Toward Massive Scalability in Image Matching

Jorge Moraleda and Jonathan J. Hull

Ricoh Innovations Inc. Suite 115, 1882 Sand Hill Road, Menlo Park, CA 94025, USA
E-mail: {moraleda,hull}@rii.ricoh.com

Abstract

A method for image matching from partial blurry images is presented that leverages existing text retrieval algorithms to provide a solution that scales to hundreds of thousands of images. As an initial application, we present a document image matching system in which the user supplies a query image of a small patch of a paper document taken with a cellphone camera, and the system returns a label identifying the original electronic document if found in a previously indexed collection.

Experimental results show that a retrieval rate of over 70% is achieved on a collection of nearly 500,000 document pages.

1. Introduction

We introduce a novel method for converting image search problems into text search problems, thus leveraging existing research and tools in this well developed field. We have implemented our system on top of the open source *Lucene* text indexing engine [3].

The only requirement of our method is that the locations of image features be *linearly orderable*. This can be achieved by obtaining a standard orientation for images, for example by making the line between eyes horizontal in faces, the sides of buildings vertical, or lines of text horizontal in document images.

In this paper we explore an application to partial document images taken with a cellphone camera. These cameras do not typically have macro lenses, and thus OCR is infeasible due to blurriness (Fig. 1). Document recognition based on a patch of text has many potential uses by enabling not only identification of the electronic source document, but also access to related electronic resources. We demonstrate the scalability and performance of our system in a collection containing almost 500,000 pages.

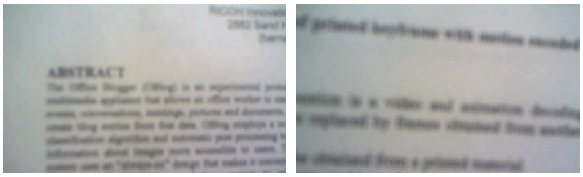

Figure 1. Typical frames from a cellphone preview video stream that can be recognized by our system. The actual resolution is 176x144 pixels.

Our method works by transforming feature descriptors to words in a synthetic language. The order of the words in the synthetic text document is used to perform geometric verification leveraging the ability of text retrieval engines to search efficiently for approximate phrases [3].

We remark that image noise results in most of the synthetic words in the query image being *misspelled* in *unnatural* ways. Thus natural language models for term expansion and string similarity models are inappropriate and custom models are required.

We also introduce novel feature descriptors suitable for document images based on word bounding boxes. Bounding boxes have previously been identified as suitable for computing descriptors in document images [4], especially in blurry ones. We introduce a unifying framework that considers all feature descriptors based on word bounding boxes as encoding piecewise linear paths connecting the word bounding boxes. This framework has enabled us to identify two new descriptors that provide superior accuracy to those presented in previous work.

2. Related Work

A number of techniques such as SIFT [7] and SURF [1] have been published for the problem of image matching using local descriptors extracted from

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.836

high resolution natural images. Two techniques specific to text documents are Document Image Decoding, an alternative to optical character recognition based on modeling imaging as a noisy communication channel [5], and LLAH which uses projective invariant features and scales to collections of 10,000 images [8]. Compared to our method the above techniques are computationally expensive and, in the case of document images, require high resolution images. In addition LLAH requires full page query images.

Erol et al. [2] demonstrate that word bounding boxes are features for document image matching that can be recognized reliably even using partial, low resolution blurry images; and introduce BWC, a descriptor based on them. We present a comparison of our approach and theirs in Section 5.

3. Synthetic Text Encoding

We index images by converting the feature vectors in the image to a (long) paragraph of text written in a synthetic language (as opposed to a natural language), which we place in a full text index. During the retrieval phase, feature vectors are extracted from an image, converted to synthetic text using the same process as in the indexing phase, and searched for in the full text index.

Conversion of feature vectors to synthetic words is accomplished via quantization. We name the word resulting from quantization of a vector the *canonical term* for that vector. The order of the words in the paragraph is determined by the order of the features in the image. Hence the requirement for linear orderability of features in an image.

3.1. Image Noise

Noise in the query images will translate into errors in the feature vectors, and in turn into spelling errors in the corresponding synthetic words. These errors will likely occur for feature vector components that lie near quantization boundaries. To address this problem, each term in the text search is given multiple alternative spellings. This is known as *term expansion*. In our implementation each vector gets indexed once using the its canonical term and the term expansion occurs at recognition time as follows. For each query vector:

1. Quantize the vector: Each element becomes one latin character. This is the *canonical* term. (See Fig. 2-a.)
2. Generate alternative spellings for the prefix (initial characters) of the canonical term for those entries in the top of the feature vector that are close to the quantization boundaries. (See Fig. 2-b.)
3. Perform a search in the index for all terms beginning with each of the alternative prefixes.
4. Score each matching term found in the index based on how many characters match, or nearly match the canonical term.

Steps 2 and 3 leverage the lexical ordering of inverted indexes. The alternative, scoring each term in the full index for similarity with the canonical term, would be prohibitively expensive. We have found that prefixes of length 9 to 12, and up to 2 character changes with respect to the canonical term's spelling produce the highest recognition accuracy for the descriptors we use. Shorter prefixes slow down retrieval significantly as too large a fraction of the index must be scanned, while longer ones yield too few features in patches of text containing a small number of lines.

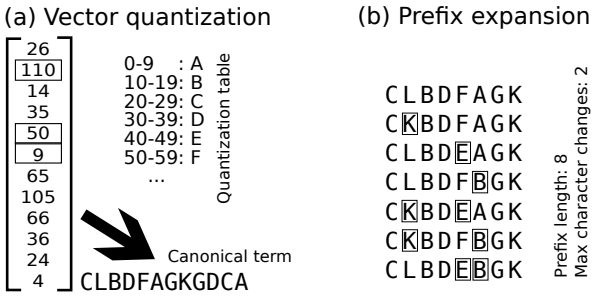

Figure 2. (a) Example of quantization of a vector into a synthetic word. (b) Prefix expansion. Notice that elements are quantized to more than one character when their value falls near the quantization boundary.

3.2. Geometric Verification

The image in the collection that best matches a given query image depends on the scores of each of the matching terms as well as their locations in the query image paragraph and the indexed image paragraph. We have modified the phrase query in the text search engine to compute the score for each indexed image for which matching terms of sufficiently high score appear in the query as follows.

1. Create a directed graph where vertices represent matching terms, and each edge linking two vertices indicates that both terms represented by the vertices appear in the same order in the query paragraph and the index paragraph, with no matching terms between them. (See Fig 3.)

2. Label each vertex with the match score and each edge with a gap penalty score that is increasingly negative the further apart both terms are in the index and query paragraphs.
3. Use dynamic programming to determine the score of the highest scoring directed path (vertex scores plus edge scores) which becomes the score for the indexed image.

This graph can be computed efficiently as the transitive reduction of the partial order: $a < b \equiv$ terms a and b appear in the same order in the query paragraph and the index paragraph.

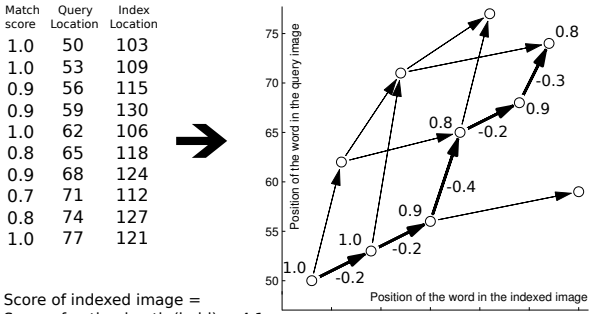

Figure 3. Example of geometric consistency evaluation to score one document image: 11 matching terms with high scores where found for a given indexed image, but only 6 were determined to be inliers after geometric verification. Only scores on the highest scoring path are shown in the graph.

4. Bounding Box Descriptors

We propose a unifying framework for describing and computing descriptors based on word bounding boxes in a document image. A descriptor is computed for every bounding box in the image. In this framework, a descriptor consists of a set of straight segments connecting the centers of bounding boxes, as well as a scale and rotation independent encoding of properties of the segments or of the bounding boxes that connect them. Any descriptor based on bounding boxes can be considered a particular case of this framework. We introduce two new ones, *Zig-Zag* coding and *Spiral* coding. We compare their performance in Sec. 5.

Fig. 4 introduces *Zig-Zag* a descriptor in which the segments form a downward left-right zigzagging lane starting at the bounding box whose descriptor we are computing. Fig. 5 introduces *Spiral* coding, a descriptor in which the segments form a spiral that wraps around the bounding box whose descriptor we are computing.

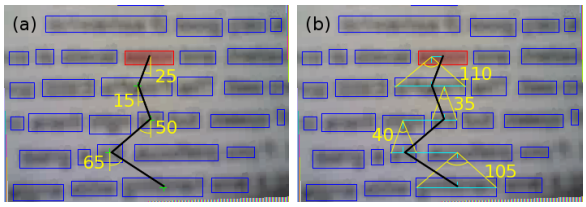

Figure 4. *Zig-Zag* coding. For each bounding box (in this example the one shaded), the segments of the descriptor form a unique zigzagging downward path starting at the given bounding box. Each segment of the path is encoded as two components of the descriptor vector. Thus in this example the 4-segment path yields feature vector [25, 110, 15, 35, 50, 40, 65, 105] which has 8 components.

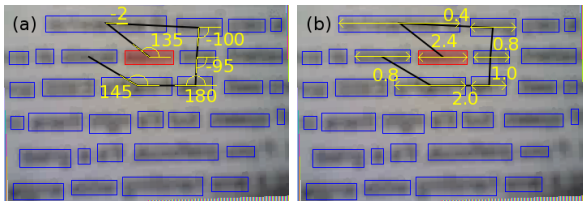

Figure 5. *Spiral* coding. For each bounding box, the segments of the descriptor form a unique spiraling path around the given bounding box. Each segment of the path is encoded as two components of the feature vector. Odd components represent the orientation of the segments (a), while even components are the ratio of the widths of the bounding boxes connected by those segments (b). This encoding is different from the one used in *Zig-Zag*.

In our system the bounding boxes of an image are stored in an R*-Tree which enables efficient computation of feature vectors. It is worth noting that in this framework not all descriptor vectors in one image necessarily have the same length. Also the framework can naturally be extended to use more than one feature vector per bounding box, by encoding two or more paths for each box. We explore the performance gains of this technique in the next section.

5. Experimental evaluation

We have implemented the descriptor computation in a Treo 700w smartphone with a 312Mhz processor. Feature vectors can be computed at a rate greater than 10 frames per second, suitable for real time processing and user feedback in the preview video stream.

To test our system, we used 8276 query images with a resolution af 176x144 (same as Treo 700w pre-

Figure 6. Performance comparison between *Zig-Zag* and *Spiral* coding and BWC [2].

Figure 7. Scalability using *Spiral* descriptors.

view video stream) produced by systematically sampling 100 distinct document pages written in a Latin alphabet at various camera locations and distances using the system described by Lookingbill et al. [6]. The properties of word bounding boxes in other alphabets have not been assessed. Since performance depends critically on the number of bounding boxes in the query image, we present our results as a function of them.

Fig. 6 compares the performance of zigzag and spiral descriptors and BWC [2]. Spiral descriptors are clearly superior to BWC, probably because of longer feature vectors, and to zigzag descriptors, probably because of the smaller redundancy between descriptors corresponding to different bounding boxes since *Zig-Zag* descriptors corresponding to words below each other two lines apart share a common suffix.

Figure 7 shows the recognition performance as a function of collection size. This demonstrates scalability on collection sizes two orders of magnitude larger than previously tested [8, 2].

6. Conclusions and Future Work

We described a new method for casting image matching as text search and demonstrated its efficiency and scalability in the problem of recognizing documents in real time from low quality blurry images captured with a camera phone.

Our experiments show higher retrieval rates than found in previously published results and scalability to document collections containing a hundred times more pages than those used in earlier studies.

Future research includes other domains of application, and an alternative scheme in which multiple variants of a term are indexed, while at recognition only the canonical term is searched. This trades search time for index size.

We thank our reviewers for their comments.

References

[1] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. In *In ECCV*, pages 404–417, 2006.

[2] B. Erol, J. Graham, E. R. Antúnez, and J. J. Hull. Hotpaper demonstration: multimedia interaction with paper using mobile phones. In *ACM Multimedia*, pages 983–984, 2008.

[3] O. Gospodnetic and E. Hatcher. *Lucene in Action*. Manning Publications Co., Greenwich, CT, USA, 2004.

[4] J. J. Hull, B. Erol, J. Graham, Q. Ke, H. Kishi, J. Moraleda, and D. G. V. Olst. Paper-based augmented reality. In *IEEE ICAT*, pages 205–209, Esbjerg, Denmark, 2007.

[5] G. E. Kopec and P. A. Chou. Document image decoding using Markov source models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(6):602–617, June 1994.

[6] A. Lookingbill, E. R. Antunez, B. Erol, J. J. Hull, Q. Ke, and J. Moraleda. Ground-truthed video generation from symbolic information. In *IEEE - ICME*, pages 1411–1414, 2007.

[7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[8] T. Nakai, K. Kise, and M. Iwamura. Camera based document image retrieval with more time and memory efficient llah. *MIRU 2008*, pages 1252–1259, July 2008.