

# Efficient Word Retrieval by Means of SOM Clustering and PCA

Simone Marinai, Stefano Faini, Emanuele Marino, and Giovanni Soda

Dipartimento di Sistemi e Informatica - Università di Firenze,  
Via S.Marta, 3 - 50139 Firenze - Italy  
marinai@dsi.unifi.it

**Abstract.** We propose an approach for efficient word retrieval from printed documents belonging to Digital Libraries. The approach combines word image clustering (based on Self Organizing Maps, SOM) with Principal Component Analysis. The combination of these methods allows us to efficiently retrieve the matching words from large documents collections without the need for a direct comparison of the query word with each indexed word.

## 1 Introduction

Nowadays, Digital Libraries and archives store large collections of documents in image format. For instance, the *Gallica* Web site maintained by the National Library of France holds more than 70,000 works (mainly books) stored as images, however text-accessible works are limited to a few thousands. For the works stored as images it is frequently possible to access the contents by browsing the table of contents, but it is more complex to perform word searches in the free text. As mentioned in [1] the use of Document Image Retrieval (DIR) techniques is essential to build successful Digital Libraries. Document Image Retrieval aims at finding relevant documents from a corpus of digitized pages relying on image features only. Important sub-tasks include the retrieval of documents on the basis of layout similarity and on the basis of the textual content [2, 3].

In this paper we focus on one specific sub-topic of text-based DIR: word retrieval, that addresses the efficient identification of the occurrences of a given word in the indexed documents. Word retrieval is tightly related to “keyword spotting” where the interest is to locate user defined words from an information flow (e.g. audio streams or sequences of digitized pages) [4–6]. In word indexing the emphasis is not only on the recognition, but also on the efficient indexing and retrieval of words. Several methods have been recently proposed for the effective retrieval of text from both printed and handwritten documents (e.g. [7–9]). Regardless of the word representation and matching strategy adopted all these methods are designed to work with a relatively small collection of documents and the scalability to larger data-sets is an issue. The efficient retrieval of words from large document repositories is the topic of the work described in this paper. Each word is represented by one simple feature vector that is based on word image zoning. The zoning is a particular kind of word representation used for holistic word recognition that consists of overlapping the word image with a fixed-size grid and

computing some features (e.g. the density of black pixels) in each grid region. Even if more appropriate representations have been proposed, the zoning works reasonably well for printed text with uniform font and allows us to cast the word retrieval as a problem of search in high dimensional vector spaces. As a matter of fact the method described in this paper can work with every word representation that encodes the word into a high dimensional feature vector. Efficient search in high dimensional vector spaces is still the subject of active research. When dealing with low dimensional spaces, then the R-tree [10] (and its variants) can be adopted to reduce the search cost from the linear one. Some methods (e.g. X-Tree [11] or cluster tree [12]) have been proposed to search in high dimensional spaces, however these methods degenerate to the linear complexity when dealing with spaces having more than a few dozens of dimensions.

In this paper we address the word indexing by exploring the effectiveness of a SOM-based word image clustering where the words are grouped considering the image similarity. The idea of using this clustering technique is extended from a previous work that exploited the SOM for character-like object clustering [8]. To reduce the search complexity we project the data in each cluster with Principal Component Analysis (PCA) and then perform an efficient search in the projected space with an appropriate search algorithm (e.g. the X-tree). In the final retrieval step we refine the sorting of the top ranked words by computing the similarity in the original space.

The paper is organized as follows: in Section 2 we describe the use of SOM for word image clustering. In Sections 3 and 4 we analyze the word indexing and retrieval, respectively. Experimental results are reported in Section 5 and some conclusions are drawn in Section 6.

## 2 Self Organizing Maps of Word Images

The Self Organizing Map (SOM [13]) is a special kind of artificial neural network that is based on competitive learning, where the output neurons of the network compete among themselves. The purpose of SOM training is the computation of an optimal clustering of a collection of patterns in  $\mathbb{R}^n$  (representing words in our case). In the Self Organizing Map the neurons are typically arranged in a two dimensional lattice: the feature map. Each neuron receives inputs from the input layer (vectors in  $\mathbb{R}^n$ ) and from the other neurons in the map. During the learning the network performs clustering and the neurons are moved in the lattice so as to reflect cluster similarity by means of distances in the map. To each element in the SOM map it is associated one real vector (in  $\mathbb{R}^n$ ) that can be considered as a prototype of the patterns in the cluster.

One advantage of the use of SOM for word clustering, with respect to other clustering algorithms like k-means, is the spatial organization of the feature map that is achieved after the learning process. Basically, more similar clusters are closer than more different ones. Consequently, the distance among prototypes in the output layer of the SOM can be considered as a measure of similarity between words in the clusters.

In our model the SOM is used to build a word image database where the clusters contain similar words (from a graphical point of view). For the SOM training we modified the `SOM_PAK` package<sup>1</sup> that implements the standard incremental learning algorithm

<sup>1</sup> The package can be found at: [http://www.cis.hut.fi/research/som\\_lvq\\_pak](http://www.cis.hut.fi/research/som_lvq_pak)



**Fig. 1.** Eight prototypes obtained from a SOM map trained with the standard training algorithm. Apart from very common words (e.g. "une") the prototypes do not correspond to actual words.



**Fig. 2.** Some prototypes obtained from one map trained with the modified training algorithm (prototypes correspond to actual words)

([13] page 109). In this algorithm, after each training step the prototype of each neuron is "moved" in the  $\mathcal{R}^n$  space so as to better represent the words belonging to its cluster and to its neighborhoods. This is obtained by replacing the prototype with the arithmetic mean of the patterns belonging to the clusters in the neighborhood.

To evaluate the suitability of the incremental learning algorithm we made some preliminary tests analyzing some maps computed with this algorithm. After sorting the words in each cluster on the basis of their distance from the cluster prototype, we noticed that several occurrences of a given word were spread in the rank. The reason is that different words are put approximately at the same distance with respect to the query, even if their mutual distance is high. This effect is due to the position of the prototypes that usually does not correspond to "actual words" unless the corresponding words are very homogeneous as in the case of stopwords (e.g. see the word 'une' in Figure 1).

To solve the above mentioned problem we modified the incremental learning algorithm. Basically, we update each prototype with the closest training vector among all the patterns in the neighborhood (instead of computing the arithmetic mean). This prototype update is made after each training epoch<sup>2</sup> and also at the end of the training process. From an algorithmic point of view we first scan the training set and associate each input pattern to the closest prototype. Subsequently, we replace the prototype with the closest associated pattern.

This solution provides good results since the final map is uniform and the prototypes usually represent the most frequent words in the dataset. The prototypes shown in Figure 2 are obtained with the modified training algorithm. To suggest the general structure of one trained SOM we show in Figure 3 the contents of two neurons. We can remark that in general the farthest words are loosely related with those closer to the prototype.

<sup>2</sup> The epoch is a cycle of presentation of all the patterns to the network.

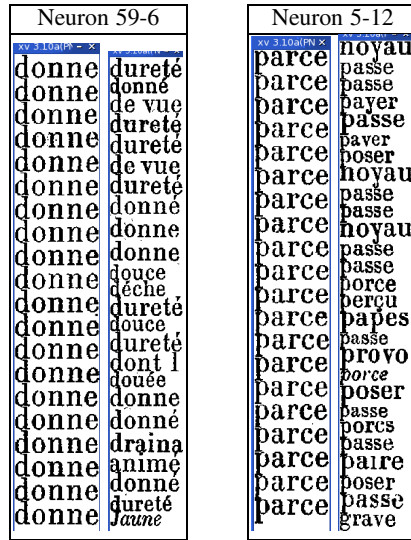


Fig. 3. Examples of the contents of two neurons. For each neuron we show the words closest to the prototype (on the left) and the farthest words (on the right).

### 3 Word Indexing

The word indexing is composed by several steps that are sketched in Figure 4 and described in this section.

During the indexing the pages are analyzed by means of one layout analysis tool that identifies the text regions and extracts the words by means of an RLSA-based algorithm. The indexed words are then split into six disjoint index partitions on the basis of their aspect-ratio (the ratio between the word height and width). In doing so the words in each partition have a similar aspect-ratio and the clustering is performed on uniform data.

For each indexed word its aspect-ratio is computed and considered to find the right partition. Next, the word image is linearly scaled to appropriate dimensions, obtaining a vectorial representation whose items contain the average gray level of the pixels belonging to the corresponding grid cell. The main problem of this approach is the high dimensionality of the feature vector (hundreds of dimensions) that is reflected into a long training time for the SOM. However, it should be remarked that this size is a problem only for the training performed during the indexing, but it is not important for the word retrieval.

To reduce the cost of the search in each cluster we also compute, during the indexing, the projection that best represents the data with PCA. The following procedure is repeated for each SOM cluster in order to compute a low dimensional hyperplane by means of Principal Component Analysis (e.g.[14], pag 568). We first compute the mean vector  $\mu$  and the  $n \times n$  covariance matrix  $\Sigma$  of the data in the cluster. The eigenvectors and eigenvalues of  $\Sigma$  are computed and the eigenvectors are sorted according to decreasing eigenvalue. The first  $h$  eigenvectors ( $e_1, e_2, \dots, e_h$ ) are combined as columns of the  $n \times h$  matrix  $A$ . It is now possible to project the data in the cluster (for instance one point  $x$ ) onto the  $h$ -th dimensional subspace according to

## Indexing

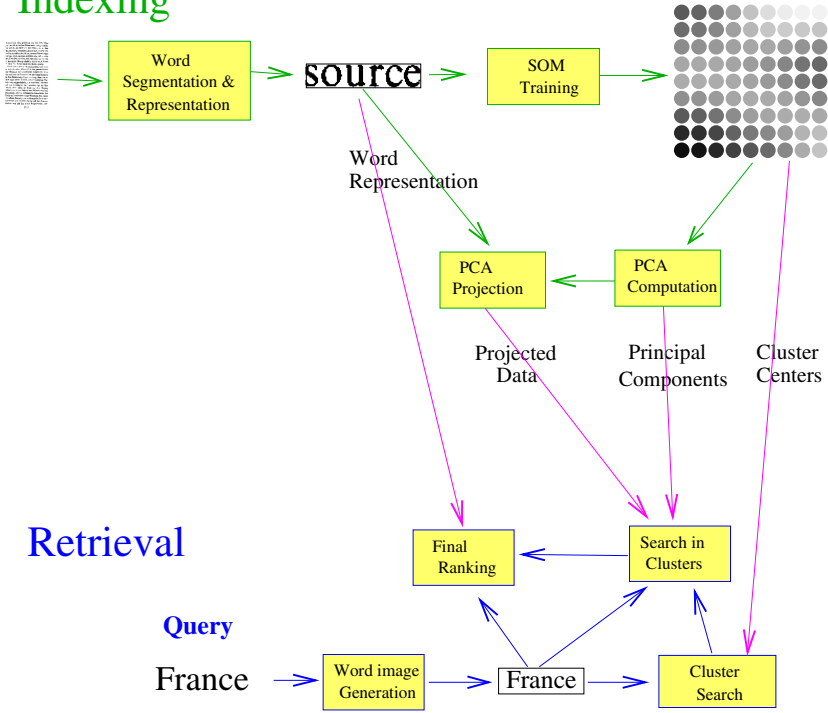


Fig. 4. Main steps in the Word Indexing and Retrieval

$$x' = A^t(x - \mu) \tag{1}$$

Summarizing, the word indexing is composed by the following steps: 1) word segmentation 2) SOM training and PCA computation from a sub-set of the words to be indexed 3) projection of all the indexed words in the lower dimensional space.

## 4 Word Retrieval

Without taking into account efficiency issues one simple approach for word retrieval would rely on the linear comparison of each indexed word with the query word image. Unfortunately, this approach is not feasible when large word databases are considered. The proposed method takes into account the main features of SOM clustering and PCA to efficiently address the word retrieval problem. Let us describe the four main steps of the word retrieval algorithm.

### 1. The Query

From the user point of view the queries are made with a simple text-based interface. Starting from the ASCII text one word image is obtained with  $\text{\LaTeX}$  by using the *Times* font. This word image is then encoded similarly to the indexed

words: we identify the partitions to be considered, and we scale the image according to the average dimensions of each partition thus obtaining a vectorial word representation.

## 2. Cluster Search

In the second step we identify the three clusters having prototypes closer to the query in order to restrict the subsequent search. Since we compare the indexed words with one  $\text{\LaTeX}$  generated query (and due to the presence of noise in indexed words) it is not usual that all the most similar words are contained in the closest cluster. To tackle this problem we consider the three closest clusters (the number of clusters to retain has been obtained after some preliminary tests). In so doing we reduce on the average the number of patterns to be processed by the subsequent step by a factor 500 (for a map having size of 50 x 30).

To illustrate the need for the multiple cluster searches we show in Figure 5 the first 31 words in the three clusters closest to the image generated for the query *alcool*. In this case the first cluster contains only a few instances of the word *alcool*, the second cluster does not contain any occurrence of the word, whereas the last one is the most populated.

chimi	chaud	alcool
chimi	chaud	chinoi
chimi	chand	cedent
alcool	chaud	aident
chimi	chaud	cedant
chimi	chaud	cedent
chimi	chaud	alcool
cedent	chaud	alcool
chassi	chand	alcool
aillent	chaud	alcool
cerami	chaud	alcool
chimi	chaud	alcool
alcool	chaud	alcool
alcool	chaud	alcool
nissent	chaud	sistent
cedant	chaud	alcool
chimi	chaud	alcool
alcool	chaud	alcool
chimi	chaud	alcool
alcool	chaud	alcool
chimi	chaud	alcool
chimi	chaud	alcool
alcool	chaud	alcool
cedant	chaud	alcool
alcool	chaud	alcool
chlori	chaud	alcool
nissent	chaud	cedant
abord	chaud	chloré
abord	chaud	cedent
nissent	chaud	alcool
sedant	chaud	alcool

**Fig. 5.** Left to right: the first words in the three clusters closest to the prototype generated for the query *alcool*

Neuron (24,18)		Neuron (25,19)		
$R^{10}$	$R^{684}$	$R^{10}$	$R^{684}$	
importa	<b>fromage</b>	fromage	fromage	
Alors, en	homme	fromage	fromage	
in formé	tourner	le temps	lessivage	
betterave	nommé	<i>losange</i>	lessivage	
tourner	Comme	immerge	fourrage	
<b>former</b>	nommé	harengs	brassage	
laisser	in formé	<b>Ouvrage</b>	lessivage	
tances;	importa	fourrage	lessivage	
(insecte	bouche	lessivage	lessivage	
<b>fromage</b>	minute;	Le <i>confit</i>	lessivage	
Comme	<b>homme</b>	fourrage	lessivage	
lessives	heures,	<i>battage</i>	lessivage	
retrouve	fournie	<i>délavage</i>	lessivage	
trumeau	laisser	fourrage	fourrage	
histoire	betterave	lessivage	lessivage	
nommé	retrouve	lustrage	harengs	
laissent	luzerne	<i>lustrage</i>	brossage	
betterave	<b>former</b>	longtemps	lessivage	
minute;	(travaux	l'ouvrage	brassage	
<b>homme</b>	trumeau	<i>éclairage</i>	lessivage	

Fig. 6. Ranking of the 20 words closest to the query *fromage*. We show two clusters and the rankings in the original space ( $R^{684}$ ) and in the projected one ( $R^{10}$ ). On the right we report the final ranking obtained by merging three lists (one list is not shown on the left).

3. **Search in Clusters** After identifying the three closest clusters we identify the most similar words by means of PCA. During the search in the three clusters we look for the  $k$  nearest neighborhoods in the projected spaces. In the left part of Figure 6 we compare the rankings obtained in the projected space and in the original one for the words belonging to two clusters when looking for the word *fromage*. The cluster (24,18) contains one occurrence of the word which is in the first position in the  $\mathbb{R}^{684}$  space, but is in the 10th position in  $\mathbb{R}^{10}$ . Two correct words in the other cluster are in the first positions in both spaces. From this example it is clear that the PCA projection approximates the proximity in the  $n$  dimensional space thus requiring a refinement in the next step.

4. **Final Ranking**

The previous step allows us to identify the most similar words in the projected spaces of the three closest clusters. To merge the three lists of top-ranked words and refine the final ranking we compute the distance in the original space between the query word and the words in the three lists. It is worth to remark that the computa-

tion in the original space is performed for a small number of words and therefore it is not problematic from the computational point of view. In the right part of Figure 6 we summarize the final ranking obtained by merging the three lists for the query *fromage* (to simplify the figure only two lists are shown in the left part of the figure).

#### 4.1 Complexity Analysis

To analyze the complexity of the system we compare the computational cost with the costs of two simpler approaches. For each method we consider the search for the words in a given index partition (identified by the query word aspect-ratio) that are more similar to the query word. Both the indexed words and the query one are represented by vectors in  $\mathfrak{R}^n$  where  $n$  depends on the partition; usually the vector contains hundreds of items (e.g.  $n = 684$  for partition number 4). For each method we consider the complexity required for computing the distances, but we do not take into account the computation required to sort the words on the basis of the above mentioned distances (this cost is constant for the three methods).

##### – Sequential scan

The simplest approach is based on the sequential comparison of each indexed word with the query by considering the original feature vector. Let  $P$  be the number of words belonging to the partition, the complexity of retrieval for this approach is therefore:

$$C_r = O(n \cdot P) \quad (2)$$

that is obtained by comparing  $P$  vectors with  $n$  dimensions.

##### – Use of SOM clustering

In the second approach we use the method proposed in this paper without the use of the PCA projection. In this case during the indexing it is required to compute the SOM clustering by considering a sub-set of the words to be indexed. Once the optimal SOM is computed we need to identify, for each word to be indexed, the cluster it belongs to. We therefore have an indexing cost:

$$C_i = SOM_t(P') + O(n \cdot P \cdot S), \quad (3)$$

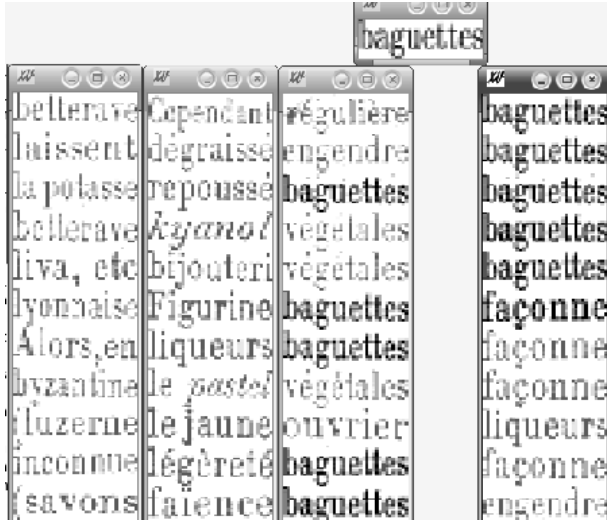
where  $SOM_t(P')$  denotes the cost of training the SOM map with the set of words  $P'$  (usually  $P' \subset P$ ), and  $S$  is the size of the map (in the current experiments the size is 50x30).

Concerning the retrieval cost we should consider two factors: the cost for selecting the appropriate cluster and the cost for computing the distance of the query from all the items in the selected clusters. Let  $J$  be the average number of words in each cluster (on the average we have  $J = \frac{P}{S}$ ). The maximum complexity obtained when evaluating three clusters is:

$$C_r = O(S \cdot n + 3 \cdot n \cdot J) \quad (4)$$

##### – SOM and PCA

The last approach that we consider is the overall method described in this paper including the PCA projection. Let  $k$  ( $k \ll n$ ) be the size of the lower dimensional space (in our experiments  $k = 10$ ). In this case the indexing cost is:



**Fig. 7.** Screenshot of the word retrieval with the query *baguettes*. On the top we report the L<sup>A</sup>T<sub>E</sub>X prototype, on the left the rankings in the three selected clusters and the final ranking is shown on the right.

$$C_i = SOM_t(P') + PCA_t(P') + O(k \cdot n \cdot P \cdot S), \quad (5)$$

where  $PCA_t(P')$  denotes the cost for computing the PCA eigenvalues and eigenvectors.

The retrieval cost is composed by three parts: the search for the best clusters, the search in the clusters by considering the  $\mathcal{R}^k$  vectors and the final refinement of the rank:

$$C_r = O(S \cdot n + 3 \cdot k \cdot J + 60 \cdot n) \quad (6)$$

Figure 7 reports one additional example of the results achieved with the whole word retrieval approach.

## 5 Experiments

The experiments described in this paper are made on two books (containing 1280 pages) that are part of an encyclopedia of the XIX<sup>th</sup> Century<sup>3</sup>. To evaluate the word retrieval we built a pseudo ground-truth by running one commercial OCR engine on the two books. The OCR engine has high recognition performance on this data-set and can be considered as a reasonable approximation of a human validated ground-truth.

The words to be used as queries have been selected from each partition with a particular emphasis on longer words. Basically, we had two types of query words: rare

<sup>3</sup> *Les Merveilles De l'Industrie*: downloaded from the web site of the *National Library of France* (<http://gallica.bnf.fr>).

**Table 1.** Performance of word retrieval for some frequent (left) and rare (right) words. Frequent words have more than 20 occurrences in the data-set.

Word	<i>Rec</i>	<i>Cor</i>
provient	17	15
femme	19	19
raison	17	9
porter	19	17
terrain	17	17
baguettes	13	11
Bontemps	13	13
Egypte	13	8
chaud	15	1
outil	15	0
vive	11	10
vent	11	1
large	4	4
moulage	18	16
graines	13	8
lignes	11	3
abondance	14	14
demande	11	8
enfants	16	15
incolore	19	15
bambou	10	10
explication	10	10
proportion	17	17
bronze	16	15
Deux	19	15

Word	<i>Rec</i>	<i>Cor</i>	<i>OCR</i>
Savon	10	6	14
intime	11	9	12
drainage	8	8	12
tournesol	1	1	11
grillage	7	7	10
assemblage	7	6	10
horizon	4	3	10
Marie	2	0	10
lande	5	5	8
herbes	4	0	8
Elton	5	4	7
abattre	2	2	6
condiment	2	0	6
japonaise	0	0	6
fromage	3	3	5
mandarin	0	0	5
dosages	4	4	4
barbare, chirurgie	1	1	4
Danemark, kilos	1	1	3
Suger	2	2	2
fumage, vasion	1	1	2
Lorenzo, idiot	1	1	2
Buch	2	0	2
lagunes, Elliot	1	1	1
geoisises, argentier	0	0	1

**Table 2.** Comparison of the performance of the word retrieval with the sequential scan. The meaning of *Rec* and *Top* is the same of Table 1.

Query	Proposed method	Sequential scan
	<i>Rec/Top</i>	<i>Rec/Top</i>
Canada	>20 / 20	14 / 8
graines	13 / 8	20 / 18
alcool	>20 / 20	>20 / 20
baguettes	13 / 11	9 / 3
Savon	10 / 6	9 / 2
raison	17 / 9	15 / 8

words (a few occurrences among the 1280 pages) and frequent words (occurring hundreds of times in the data-set). Table 1 summarizes the results obtained for some of the test words. For each query we analyzed the list of the 20 top-ranked words and computed two values: *Rec* is the number of correct words found in the list; *Top* denotes the number of subsequent correct words reported at the top of the list. *OCR* denotes the

number of words found by the OCR engine. We considered also the following words: *Canada, alcool, peaux, violet, France, Louis, nombre, cylindre, verre, chaque, lorsqu, aniline, volume, soluble, lessive, culture, sulfure, production*. For the latter list of words all the first 20 answers were correct ones.

### 5.1 Comparison with the Sequential Scan

As discussed in Section 4.1 the proposed approach has a better behaviour with respect to the naive sequential comparison in terms of computational complexity. However, the reduced complexity is useless if we obtain a worst effectiveness. In other terms, the risk is that by using the PCA we loose too much information and some relevant words are not identified by the third step of the retrieval (and therefore cannot be considered in the final ranking). To analyze this aspect we made some preliminary tests comparing the number of correctly retrieved words for the sequential scan and for the proposed approach (see Table 2). We considered six words: in one case (word *alcool*) the two methods provide the same result, whereas in one case (word *graines*) the sequential scan provides better results. However, for the remaining four words the proposed method is not only more efficient, but also more effective, retrieving more correct words.

## 6 Conclusions

We described a general system for performing word image retrieval by means of a SOM-based word image clustering combined with PCA. The proposed approach addresses the efficiency issues related to the proximity search of large quantities of high dimensional vectors. From the preliminary tests reported in this paper we conclude that the efficiency gain is not obtained at the cost of a reduced retrieval effectiveness.

One restriction of SOM clustering is the need to compute a new set of clusters when dealing with different documents. However, the adaptation to different languages and fonts of the clustering is automatic and does not require additional interaction with the user. Some aspects will require additional investigations, namely the introduction of more appropriate word image distances as well as the use of efficient algorithms to search in the projected spaces (e.g. the X-tree algorithm).

## References

1. H. S. Baird, "Digital libraries and document image analysis," in *Proc. 7th ICDAR*, pp. 2–14, 2003.
2. D. Doermann, "The indexing and retrieval of document images: A survey," *Computer Vision and Image Understanding*, vol. 70, pp. 287–298, June 1998.
3. M. Mitra and B. Chaudhuri, "Information retrieval from documents: A survey," *Information Retrieval*, vol. 2, no. 2/3, pp. 141–163, 2000.
4. J. D. Curtis and E. Chen, "Keyword spotting via word shape recognition," in *Proceedings of the SPIE - Document Recognition II*, pp. 270–277, 1995.
5. J. Trenkle and R. Vogt, "Word recognition for information retrieval in the image domain," in *SDAIR*, pp. 105–122, 1993.

6. W. Williams, E. Zalubas, and A. Hero, "Word spotting in bitmapped fax documents," *Information Retrieval*, vol. 2, no. 2/3, pp. 207–226, 2000.
7. K. Kise, M. Tsujino, and K. Matsumoto, "Spotting where to read on pages - retrieval of relevant parts from page images," in *Document Analysis Systems V*, pp. 388–399, Springer Verlag- LNCS 2423, 2002.
8. S. Marinai, E. Marino, and G. Soda, "Indexing and retrieval of words in old documents," in *Proc. 7th ICDAR*, pp. 223–227, 2003.
9. C. L. Tan, W. Huang, Z. Yu, and Y. Xu, "Imaged document text retrieval without OCR," *IEEE Transactions on PAMI*, vol. 24, pp. 838–844, June 2002.
10. A. Guttman, "R-tree: a dynamic index structure for spatial searching," in *Proc. ACM SIGMOD*, pp. 47–57, 1984.
11. S. Berchtold, D. A. Keim, and H.-P. Kriegel, "The X-tree: an index structure for high-dimensional data," in *Proc. 22nd VLDB*, pp. 28–39, 1996.
12. D. Yu and A. Zhang, "Clustertree: integration of cluster representation and nearest-neighbor search for large data sets with high dimensions," *IEEE Transactions on Knowledge and Data Discovery*, vol. 15, no. 5, pp. 1316–1337, 2003.
13. T. Kohonen, *Self-organizing maps*. Springer Series in Information Sciences, 2001.
14. R. O. Duda, P. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & sons, 2001.