Discrete Point Based Signatures and Applications to Document Matching

Nemanja Spasojevic[1], Guillaume Poncin[2], and Dan Bloomberg[3]

Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA
sofra@google.com[1],gponcin@google.com[2],dbloomberg@google.com[3]

Abstract. Document analysis often starts with robust signatures, for instance for document lookup from low-quality photographs, or similarity analysis between scanned books. Signatures based on OCR typically work well, but require good quality OCR, which is not always available and can be very costly. In this paper we describe a novel scheme for extracting discrete signatures from document images. It operates on points that describe the position of words, typically the centroid. Each point is extracted using one of several techniques and assigned a signature based on its relation to the nearest neighbors. We will discuss the benefits of this approach, and demonstrate its application to multiple problems including fast image similarity calculation and document lookup.

Keywords: image processing, feature extraction, image lookup

1 Introduction

Over the past decade vast amounts of digitalized documents have become available. Projects like Google Books and Internet Archive have brought millions of books online, extremely diverse in form and content. Such a corpus requires fast and robust document image analysis. This starts with image morphology techniques, which are very effective for various image processing tasks such as extraction of word bounding boxes, de-skewing, connected component analysis, and page segmentation into text, graphics and pictures [1].

Image feature extraction is a well studied problem. Many techniques like SURF [2] and SIFT [3], [4], FIT [5] perform well at point matching across images, and image lookup from a database. However these techniques do not fare as well on repetitive patterns, such as text in document images. In addition, both SURF and SIFT extract thousands of key features per image, and features are matched by nearest neighbor search in the feature space which requires sophisticated indexing mechanisms. Unlike images of 3D objects, document images are projections of 2D image (paper) on the 3D scene (with significant warping caused by camera proximity and paper curvature). Locally Likely Arrangement Hashing (LLAH) [6] exploits this by making signatures from affine invariant features. This was shown to work well and achieves high precision in document page retrieval from a corpus of 10k pages. LLAH uses discrete features which are directly used for indexing, thus simplifying the process. Other retrieval techniques use word shape coding [7], which should be robust to image deformation, but still suffer significant degradation in performance on non-synthetic document images.

In this paper we describe a novel method for extracting signatures based solely on the pixels of the page, and show how it can be applied to a range of problems in document matching. Our signatures are created from either centroids of the words, or centers of word bounding boxes. Because of this, we can use them to match documents for which we only have word bounding box information. The number of signatures extracted per page is the same as the number of words, which results in an order of magnitude less features than previous techniques.

We will demonstrate the use of these signatures in two applications: page similarity detection and image lookup from a database of indexed images. This work complements similar techniques based on OCR'd text to find similar pages or similar regions between two books, but works independently of language. In particular it can be applied to documents where OCR usually performs poorly (Chinese text, old text, etc.).

2 Algorithm Overview

In this section we cover point cloud extraction from the raw image and creation of signatures. We also describe two possible improvements: filtering high-risk signatures and superposition for ambiguous signatures.

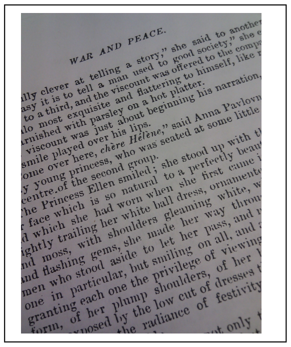
2.1 Word Position Extraction

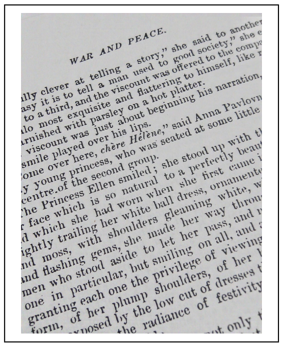
In order to perform signature extraction on a point cloud, the first step is to extract word centroids from the document image. For this, we use an open source image processing library named *Leptonica* [8]. We use a number of steps, including conversion of the raw image (Fig. 1a) to grayscale, background normalization (Fig. 1b), binarization (Fig. 1c), deskewing (Fig. 1d), and a series of morphological operations to extract the connected components representing words (Fig. 1e). The final step is to extract the centroids (Fig. 1f). This approach assumes that the document image has either vertical or horizontal text lines, which is typically the case in printed material and digital books (or we can correct for).

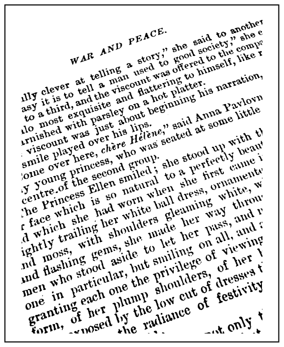
The benefit of operating on word positions is that it only requires the word bounding boxes, not the image. This way we can use, PDF or OCR word bounding box information as a source of points, which greatly reduces the cost of computing signatures when working with a large number of pages.

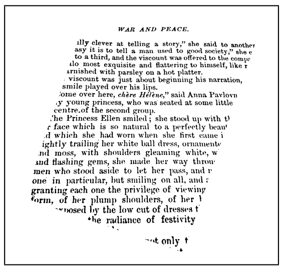
2.2 Signature Calculation From Point Cloud

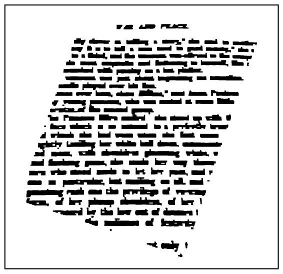
An overview of signature extraction is shown in Algorithm 1. The basic idea is, for each point, to select $kNNCount$ nearest neighbors, sort them in radial distance order, and compute the angle between the selected and neighbor point. The angle is discretized based on how many bits we decide to use. Discretized


(a) Raw


(b) Background normalized gray


(c) Binary


(d) Deskew


(e) Word-like connected components

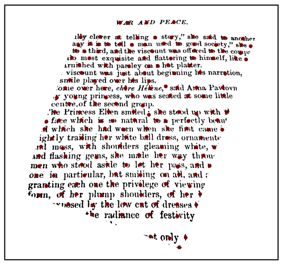

(f) Word-like connected components centroids

Fig. 1: Examples of image processing steps while extracting the word centroids.

angles (sub-signatures) are concatenated together in order of radial distance, creating the final signature. An illustration of this process is shown on Fig. 2. In this paper a single angle is represented by 4 bits, i.e. using buckets of 22.5°.

Distortions such as skew are corrected for during word extraction. Other types of distortion like severe warping due to lens distortion or paper curvature are more challenging to deal with. However, the signature extraction algorithm achieves a reasonable degree of robustness to those. The main failure mode for matching signatures is actually word segmentation errors leading to changes in the neighborhood of a point. For example the merging of two words into one causes a word centroid to be shifted and another one to be missing. Flips in discretized values of the angle or reordering of the radial distances can also modify the signature. However in practice, such failures have limited impact on the result since they only affect small regions.

2.3 Signature Filtering Based on Estimated Risk

The idea is to only keep signatures that are stable with high confidence. We filter out signatures with high probability of bits shifting or flipping. Each signature

Algorithm 1 Signature calculation from point cloud

```
 1: $kNNCount \leftarrow 8$                     ▷ How many nearest neighbors we care about
 2: $kBitPerAngle \leftarrow 4$       ▷ How many bits per neighbor to neighbor we dedicate
 3: $kMask \leftarrow 1 << kBitPerAngle$
 4: $points$                  ▷ Word positions for a given image (image or word box based)
 5: $signatures \leftarrow \emptyset$
 6: for all $point \in points$ do
 7:     $nn\_points \leftarrow NearestNeighbors(point,\ kNNCount)$
 8:     $nn\_points \leftarrow SortByRadiusInIncreasingOrder(point,\ nn\_points)$
 9:     $signature \leftarrow 0$
10:     for all $neighbor \in nn\_points$ do
11:         $alpha \leftarrow CalculateAngle(point,\ neighbor)$
12:         $alpha\_discrete \leftarrow floor(kMask \times alpha/2\_PI)$
13:         $signature \leftarrow signature << kBitPerAngle$
14:         $signature \leftarrow signature \mid alpha\_discrete$
15:     end for
16:     $signatures.push\_back(signature)$
17: end for
```

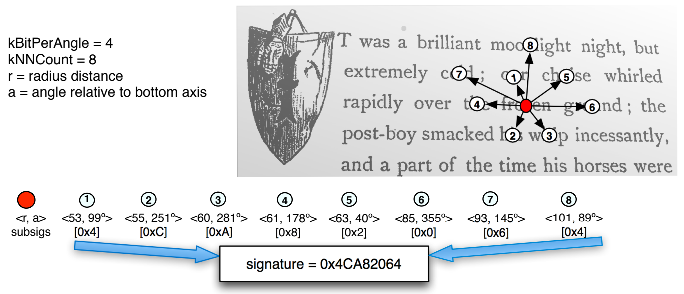

kBitPerAngle = 4
kNNCount = 8
r = radius distance
a = angle relative to bottom axis
T was a brilliant moonlight night, but extremely cold; our chaise whirled rapidly over the frozen ground; the post-boy smacked his whip incessantly, and a part of the time his horses were
1
2
3
4
5
6
7
8
<r, a>
subsigs
<53, 99º>
[0x4]
<55, 251º>
[0xC]
<60, 281º>
[0xA]
<61, 178º>
[0x8]
<63, 40º>
[0x2]
<85, 355º>
[0x0]
<93, 145º>
[0x6]
<101, 89º>
[0x4]
signature = 0x4CA82064

Fig. 2: Overview of signature creation from word centroids

S is composed of smaller sub-signatures $S = [s(0)][s(1)]..[s(N)]$. We consider signature variations that comes from slight shifts of word positions. Small shifts may lead to changes in discretized angle value, e.g. $s(0)$ flipping from 13 to 14 due to small word position shifts, or in the order of sub-signatures, e.g. $s(0)$ and $s(1)$ swapping as they had almost same radial distance. If we can estimate the confidence of a given signature, we can filter out weaker ones.

Let's consider the probability of a discretized angle flipping. If we have angle α and its discrete value a, where for example $a \leq \alpha < a + 1$ then we can say that distance of from the edge is $\epsilon = \mid \alpha - (a + 0.5) \mid$ assuming that $a + 0.5$ is edge of discretization. One can easily see that if $\epsilon = 0$ a random perturbation of points will lead to a flip in 50% of the cases. If we say that probability of flip is $p(\epsilon)$, then the probability that the entire signature changes due to at least one sub signature flipping can be expressed as:

$$P_{flip} = 1 - \prod_{i=1}^{N} (1 - p(\epsilon_i)) \tag{1}$$

Similarly we can estimate the probability of two neighboring points swapping as $p(\delta r)$ where $\delta r = \mid r_1 - r_2 \mid / \; (\frac{1}{2}(\mid r_1 \mid + \mid r_2 \mid))$ is the relative radial distance between consecutive neighbor points from the choosen point. The probability that the signature changes can be expressed as:

$$P_{swap} = 1 - \prod_{i=1}^{N-1} (1 - p(\frac{\mid r_i - r_{i+1} \mid}{\frac{1}{2}(\mid r_i \mid + \mid r_{i+1} \mid)})) \tag{2}$$

Finally the chance of a signature changing due to swap or flip is $P_{flip\_or\_swap} = 1 - (1 - P_{flip}) \times (1 - P_{swap})$.

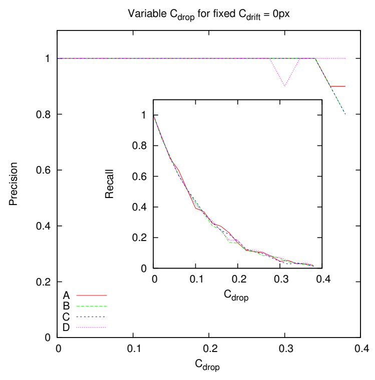
In this paper we naively modeled the probability distribution as $p(x, w) = 0.5 \times (w - x)$ if $x \in [0, w)$ and 0 otherwise. x is variable, while w is a threshold parameter. For angular discretization $w_{angle} = 0.05$, and for radius risk $w_{radius} = 0.01$. In experiments where we use signature filtering, signatures with $P_{flip\_or\_swap} > 0.6$ are filtered out.

2.4 Superposition of Ambiguous Signatures

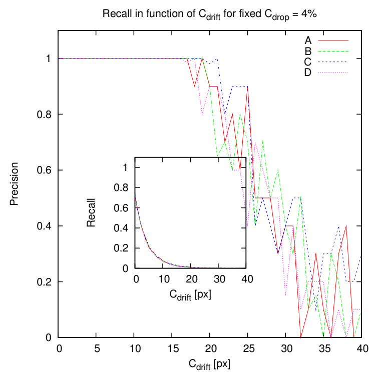
Let us consider the problem of angle discretization. We often end up on one side or the other side an edge in the discretization function. One option is to use both values when composing the signature. We can consider a signature to be a superposition of states (angles), and by calculating the signature we project mixtures to their discrete values. But we can also create all possible projections; i.e. the set of all signatures. For example if the 1st and 3rd sub-signatures have two possible states then we have 4 possible signatures. For example $[\{s_1, s'_1\}][s_2][\{s_3, s'_3\}][s_4]$ would lead to $\{[s_1][s_2][s_3][s_4],\ [s_1][s_2][s'_3][s_4],\ [s'_1][s_2][s_3][s_4],\ [s'_1][s_2][s'_3][s_4]\}$. In the following, superposition was used only where indicated for angles within $\epsilon < 0.05$ of a discretization edge.

3 Evaluation on Synthetic Data

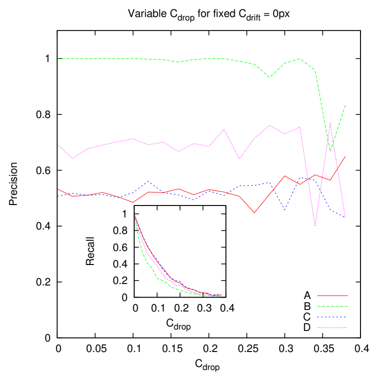
In this section we evaluate precision and recall of signatures for the task of matching two pages. The evaluation is done on synthetic data. All the signatures used were 32bit (generated based on 8 nearest neighbors). We start from an 'original' point cloud and a 'copy' point cloud derived from the original with some added distortions. All points are within $H = 1200$ pixels, $W = 1600$ pixels, and we use a fixed number of points per page for the original page ($N_{original} = 300$) (on average one point per 80×80 pixel square). The 'copy' page then has ($N_{copy} = N_{original} \times (1 - C_{drop})$ points where $N_{original} \times C_{drop}$ points are



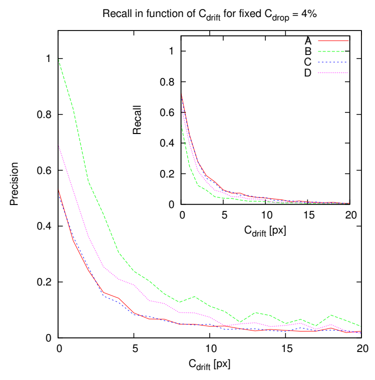
(a) Recall in function of C_{drop} for fixed $C_{drift} = 0$



(b) Recall in function of C_{drift} for fixed $C_{drop} = 4\%$



(c) Recall/precision in function of C_{drop} for fixed $C_{drift} = 0$



(d) Recall/precision in function of C_{drift} for fixed $C_{drop} = 4\%$

Fig. 3: Signature precision/recall evaluation for the random (3a, 3b) and grid (3c, 3c) point cloud distribution. Four scenarios: (A) original signatures, (B) risky signatures filtered, (C) superposition of ambiguous signatures, (D) only using unique signatures

randomly be dropped to simulate word segmentation errors. Each copy point is also moved from its original position as follows:

$$(x_{copy}, y_{copy}) = (x_{original} + rand() \times C_{drift}, y_{original} + rand() \times C_{drift}) \quad (3)$$

where $rand()$ is random float from [0 , 1). In this way we try to emulate realistic differences that may occur between two sets of point clouds.

We experimented first with randomly distributed point clouds, and then with points aligned on a square grid (e.g. Chinese texts) with small x and y variations (less than 5 pixels). These two sets are called *random* and *grid* set.

Random Distribution: on Fig. 3a, 3b we see that precision for this set remains 100% even as the amount of drop increases. This is expected, because signatures in this case are fairly unique and can withstand some amount of perturbations. For variable drop at $C_{drop} = 10\%$ recall is 40% (Fig. 3a), while for variable drift (drop fixed at 4%) at $C_{drift} = 5\%$ pixels recall recall is 20% (Fig. 3b). In both cases recall shows a steep drop, but there is significant robustness to points being dropped and shifted around. Also it's interesting to note that there is practically no difference in results between scenarios A-D.

Grid Distribution: This is probably the hardest type of distribution for our algorithm. It is so regular that unrelated signatures are likely to collide since neighborhoods all look similar. This is verified in Fig. 3d, where starting precision is 50% (curve A). This means that many unrelated signatures match across two identical point clouds. Note that precision is constant as we randomly drop points, and it seems to perform best in the case where we filter out weak signatures, but also pretty well when we filter out signatures that occur multiple times on one page. However in Fig. 3d, 3c we see that increased precision when filtering comes at the expense of recall.

4 Document Image Similarity Application

A common application of analysis is the detection of similar pages. For instance, one may want to cluster duplicate pages when merging multiple scans of a document, or find corresponding pages between 2 different versions. This can be done based on OCR text, but OCR is an expensive operation, which does not work well on all scripts. Document digitization is typically done using sheetfed scanners or other techniques in which acquired images have little warping, moderate skew ($\pm 3°$), no scale difference, and small translation variation. These are optimal conditions for our signatures, although we have shown how we could correct for imperfect conditions. Jaccard J similarity is used to estimate document image similarity. The similarity between two pages is calculated as:

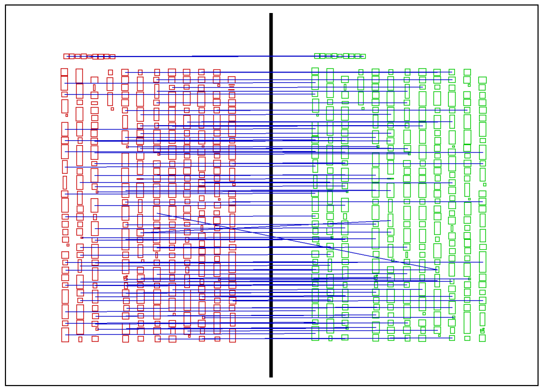
$$J_s(p_1, p_2) = \frac{\mid S(p_1) \cap S(p_2) \mid}{\mid S(p_1) \cup S(p_2) \mid} \tag{4}$$

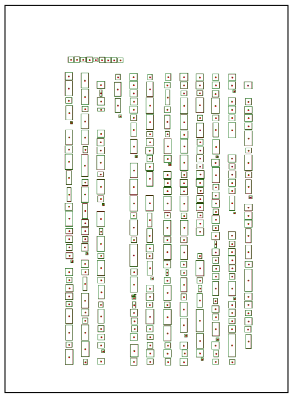
where $S(p)$ is the set of extracted signatures from page p. Depending on image distortion and word segmentation discrepancy between pages, the number of matching signatures may be low compared to the total number of signatures. We can get better estimate using matching signatures to calculate an affine transform from one image to another, and then use that transform to align word bounding boxes between pages. We declare that boxes match when their centroid is close after transformation and they have roughly the same size. From the matching box count we calculate similarity J_b as:

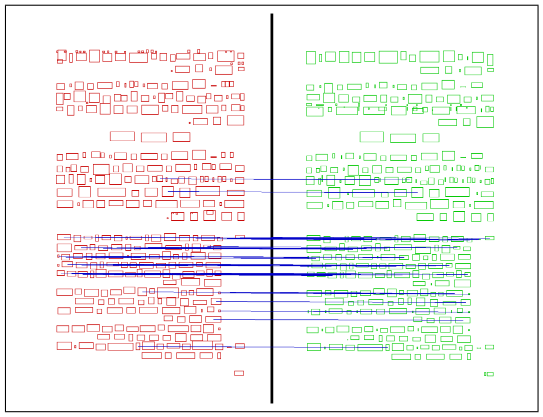
$$J_b(p_1, p_2) = \frac{\mid MatchCount(p_1, p_2) \mid}{\mid BoxCount(p_1) + BoxCount(p_2) - MatchCount(p_1, p_2) \mid} \tag{5}$$

In Fig. 4 we show both signature matches and box matches. The two example pages we use are in Chinese and Arabic. Initially few signatures match (Fig.

4a, 4c), but the number improves significantly after alignment. Similarities were initially 19% for Chinese, and 5% for Arabic. After alignment and similarity recalculation, we measured 93% for Chinese and 37% for Arabic. Note that in Arabic we still fail to align many boxes due to inconsistent word segmentation, but we align enough to have confidence in the measurement. With Chinese, similarity is high because segmentation is consistent despite being wrong.


(a) Matching signatures (Chinese text)


(b) Matching boxes (Chinese text)


(c) Matching signatures (Arabic text)

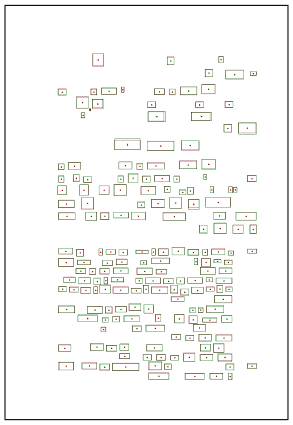

(d) Matching boxes (Arabic text)

Fig. 4: Examples of matching signatures and aligned boxes in the page similarity calculation. Shown on example of Chinnese (4a, 4b), and Arabic (4c, 4d) scripts.

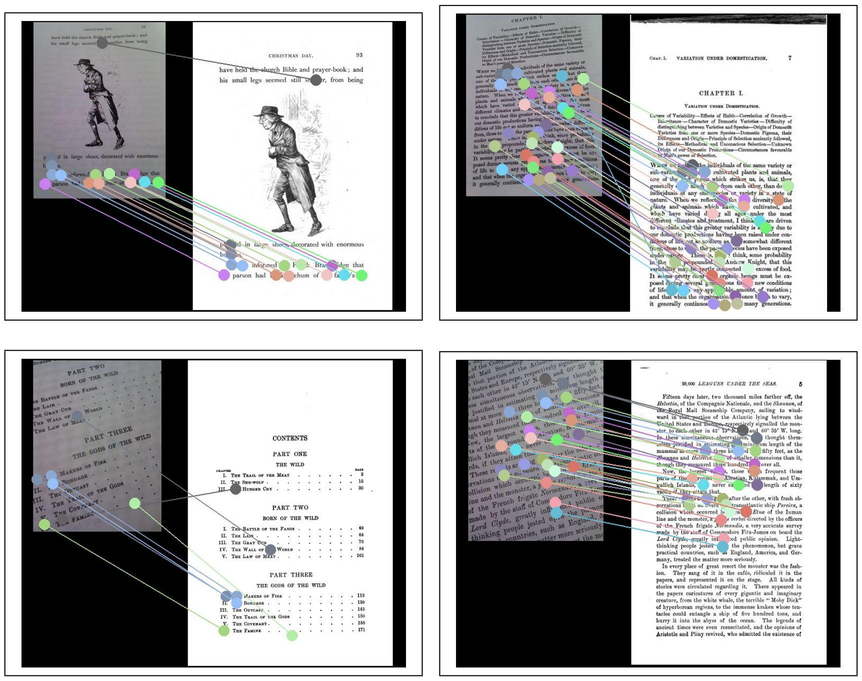


Fig. 5: Examples of image lookups

5 Document Image Lookup Application

Document retrieval based on a photograph of a page is another common application. For example, a user takes a photo with his mobile phone, sends it to a server, and the digital reference to the page is returned. We ran two experiments. The first with a small set of English language classics for a total of 4.1K pages. The second with a much wider variety of books, totalling 1M pages. In both cases, we queried for 120 photographs of book pages taken with a NexusOne Phone camera (5MP). Our server runs on a 2.2GHz PC with 8GB of RAM.

The pipeline is straightforward. We built an index based on OCR bounding boxes since we had them available (instead of running the image processing). The signature calculation and indexing takes on average 8.6ms per page (for 32bit signatures, for the 4.1K set). The index maps the signature to a list of $< book, page, position >$ tuples. At query time, we first process the input image using Leptonica to extract word centroids from which we generate signatures. We look up each signature in the index and bucket hits per book page. Finally we pick the page with the largest number of matching signatures. We only need to look up around 200 signatures, so queries run very quickly.

Overall accuracy exceeds 95% on the 4.1K set, as shown in Table 1. We tried various signature sizes and found that they had little impact on the results

Table 1: Image lookup results

Index Size	Accuracy	Signature size [bits]
4.1K	0.966	16
4.1K	0.949	32
1M	0.871	32

themselves. The downside of small signatures is that lookup takes longer because we run into many more false positives. Lookup times are sub-millisecond, once we have the list of query signatures.

We then ran the same experiment on a set of 1M pages, which is orders of magnitude larger than other sets that we know of, e.g [6],[5]. Accuracy degraded only a little to 87%, demonstrating that this technique works well with tens of thousands of books worth of data in the index. The index is also compact: 1M pages resulted in about 386M signatures, allowing us to store the entire index in memory on a single machine ($< 4GB$). We filtered out 0.8% of signatures, which were shared by over 1000 pages, to prevent false positives. On average each query results in about 2000 candidate matches, most of which agree.

6 Conclusion

We have presented a method for extracting signatures from document pages, that works using bounding boxes found from either OCR, or image processing independent of either language or the age of the document. We showed the robustness of this technique on synthetic data, including post-processing to improve the results in various cases. We then presented two simple applications of these signatures: one related to page similarity measurement, and the other to the retrieval of documents from camera pictures on a large corpus. The latter was shown to be efficient on a large corpus of 1M pages.

References

1. Bloomberg, D., Vincent, L.: Document Image Analysis, Mathematical morphology: theory and applications. ed. Najman L., Talbot H., 425–438 (2010)
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. LNCS (Proc. ECCV06), V.3951, 404–417 (2006)
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints, IJCV 60(2), 91–110 (2004)
4. Ke, Y., Sukthankar, R.: PCA-SIFT: A More Distinctive Representation for Local Image Descriptors Proc. CVPR'04, 506–513 (2004)
5. Liu, Q., Yano, H., Kimber, D., Liao C., Wilcox, L.: High accuracy and language independent document retrieval with a fast inv. t., Proc. ICME'09, 386–389 (2009)
6. Nakai, T., Kise, K., Iwamura, M.: Hashing with Local Combinations of Feature Points and Its App., Proc. CBDAR05, 87–94 (2005).
7. Shijian, L., Linlin L., and Chew Lim, T.: Document Image Retrieval through Word Shape Coding, IEEE TPAMI, V.30 N.11, 1913–1918 (November 2008)
8. http://www.leptonica.org/