A two-stage approach to segmentation-free query-by-example word spotting

Leonard Rothacker[a], Marçal Rusiñol[b], Josep Lladós[b], Gernot A. Fink[a]
a: Department of Computer Science, TU Dortmund University, Otto-Hahn-Str. 8, 44221 Dortmund, Germany
b: Computer Vision Center, Dept. Ciències de la Computació, Edifici O, Universitat Autònoma de Barcelona, 08193, Bellaterra, Spain

Abstract

With the ongoing progress in digitization, huge document collections and archives become available to a broad audience. Scanned document images can be transmitted electronically and studied simultaneously throughout the world. While this is very beneficial by itself, it is often impossible to search these document collections automatically. Optical character recognition usually fails when it comes to handwritten or historic documents. In order to address the need of exploring document collections rapidly, researchers are working on word spotting. In query-by-example word spotting scenarios the user selects an exemplary occurrence of the query word in a document image. The word spotting system then retrieves all regions in the collection that are visually similar to the given example of the query word. The best matching regions are presented to the user and no actual transcription is required.

An important property of a word spotting system is the computational speed with which queries can be executed. In our previous work we presented a relatively slow but high precision method. In the present work we will extend this baseline system to an integrated two-stage approach. In a coarse-grained first stage, we will filter document images efficiently in order to identify regions that are likely to contain the query word. In the second fine-grained stage, these regions will be analyzed with our previously presented high precision method. In our evaluation we will report recognition results and query times for the well-known George Washington benchmark. We achieve state-of-the-art recognition results while the query times can be reduced down to 50% in comparison to our baseline.

Introduction

Text recognition in scanned documents usually refers to transcribing images showing text into a machine-based representation like ASCII or UTF-8 [2]. Even though researchers have been investigating this topic for decades, only the recognition of nicely machine-printed texts can be considered as solved. For handwritten documents and historical manuscripts results are far from being satisfactory. Difficulties with handwritten documents are caused by the extreme variability in human writing. Historic documents often show severe degradations, like ink bleed through, bad contrast due to changes in paper color, text line deviations, old fonts and other artifacts caused by old technical standards and storage. Nevertheless, in order to explore these kinds of documents, algorithmic methods for human assistance exist. They usually work in a relatively constrained scenario, but will spare a significant amount of work if they are applicable.

One of the most prominent examples in this regard is word spotting. The task is not to transcribe entire images of text but to automatically detect regions in document images where the query word is likely to be found. The results are presented to the user in form of a ranked list of these regions. In comparison to a full transcription of the document images, word spotting is much more robust against recognition errors. As long as the relevant regions are among the top ranks of the list, the user can finally decide what he actually wants to use from the given results [9]. Errors in a text transcription are not as easy to recognize and even small mistakes might corrupt further processing like a full-text search.

In this scenario the query is an exemplary occurrence of the respective word in the image. Query-by-example word spotting works only as long as the variability in the text is low, as, e.g., in single writer scenarios or documents printed in a single font. Apart from that no prior knowledge of the problem domain is required.

A widely noticed approach to word spotting was presented by Rath and Manmatha [1]. Their method follows a relatively classic pattern recognition pipeline. A document image is first segmented into word regions. After skew and slant normalization, each region is represented by a sequence of feature vectors. Words that are similar to the query are then found by computing distances between feature vector sequences using Dynamic Time Warping. An effect of a prior document segmentation is that subsequent processing steps can be specifically designed for working with word regions. Examples include specialized features, like the upper and lower word contour or the number of ink background transitions along the word image's columns. However, while this may seem advantageous, a severe disadvantage is that such a system will not be able to recover from segmentation errors. It is implicitly assumed that a perfect segmentation is possible. If the assumption fails these errors cannot be handled but all further steps will be based upon them. For instance, if the word segmentation fails, the information encoded in the feature representation will not be useful. Additionally, the processing steps in such a pipeline are only able to work in a locally optimal manner and no information regarding the actual objective, i.e., the recognition, can be taken into account. When segmenting lines or words a recognition step is already incorporated because knowledge about their appearance is assumed. These steps are usually based on heuristics and consequently also the recognition will be based on heuristics. In challenging unconstrained handwriting recognition scenarios it is impossible to rely on knowledge that would justify those assumptions. Therefore, we propose to avoid any early decisions but rather integrate as much information as possible into the final recognition.

The first methods for segmentation-free word spotting were presented in [11] and [12]. Leydier et al. [11] used a keypoint-based approach. Local zones-of-interest form the query word are matched with the most similar local zones in the document image. Regions are retrieved where the spatial configuration of matching zones in the query and the document image fit. In [12] Gatos and Pratikakis use a patch-based framework for segmentation-free word spotting. After preprocessing and normalization they obtain text regions in a filtering step. Within these salient image regions, patches are sampled that are finally matched with the query. By over-segmenting the text regions they are able to analyze all possible word locations.

In our previous works on segmentation-free word spotting [5],[6] we demonstrated how very accurate results can be achieved with a statistical sequence model. The model captures the spatial sequential structure of the query word in a dynamic probabilistic way. Similarity scores were computed between the model and patches that were densely sampled over the document image.

However, a drawback of this method is the high computational effort that is required when processing a complete document image without focusing the recognition on potentially relevant regions as for example in [12]. A word spotting system that is not able to respond instantly is not likely to be accepted by potential users. In order to improve the computational efficiency, we propose a two-stage approach now. In the first stage we identify potentially interesting, i.e., salient regions in a fully segmentation-free manner. In this step we do not incorporate any heuristic decisions but only identify regions in the document image that are visually similar to the query image. We implement this efficiently using an index structure for looking up small image patches in the document image that are similar to small image patches in the word region. This index is independent of a particular query and can be precomputed for any document image. In the second step we investigate these salient regions in more detail. We apply our statistical sequence model for obtaining highly accurate matches. A major advantage of the approach is that both stages are based upon those small image patches that we can find in the query and the document image. No additional preprocessing or feature extraction is required.

Related work

A recent trend in document analysis is to adapt methods from Computer Vision. These algorithms usually have to work in completely unconstrained scenarios dealing with a huge variety of different objects and their appearances in natural scene images. One of the best known approaches that found its way into document analysis are Bag-of-Features. These are statistically estimated feature representations that automatically integrate information from the problem domain, for example document images. In contrast, a heuristically designed feature representation only captures a much more limited number of aspects that have been considered in the manual design process.

The basic idea behind Bag-of-Features representations is to estimate feature representatives that are typical for the problem domain. When applied to images, gradient-based local image descriptors are very popular, for example SIFT descriptors [8]. They encode the local neighborhood at a given point in the image, i.e., a small image patch.

As the typical image descriptors are not directly available, but have to be estimated statistically, three steps are required for computing a Bag-of-Features. First, the typical image descriptors have to be estimated, usually by clustering a large set of descriptors from an image sample set. Second, the descriptors in a given image have to be mapped to the most similar typical image descriptor. Finally, the Bag-of-Features is obtained as the histogram of typical image descriptor frequencies. One of the first and widely recognized methods of Bag-of-Features in Computer Vision was presented in [10]. They demonstrated how objects can be efficiently retrieved from a large number of movie frames. For this purpose objects as well as movie frames are represented by Bag-of-Features representations. The large scale retrieval is possible because features from the query can efficiently be localized in each movie frame with an inverted file structure. This index saves all locations of each feature in a frame.

With respect to word spotting the methods of Shekhar and Jawahar [7], Rusiñol et al. [3] and Almazán et al. [4] are especially relevant to our work. Shekhar and Jawahar [7] use Bag-of-Features representations for retrieving query words from large volumes of segmented word word images. In analogy to [10] they use an inverted file structure for efficient indexing. The index stores pointers for each feature to all segmented word images that

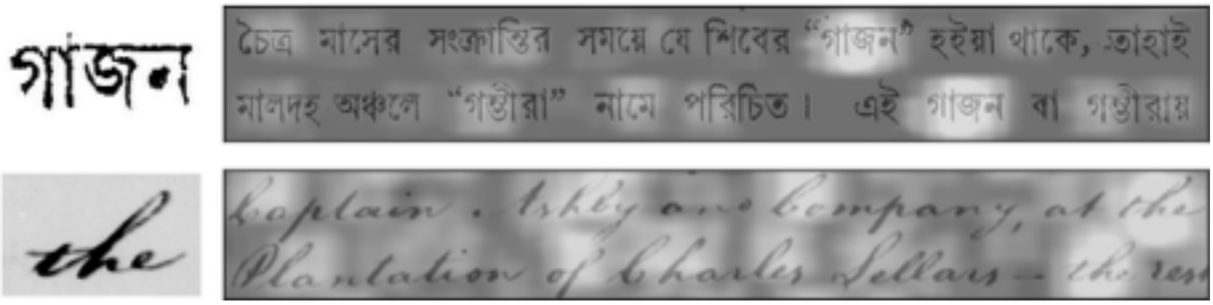

Figure 1: Segmentation-free query-by-example word spotting. Top: Old printed Bangla book. Bottom: Historical George Washington dataset. Queries are shown on the left. On the right the word occurrence probability is indicated by brightness.

contain that specific feature. Given the features occurring in a query word image, the relevant word images in the database can be retrieved almost instantly. In [3] Bag-of-Features have been applied to segmentation-free word spotting for the first time. In [4] it has been shown how recognition accuracy and speed can be improved by using Histogram-of-oriented-Gradients in the same segmentation-free scenario. Both representations have in common that no information about line and word locations is needed. Instead, they are computed uniformly over the document image. While the Bag-of-Features captures the simple occurrences of typical gradient-based local image features, the Histogram-of-oriented-Gradients representation directly captures the image gradients in a grid of cells. For segmentation-free word spotting the query image is transformed according to the respective feature description. This query feature description is then compared to patch feature descriptions that are densely extracted from the document image. This way no prior assumptions about word positions have to be made but all possible locations are taken into account. Such an over-segmentation strategy is computationally very demanding. In [3] the search space is reduced by only considering a single patch size for all queries. In [4] the patch size is adapted to the query size. The huge amount of feature descriptions is efficiently compressed and stored in memory with product quantization. Note, however, that a limited variability of the script's visual appearance is assumed. Only a single patch size is used for spotting a certain query word. For that reason, it cannot be guaranteed that differently scaled instances of the query will be found reliably.

A drawback of the methods presented in [3, 4] is the limited flexibility with respect to modeling spatial gradient configurations. The more detailed the spatial information the more specific becomes the feature representation of a query word image. While for single writer scenarios a relatively explicit representation is advantageous more abstraction will be needed once the variability increases. We, therefore, proposed to integrate the Bag-of-Features with a statistical sequence model, specifically Hidden Markov Models (HMM) [13, 6]. As shown in numerous examples, HMMs are able to model this spatial information in a dynamic probabilistic way [2]. For a segmentation-free application we adapted the patch-based framework presented in [3, 4] but created a sequence of Bag-of-Features representations from the query word image as well as from each patch. After the Bag-of-Features HMM was estimated from the query we obtained a probabilistic similarity score for each patch position with Viterbi decoding. Figure 1 illustrates this for two dataset that we used for evaluating the method. In [5] old printed Bangla documents have been used while in [6] handwritten historic documents written by George Washington and his associates have been considered. The probabilistic score maps show that the variability in the printed document scenario is much smaller than in the handwritten document scenario. But even though the detections in the handwritten word spotting case are less prominent, their scores are still good enough to be considered as most important in the given example. In terms of word spotting accuracy we clearly outperformed the results reported in [3] and [4] on the same benchmark. In terms of computational speed, however, the massive generation of Bag-of-Features sequences from patch representations and their evaluation with the Viterbi algorithm was considerably slower.

In the remainder of this paper we will present a two-stage method for improving the computational efficiency by only applying the Bag-of-Features HMM at document image locations that are salient with respect to the query word. The query specific saliency map is based on an inverted file structure. Locations of typical features in the document image can be efficiently indexed this way. The major difference in computing the saliency map compared to [12] is that we already integrate information from the query instead of just looking for arbitrary text areas. This makes the salient regions specific to the query and we can apply the Bag-of-Features HMM in a more focused manner.

In our experimental evaluation on the George Washington benchmark we will show that speed-ups of more than 50% are possible while the word spotting accuracy is only effected marginally.

A two-stage approach for segmentation-free word spotting

The sole application of the Bag-of-Features HMMs for segmentation-free word spotting [6] is very costly with respect to its computational efficiency. Running such a high precision method for all densely sampled patches on the document image does also not always seem to be appropriate. The majority of the patches is visually very dissimilar to the query and the question raises if these patches cannot be rejected with a more efficient approach.

In this section we will present a method that performs a segmentation-free coarse-grained analysis of the document image followed by a fine-grained application of the Bag-of-Features HMM. The coarse-grained analysis produces a saliency map identifying regions in the document image that are similar to the query. The detailed analysis in the second stage is only applied in the local neighborhood of these salient locations. For the overall segmentation-free property of the method it is very important that both stages rely on the same local image features. This means that if there is no indication of the query word in some region in the saliency map, also the Bag-of-Features HMM will produce low scores in this particular area. The same features that did not match with the query in the first stage will not start matching in the second stage. For that reason we do not incorporate any explicit prior segmentation step that is based on heuristic assumptions like distances between words or lines in the document image.

The two-stage method for segmentation-free word spotting consists of the three processing steps: document image representation, model estimation and model decoding. As the computation of local image features is independent of any particular query and the features are shared among both stages they have to be computed once for each document image. In order to search for a query it has to be modeled with the Bag-of-Features HMM. This incorporates a statistical model estimation procedure. After the query model is available it can be used for retrieving relevant regions in document images. The decoding step consists of the coarse-grained and the fine-grained analysis stages. After efficiently identifying regions in the document image that are similar to the query, these salient areas are analyzed in more detail using the Bag-of-Features HMM. Finally, regions are ranked according to their similarity with the query and presented to the user. The overall process is visualized in Figure 2 for an exemplary document image section of George Washington's letters [15].

1. Document image representation

Documents are represented by typical local image features that are extracted on a dense grid in the image. This is shown at the top of Figure 2. The descriptors that have been considered here (SIFT [8]) consist of histograms of oriented gradients and are intended for capturing the main directions of the pen stroke in the local neighborhood of a grid point. Each of the highly overlapping image features is an abstraction of the document's visual appearance representing mainly the information that is relevant for recognizing handwritten words. This has previously been demonstrated in many applications to document analysis cf. [3, 6, 7, 9, 13]. The "Dense Grid of Descriptors" in Figure 2 exemplarily shows the local neighborhood of a single image feature in the dense grid.

In the next step typical image features are found with a cluster analysis using Lloyd's algorithm [16]. For this purpose a sample set of local image features is required that is representative for the problem domain. In case of a word spotting system all document images are know a priori and no unknown documents need to be considered. If new documents should be added in the future, the cluster analysis can simply be repeated. The basic idea of the cluster analysis is to group similar image features into a given number of clusters. Each cluster has a representative that maximizes the average similarity to all its elements. Note that the representative is usually not among the set of local image features used in the cluster analysis. Finally, all descriptors in the dense grid are assigned to their most similar cluster representative, i.e., the typical image feature. The process in known as quantization. In Figure 2 the "Descriptor Quantization" visualizes the dense grid with points. The points' colors indicate the typical image feature that they have been assigned to. Note how similar color patterns correspond to similar patterns of the pen stroke in the section of the document image. In the following words will be spotted by the occurrence of typical image features in the document image that are also appearing in the query word region.

2. Model estimation

When a user wants to query the word spotting system he has to select an exemplary occurrence of the word in a document image. The respective region also specifies the typical image features that represent the query word. In order to capture the query word's spatial sequential structure, we extract a Bag-of-Features representation for each column within the dense grid. The Bag-of-Features only captures the relative frequency of each typical image feature, not their order. This will be beneficial in the subsequent decoding step. The sequence of Bag-of-Features representations is then modeled with a Bag-of-Features HMM. HMMs are generative finite state machines. At each point in time one state is active and generates an output according to an underlying statistical

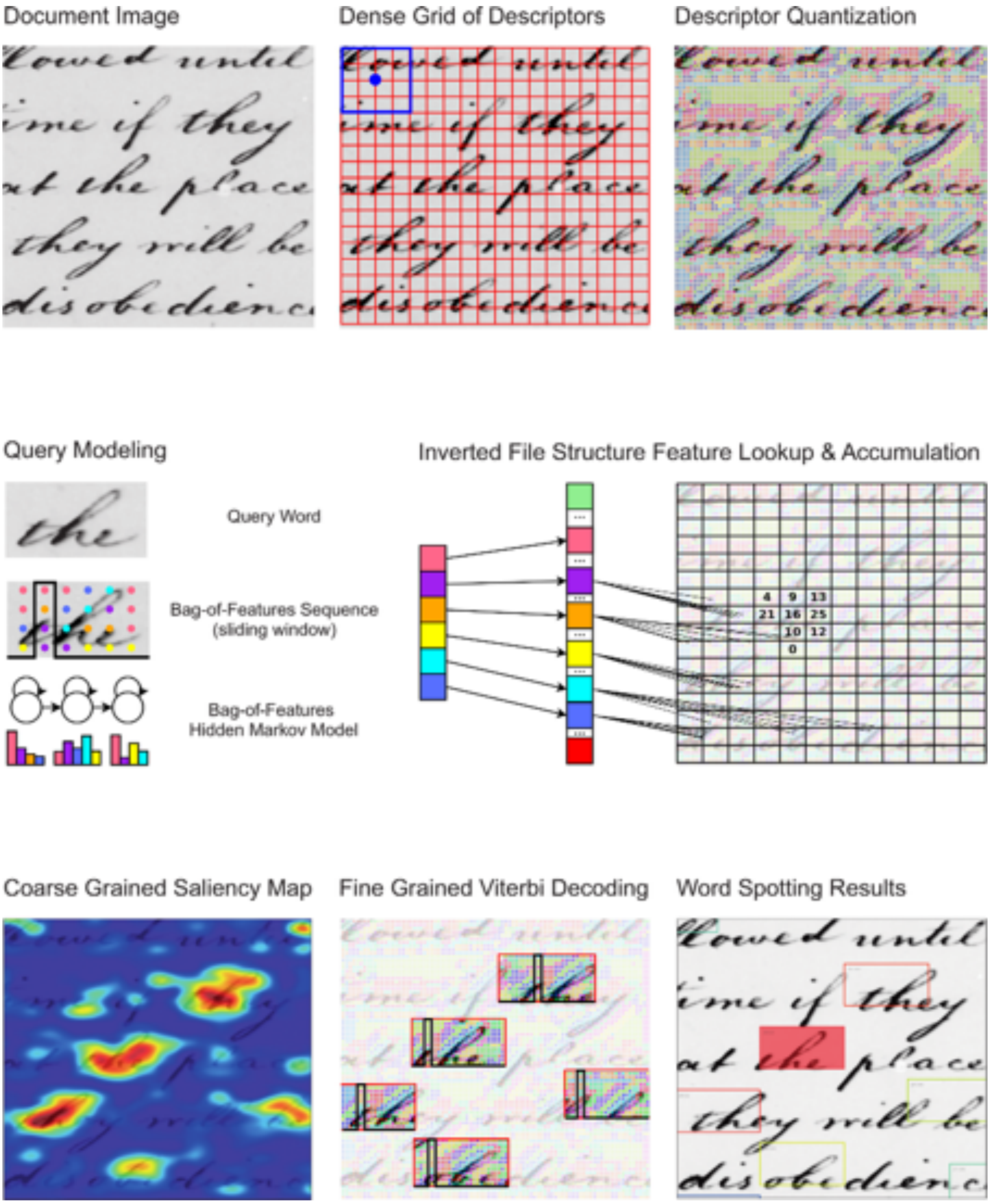
Document Image
Dense Grid of Descriptors
Descriptor Quantization
Query Modeling
Query Word
Bag-of-Features Sequence (sliding window)
Bag-of-Features Hidden Markov Model
Inverted File Structure Feature Lookup & Accumulation
4
9
13
21
16
25
10
12
0
Coarse Grained Saliency Map
Fine Grained Viterbi Decoding
Word Spotting Results

Figure 2: Overview of the two-stage word spotting method. Top: Document image representation. Each document image is represented by typical local image features. In the dense grid these are indicated with different colors. Middle: Query word modeling and inverted file structure lookup & accumulation. Some accumulator cells show counts of how many features from the query model have been detected. Bottom: Coarse-grained saliency map and fine-grained analysis with the Bag-of-Features HMM. Saliency scores are indicated with blue to red colors. For the fine-grained analysis the Bag-of-Features sequence extraction is visualized for all detected regions. The rank of the finally retrieved regions is indicated with blue to red colors. The section of the document image is taken from the George Washington Papers [15].

process. In the estimation procedure, parameters are optimized with respect to the probability of generating this sequence of Bag-of-Features representations from the query word. The process is referred to as Baum-Welch-Algorithm [14]. In the query-by-example scenario it is important that the Bag-of-Features HMM can be estimated with only a single sample [6]. This is usually impossible with continuous HMMs [2]. Figure 2 exemplarily sketches the model estimation for the query word “the”. Below the query word region, the extraction of the Bag-of-Features sequence is visualized. A sliding window, shown in black, is moved over the dense grid in writing direction. At each window position a histogram of typical image feature frequencies is created. For the window position, visualized in Figure 2, the purple feature has a higher frequency than the red and orange features. The Bag-of-Features HMM models feature probabilities for the query word directly within the states. In the given example the states roughly represent the features occurring at the beginning, in the middle and at the end of the query word. For example, the cyan features have a higher probability in the middle while the red features are rather probable at the beginning and the end.

3. Model decoding

After the model has been estimated it can be used for retrieving regions in document images that are similar in terms of typical image feature occurrences. The overall process consists of two stages. In the first stage a coarse-grained analysis is performed. It roughly identifies areas in a document image containing typical image features that also occur in the query word. In the second stage these regions are explored in more detail by measuring their similarity with the Bag-of-Features HMM.

3.1 First stage

The most important component in the first stage is an inverted file structure that we use for localizing typical features in the document image. A similar approach has been used for object detection [10] and retrieving segmented word images [7]. The inverted file structure index contains an entry for each typical image feature that has been determined in the cluster analysis. Each entry indexes all feature locations. The inverted file structure only has to be computed once and feature location lookups become very efficient afterwards. In our scenario we want to lookup features from the query word. Every feature with a non-zero probability in any of the states of the query model will be localized this way. For an actual occurrence of the query word in a document, we expect to find a larger number of detections in roughly the same area. We, therefore, accumulate detections in a cell structure over the document image. For each cell the number of feature detections is counted.

In Figure 2 the inverted file structure lookup and accumulation is visualized for the exemplary query. All features occurring in the query model are localized in the section of the document image. Note that only few features with respect to the total number of features will actually occur in a specific query word region. Also, not all features might be reappearing in a document. In Figure 2 this is indicated for the red feature. For these reasons, inverted file structure lookups are very efficient.

The resulting accumulator matrix can be interpreted as a coarse-grained saliency map. It is coarse-grained because no spatial relations between features have been taken into account, yet. In Figure 2 saliency measures are visualized with blue to red colors. Due to the coarse-grained character, detections are usually not precisely located over the relevant regions.

The final output of the first stage are regions-of-interest that are positioned at the peaks in the saliency map, i.e., a region is created for each locally optimal score. The regions’ sizes equal to the size of the query word. The locally optimal scores are determined in such a way that the regions do not overlap.

3.2 Second stage

The regions-of-interest from the first stage are analyzed in more detail in the second stage. A sequence of Bag-of-Features representations is extracted from each region. This captures the spatial sequential structure in writing direction. As no information about the feature order is encoded in a Bag-of-Features, these representations are robust against smaller vertical displacements of the detected region with respect to the relevant region in the document. The horizontal displacements can be handled by the Bag-of-Features HMM that models the sequential structure in a dynamic probabilistic way. Using the Viterbi algorithm the probability of generating the Bag-of-Features sequence from a detected region can be computed. Each region obtains a new similarity score with respect to the query model. However, even though the Bag-of-Features HMM is relatively robust against displacements of the detected regions, better scores can be obtained when these regions exactly fit the relevant regions in the document. For this reason we dilate the search area locally in order to overcome the coarse detections from the first stage. Finally, we extract detected regions with locally optimal scores and rank them accordingly. Figure 2 visualizes the fine-grained evaluation of the HMM and the retrieved regions that are presented to the user. The regions’ scores are indicated with blue to red colors.

Method	RoI dilation	Overlap threshold	mAP	mR	mQT	Speedup
BoF [3]	—	50%	30.4%	71.1%	340 ms	
HoG [4]	—	50%	54.4%	—	15 ms	
VT [6]	—	50%	69.7%	83.0%	4100 ms	Baseline
IFS	—	50%	50.1%	60.1%	380 ms	10x
IFS+VT	—	50%	60.4%	60.0%	540 ms	7x
IFS+VT	3x3	50%	64.9%	69.2%	980 ms	4x
IFS+VT	5x5	50%	69.6%	80.7%	1830 ms	2x
IFS+VT	7x7	50%	70.0%	82.5%	2750 ms	1.5x
VT [6]	—	25%	71.4%	96.8%	4100 ms	Baseline
IFS	—	25%	48.0%	93.1%	380 ms	10x
IFS+VT	—	25%	53.9%	92.1%	540 ms	7x
IFS+VT	3x3	25%	64.7%	94.7%	990 ms	4x
IFS+VT	5x5	25%	70.9%	96.2%	1830 ms	2x
IFS+VT	7x7	25%	71.6%	96.3%	2760 ms	1.5x

Table 1: Evaluation of the two-stage approach for segmentation-free word spotting

In the following evaluation, we will focus on performance measures in terms of the two stages. We will evaluate their individual and joint accuracy as well as speed. The region-of-interest dilation will be of special interest with respect of the joint evaluation of both stages.

Evaluation

The effect of the two-stage method is evaluated on papers written by George Washington and his associates [15]. The dataset has first been used for evaluating a word spotting system in [1]. In order to measure and compare segmentation-free word spotting performance, Rusiñol et al. [3] defined a benchmark consisting of a subset of 20 pages and 4860 words. For all words ground truth annotations are available, consisting of a bounding box in the document image and a word label. The 20 pages are written in an overall very similar style thus the benchmark can be considered as a single writer scenario. Following this evaluation protocol, cf. [3, 4, 5, 6], we use every word as a query without any further modification, like filtering short words or stemming words. For each query we retrieve a ranked list of regions throughout all 20 pages. By comparing each of those regions with the ground truth annotations, we can decide if a single region is relevant with respect to the query or not. A region is considered as relevant if it overlaps with a bounding box from the ground truth by more than a given threshold and the corresponding word label matches with the query word. The choice of the overlap threshold is critical for the performance measures and we will report results with two different values in the following. In Table 1 we refer to this as “Overlap threshold”.

Given the list of relevant and non-relevant regions, mainly two aspects are important for users of word spotting systems. All relevant regions should be ranked first and the list should contain all relevant words. The first requirement is measured by Average Precision while the second requirement is measured by Recall. An Average Precision of 100% refers to a list where all relevant regions are listed first. A Recall of 100% refers to a list that is complete, i.e., no relevant regions are missing. In order to report results over all queries, we compute means over the individual results, thus mean Average Precision and mean Recall. In Table 1 we refer to these measures as “mAP” and “mR”.

The motivation for the two-stage approach was to improve the computational speed. Mean runtimes for retrieving a single query on a single page are reported as well as the relative speedups with respect to our baseline method presented in [6]. Results have been measured on a Xeon 3.0 GHz. In Table 1 we refer to the mean query time per page as “mQT” and to its relative improvement as “Speedup”.

Our experiments focus on the results obtained with the stages individually and jointly. Additionally, we report results for different overlap thresholds. As the first stage is based on an inverted file structure, it is referred to as “IFS” in Table 1. Analogously, scores in the second stage are computed with the Viterbi algorithm, thus referring to it as “VT”. For the individual application of the first stage (IFS) we directly use the regions-of-interest as output of the word spotting system. For the individual application of the second stage (VT) we use the same approach as presented in [6], i.e., we densely sample patches that are decoded with the Viterbi algorithm for obtaining similarity scores. Please note that the results report in [6] differ due to some parametric optimizations.

In the joint evaluation of both stages, the region-of-interest dilation is important. In Table 1 the different dilation masks can be found in column “RoI dilation”. A dilation mask of 3x3 refers to extending the search area to all neighboring cells in the accumulator matrix from a detected region in the first stage. Larger masks extend the search area further.

Starting with the results obtained for our baseline system (VT) [6] we observe an upper bound for retrieval accuracy. This is expected as we are applying a fine-grained analysis throughout entire document images. When, in contrast, only the coarse-grained analysis in first stage is applied, a significant drop in mean Average Precision can be observed. As the spatial sequential structure of the query word is not modeled in the coarse-grained analysis, the simple occurrence of features from the query words is already a strong indication for a relevant region. Problems occur, for example, when single or multiple characters from the query word appear within other, typically longer words. When putting both stages together, we first of all rerank the results with the Viterbi algorithm without dilating the search area. While the mean Recall stays constant, there is a considerable increase in mean Average Precision. This nicely demonstrates the effect of adding spatial information to the query word modeling. The constant mean Recall is due to the fact that the search space has not been extended in the second stage. The list of relevant regions stays the same.

When increasing the search area, improvements can be observed for both mean Average Precision and mean Recall. Now, the regions-of-interest can be positioned better over the relevant regions in the document image and the Bag-of-Features HMM produces better matches when being evaluated with the Viterbi algorithm. Regions-of-interest that only slightly overlapped with relevant regions in the document are discarded for better matches in the local neighborhood. Additionally, more regions will be regarded as relevant if the detections from the first stage were not precise enough to produce sufficient overlap percentages.

Regarding the overlap threshold a reduction from 50% to 25% shows that over 90% of the relevant words can be roughly located in the first stage. The localizations are just to imprecise to produce over 50% overlaps. This nicely motivates the application the regions-of-interest dilation.

Finally, we measured strong improvements in the mean query time with respect to our baseline system. The sole application of the inverted file structure is more than 10 times faster. Using the two-stage approach, the size of the search area (RoI dilation) is strongly related to the mean query time and recognition accuracy. When matching the accuracy of the baseline system, still a speedup of factor 2 is possible.

When comparing our results to those reported in [3] and [4] we clearly outperform their recognition accuracy with our integrated approach. Low recognition scores in [3] are due to the uniform patch size used for all queries in the retrieval stage. Small words are hard to detect as patch contains a lot of context on the document image besides the relevant word. Results reported in [4] show good recognition accuracy and very fast retrieval times. The Histogram-of-oriented-Gradients (HoG) features nicely encode handwritten words in the single writer scenario. In comparison to our two-stage approach we see two important aspects. Encoding full page HoG representations in memory does not scale for very large collections of document images. In contrast, the inverted file structure in our coarse-grained stage is able to rapidly reduce the search space over all document pages in the collection. This capability has been demonstrated for segmented word images in [7]. Finally, the application of Hidden Markov Models offers more flexibility, especially when more than a single sample of a query is available. Please note that the query times to neither [3] nor [4] are directly comparable as different machines have been used in the evaluation. However, they are suitable for showing the general order of magnitude in which the methods operate.

Conclusion

In this paper we presented a two-stage approach for segmentation-free word spotting on the George Washington benchmark. With respect to our baseline system, considerable improvements in computational speed have been observed. With the sole application of the coarse-grained first stage, speedups of more than 10 times are possible. This comes at the cost of decreased recognition accuracy. When adding the fine-grained word spotting stage, the benefits of integrating a statistical sequence model can be demonstrated. The recognition accuracy increases substantially. Computational speed can still be reduced to 50% without loosing any precision. In comparison to [3] and [4], more accurate results can be achieved at the cost of higher query execution times.

Furthermore, we present the effect of different overlap thresholds for the evaluation. We can show that the fine-grained stage is very well able to handle regions-of-interest having smaller overlaps with the ground truth. For larger dilation masks, very high mean Recall scores can be achieved without any loss in mean Average Precision. We doubt that an overlap threshold of 25% or 50% will make any difference for users of word spotting systems as detected regions will usually be presented in a larger context in the document image.

In our future research we will work on better region localizations in the coarse-grained stage. This might be achieved by taking some limited amount of the query word's spatial structure into account.

Acknowledgments

This work is partially supported by the German Academic Exchange Service on the basis of a *DAAD-Doktorandenstipendium.*

References

[1] T. Rath, R. Manmatha, Word spotting for historical documents. International Journal on Document Analysis and Recognition, vol. 9, no. 2–4, pages 139–152, 2007.

[2] T. Plötz, G. A. Fink, Markov Models for Handwriting Recognition. SpringerBriefs in Computer Science, Springer, 2011.

[3] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, Browsing heterogeneous document collections by a segmentation-free word spotting method. Proceedings of the International Conference on Document Analysis and Recognition, pages 63–67, 2011

[4] J. Almazán, A. Gordo, A. Fornés, E. Valveny, Efficient exemplar word spotting. Proceedings of the British Machine Vision Conference, pages 67.1–67.11, 2012.

[5] L. Rothacker, G. A. Fink, P. Banerjee, U. Bhattacharya and B.B. Chaudhuri, Bag-of-Features HMMs for Segementation-Free Bangla Word Spotting. International Workshop on Multilingual OCR, 2013.

[6] L. Rothacker, M. Rusiñol, G. A. Fink, Bag-of-Features HMMs for Segmentation-Free Word Spotting in Handwritten Documents. Proceedings of the International Conference on Document Analysis and Recognition, 2013.

[7] R. Shekhar and C. Jawahar, Word image retrieval using bag of visual words. International Workshop on Document Analysis Systems, pages 297–301, 2012.

[8] D. Lowe, Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, vol. 60, no. 2, pages 91-110, 2004.

[9] J. Lladós, M. Rusiñol, A. Fornés, D. F. Mota, and A. Dutta, On the influence of word representations for handwritten word spotting in historical documents. International Journal of Pattern Recognition and Artificial Intelligence, vol. 26, no. 5, 2012.

[10] J. Sivic and A. Zisserman, Video Google: A text retrieval approach to object matching in videos. International Conference on Computer Vision, vol. 2, 2003.

[11] Y. Leydier, A. Ouji, F. LeBourgeois, and H. Emptoz, Towards an omnilingual word retrieval system for ancient manuscripts. Pattern Recognition, vol. 42, no. 9, pages 2089– 2105, 2009.

[12] B. Gatos and I. Pratikakis, Segmentation-free word spotting in historical printed documents. Proceedings of the International Conference on Document Analysis and Recognition, 2009, pages 271–275.

[13] L. Rothacker, S. Vajda, G. A. Fink, Bag-of-Features Representations for Offline Handwriting Recognition Applied to Arabic Script. Proceedings of the International Conference on Frontiers in Handwriting Recognition, 2012.

[14] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," The Annals of Mathematical Statistics, vol. 41, pages 164–171, 1970.

[15] George Washington Papers at the Library of Congress, 1741–1799, http://memory.loc.gov/ammem/gwhtml/gwhome.html. Manuscript Division, Library of Congress, Washington, D.C.

[16] S. P. Lloyd, Least squares quantization in PCM, IEEE Transactions on Information Theory, vol. 28, no. 2, pages 129–137, 1982.