. Kilic, “The Detection of Duplicates in Document Image Databases”, Technical Report CS-TR-3739, University of Maryland, 1997.
3. J. J. Hull, “Document Matching on CCITT Group 4 Compressed Images”, SPIE Conference on Document Recognition IV, pages 82-87, 1997.
4. J. J. Hull, “Document image matching and retrieval with multiple distortion-invariant descriptors”, Proceedings of DAS, pages 383-400, 1994.
5. J. J. Hull, “Document image similarity and equivalence detection”, International Journal on Document Analysis and Recognition, Vol. 1, No. 1, pp.37-42, 1998..
6. R. Hunter, A. H. Robinson,” International Digital Facsimile Coding Standards,” Proceedings of the IEEE, Vol. 68, No. 7, pp. 854-867, 1980.
7. I. T. Phillips, S. Chen, R. M. Haralick, “CD-ROM document database standard”, Proceedings of the 2nd ICDAR, pp. 478-483, 1993.
8. A. L. Spitz, “Skew determination in CCITT group 4 compressed document images,” Proceedings of SDAIR, pp. 11-25, 1992.
9. A. L. Spitz, “Using character shape codes for word spotting in document images”, Shape, Structure and Pattern Recognition, pages 382-389. World Scientific, 1995.
10. A. L. Spitz, “Using character shape coding for information retrieval”, Proceedings of the 4th ICDAR, pp. 974-978, 1997.