2014 22nd International Conference on Pattern Recognition

An Application-Independent and Segmentation-Free Approach for Spotting Queries in Document Images

Houssem Chatbri*, Paul Kwan†, and Keisuke Kameyama‡
*Graduate School of Systems and Information Engineering, Department of Computer Science, University of Tsukuba, Japan
†School of Science and Technology, University of New England, Armidale NSW 2351, Australia
‡Faculty of Engineering, Information and Systems, University of Tsukuba, Japan
*chatbri@adapt.cs.tsukuba.ac.jp, †paul.kwan@une.edu.au, ‡kame@adapt.cs.tsukuba.ac.jp

***Abstract*—We report our ongoing research on an application-independent and segmentation-free approach for spotting queries in document images. Built on our earlier work reported in [1][2], this paper introduces an image processing approach that finds occurrences of a query, which is a multi-part object, in a document image, through 5 steps: (1) Preprocessing for image normalization and connected components extraction. (2) Feature Extraction from connected components. (3) Matching of the query and document image connected components' feature vectors. (4) Voting for determining candidate occurrences in the document image that are similar to the query. (5) Candidate Filtering for detecting relevant occurrences and filtering out irrelevant patterns. Compared to existing methods, our contributions are twofold: Our approach is designed to deal with any type of queries, without restriction to a particular class such as words or mathematical expressions. Second, it does not apply a domain-specific segmentation to extract regions of interest from the document image, such as text paragraphs or mathematical calculations. Instead, it considers all the image information. Experimental evaluation using scanned journal images show promising performances and possibility of further improvement.**

I. INTRODUCTION

Storing documents in digital libraries has become a frequent task nowadays due to the availability of large storage media [3]. In addition to insuring document preservation and affordable online access, digital libraries provide valuable data for document analysis, which paved the way to numerous applications on writer identification [4], automatic document classification [5], historical document analysis [6], word and mathematical expression spotting [7][8], etc.

We focus in this work on the area of query spotting, that is finding occurrences of a query in a document image. This area of research has gained attention as early as digital libraries started to become popular [9]. In case of printed documents with standard fonts and high resolutions, spotting can be implemented using Optical Character Recognition (OCR) by *recognize-then-retrieve* approaches [10][11]. However, when documents are old, handwritten, or multi-lingual, more sophisticated and *recognition-free* approaches become needed [12][13]. Such methods usually extract features from the query and match them in the document image [7].

Approaches for spotting are often restrictive to deal with particular classes of queries such as words [7] or mathematical expressions [8]. In addition, since the query class is already known, regions of interest are usually extracted to facilitate the spotting process. In this work, we introduce an application-independent approach by supposing that a query is any type of multi-part objects, which includes words, mathematical expressions, diagrams, etc. Furthermore, we consider all the document image information and do not apply segmentation. To the best of our knowledge, application-independence and the segmentation-free aspect make the proposed approach novel compared to the state-of-the-art.

The proposed approach is modular and finds occurrences of a query in a document image through 5 steps: (1) Preprocessing for image normalization and connected components extraction. (2) Feature Extraction from connected components. (3) Matching of the query and document image connected components' feature vectors. (4) Voting for determining candidate occurrences in the document image that are similar to the query. (5) Candidate Filtering for detecting relevant occurrences and filtering out irrelevant patterns.

Since our approach is modular, each of its steps can be further improved and tuned for a particular application. In this work, we report an application independent implementation of the approach.

The remainder of this paper is organized as follows: Sec. II overviews the state-of-the-art of spotting in document images. The proposed approach is explained in detail in Sec. III. Experimental results are presented and discussed in Sec. IV. Sec. V announces our conclusions and future directions.

II. RELATED WORK

As stated above, methods for spotting can be categorized as OCR-based or recognition-free. In this section, we review references of recognition-free methods, as our contribution is of this category. We refer the reader to references [11][14][15][16] for information on OCR-based methods.

Methods for word spotting often start by segmenting the document image into words using a priori knowledge about the distance between characters and words. An early method of this type has been introduced by Manmatha et al. [9], as a new alternative to OCR at the time. The authors presented two algorithms for word spotting by estimating the shift between the query and the words in the document image. The document image is subject to normalization and segmentation into words. Then, the number of words is pruned using the areas and aspect ratios of the words. The two spotting algorithms calculate the shift between the query word and the document image words using the Euclidean distance and Scott and Longuet Higgins' algorithm.

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.498

Another segmentation-based method has been presented by Rath and Manmatha [17]. The authors tackled the problem of word spotting in historical documents. After segmenting the document image into words, the ink pixel distribution is used for word feature extraction. Matching is done using Dynamic Time Warping [18].

Other word spotting methods use a priori knowledge about the document's language. For instance, Lu and Tan presented a method for word spotting in Chinese documents [19]. They employ a connected component tracing algorithm to segment the query and document image into strokes, then a grouping algorithm to merge the connected components into characters. After pruning the number of words in the document image, one character from the query is matched against the document image words, by using a modified Hausdorff distance that has been tuned for Chinese characters. Then, the remaining characters in the query words are used to validate the matches.

Likewise, Sari and Kefali presented a method for word spotting in Arabic documents [20]. Connected components are extracted from the query and document image. Then, the connected components are represented by specific features of Arabic characters (e.g. diacritics, loops, etc.). Matching is done using a string matching algorithm.

Other researchers aimed for language-invariance. In [13], Lee et al. use the SIFT descriptor [21] for word spotting. The user's query is introduced online and normalized in a specific font. Then, SIFT is used for feature extraction and matching of the query in the document image. Query occurrences are detected using clustering. Their approach has been evaluated using English and Korean documents, and demonstrated to be language-invariant.

Domain-specific approaches have been introduced. For instance, Zhu et al. presented a framework for signature detection and matching [22]. Their approach is based on the view that signatures posses a multiscale characteristic structural saliency.

Approaches for mathematical expression spotting have been also presented, though less focus has been devoted to this area. Zanibbi and Yu introduced an approach for mathematical expression spotting using handwritten queries [8]. Their approach works as follows: Recursive X-Y Cutting [23] is used to produce X-Y trees for the document image and the query, and pruning is used to discard irrelevant regions such as text. Then, spotting is done by looking up the query in the document image index using features of its X-Y tree, producing a set of candidates. Candidate ranking is done using Dynamic Time Warping.

In another work [24], Zanibbi and Yu introduced a spotting approach that allows the user to input different formats of query (e.g. handwritten image, query in LaTeX format).

Summary

Many approaches have been presented for query spotting in document images. Two characteristics can be noticed in these methods:

- Domain-specificity: Such as language or query class dependence.

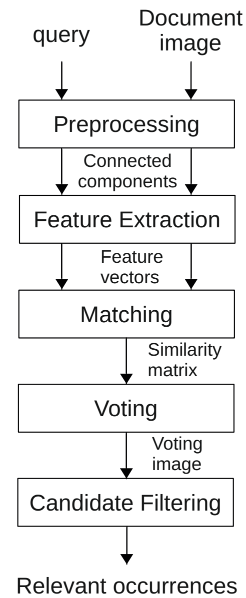


Fig. 1. Overview of the proposed approach.

- Prior segmentation of the document image in order to extract regions of interest such as text paragraphs or mathematical calculations.

In this work, we aim to design an application-independent approach that considers all the document image information without prior segmentation.

III. The Proposed Approach

In this section, we explain our approach for query spotting in document images in detail. Fig. 1 shows an overview of our approach: First, the query and document image are subjected to a Preprocessing step that is charged of noise reduction and *connected components* extraction (for the ease of description, we will refer to connected components simply as components).

Then, features are extracted from components and represented with *feature vectors*.

Afterwards, the feature vectors corresponding to the query components and the document image components are matched and the similarity scores are stored in a *similarity matrix*.

Next, *similarity matrix* is used to vote for locations of candidate occurrences of the query in the document image.

Finally, relevant occurrences are detected and irrelevant patterns are filtered out.

A. Preprocessing

Document images are usually prone to noise due to the quality and age of the document and imperfection of scanning devices. Therefore, a preprocessing step for noise reduction and input normalization is needed. For this purpose, we use our previously reported *Adaptive Thinning Framework (ATF)* [1][2]. ATF produces a 1-pixel width representation of images, and it is robust against noise compared to conventional thinning algorithms.

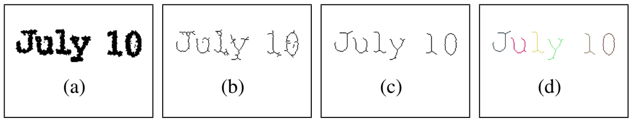


Fig. 2. Image preprocessing: (a) Original image. (b) Thinning using a conventional algorithm [25]. (c) Thinning using ATF. (d) Components extraction.

After preprocessing, components are extracted from the image. Fig 2 illustrates this procedure.

The output of this step is the components of the query and document image, respectively $\{C_i^Q\}_{i\leq M}$ and $\{C_j^{DOC}\}_{j\leq N}$, where M and N are the number of components of the query and document image.

B. Feature extraction

The inputs of this step are the components of the query and document image, respectively $\{C_i^Q\}_{i\leq M}$ and $\{C_j^{DOC}\}_{j\leq N}$.

For each component, a feature vector is generated using a shape descriptor, as shape is the only information available after the Preprocessing step.

In this work, we use the feature extraction mechanism described in the Contour Points Distribution Histogram (CPDH) shape descriptor [26]. For each component C of the query and the document image, a feature vector $\overrightarrow{H}$ is extracted as follows: The distribution of shape points in the shape enclosing circle is calculated in polar coordinates. Then, the point distribution is represented in a 2-dimensional histogram of norms and angles.

Due to the use of the enclosing circle, CPDH is scale-invariant, and rotation-invariance can be achieved by using shifted matching. In addition, the feature extraction stage of CPDH is computationally efficient.

The output of this step is the feature vector sets $\{\overrightarrow{H}_i^Q\}_{i\leq M}$ and $\{\overrightarrow{H}_j^{DOC}\}_{j\leq N}$, corresponding to $\{C_i^Q\}_{i\leq M}$ and $\{C_j^{DOC}\}_{j\leq N}$.

C. Matching

This step performs matching of $\{\overrightarrow{H}_i^Q\}_{i\leq M}$ and $\{\overrightarrow{H}_j^{DOC}\}_{j\leq N}$ and stores the similarity scores in a *similarity matrix* $S_{M,N}$. Each cell $S(i,j)$ is calculated using the Histogram Intersection measure between $\overrightarrow{H_i^Q}$ and $\overrightarrow{H_j^{DOC}}$ as follows:

$$S(i,j)=\sum_{k=0}^{K-1}\sum_{l=0}^{L-1} min(H_{kl}^Q, H_{kl}^{DOC}) \tag{1}$$

where K and L are the norm and angle dimensions of the CPDH feature vector. $S(i,j)$ takes real values in the interval $[0,1]$. Large values express similarity between components, and small values express dissimilarity.

At this stage, S holds the similarity scores between the components of the query and all the components of the document image. For the sake of efficiency, a pruning step can be envisaged by thresholding S to keep only significant similarity scores. However, such a pruning method is not sufficient as most shape descriptors are prone to false positive. Therefore, before applying this method, we apply another pruning mechanism.

Our mechanism relies on the view that if a document image component $C_{j_0}^{DOC}$ is visually similar to a query component $C_{i_0}^Q$, it should be nearly similar or dissimilar to the remaining components of the query $\{C_i^Q\}_{i\neq i_0}$, in the same way as $C_{i_0}^Q$.

Algorithm 1 Pruning mechanism

$S_{M,M}^Q \leftarrow \{\sum_{k=0}^{K-1}\sum_{l=0}^{L-1} min(H_{kl(u)}^Q, H_{kl(v)}^Q)\}_{u\leq M, v\leq M}$
for j_0 from 1 to N **do**
  for i_0 from 1 to M **do**
    if $S(i_0,j_0)>\alpha$ **and** $D(i_0,j_0)>\theta$ **then**
      prune($C_{j_0}^{DOC}$)
    end if
  end for
end for

Algorithm 1 explains the pruning mechanism; The query auto-correlation matrix, $S_{M,M}^Q$, is calculated by matching the query's components against each other using the Histogram Intersection measure (Eq. 1). S^Q is symmetric with 1-values on the diagonal. Afterwards, if a document image component $C_{j_0}^{DOC}$ is found to be similar to a query component $C_{i_0}^Q$ (i.e. $S(i,j)>\alpha$), but does not keep a similarity pattern to $\{C_i^Q\}_{i\neq i_0}$ in the same way as $C_{i_0}^Q$ does, then $C_{j_0}^{DOC}$ is pruned.

The similarity pattern between $C_{j_0}^{DOC}$ and $\{C_i^Q\}_{i\neq i_0}$ is estimated by calculating the Euclidean distance between $\{S^Q(i,i_0)\}_{i\leq M}$ and $\{S(i,j_0)\}_{i\leq M}$ as follows:

$$D(i_0,j_0)=\frac{1}{M}\sum_{i=1}^{M}(S^Q(i,i_0)-S(i,j_0))^2 \tag{2}$$

The similarity threshold α and the dissimilarity threshold θ have direct effects on the system's performance. Small values of α and θ increase the number of true negatives. Large values of α and θ lead to keeping a lot of components in the document image, hence increasing the number of true positives but also false positives.

The output of this step is *similarity matrix* S after pruning of false positive and small similarity scores.

D. Voting

The aim of this step is to estimate locations of candidate occurrences of the query in the document image using *similarity matrix* S and the query components relative locations.

The candidate locations are determined by generating a *voting image* I_V, which is a grayscale image that has the same dimensions of the document image, and where bright spots show locations of candidate occurrences of the query. I_V is produced by calculating a *voting matrix* Mat^i corresponding to each query component C_i^Q and then merging the matrices. Below, we detail the method of calculating Mat^i.

Mat^i has the same dimensions as the document image and holds the votes corresponding to a query component C_i^Q. In

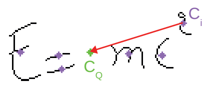


Fig. 3. Illustration of the component centroid c_i, the query centroid c_Q, and $\overrightarrow{c_i c_Q}$. The circles in purple highlight the components' centroids.

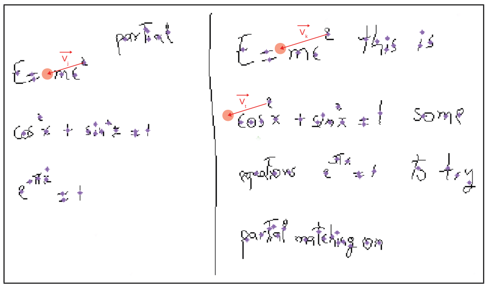


Fig. 4. The *voters* are C_j^{DOC} (the symbol '2' on the left), C_k^{DOC} (the symbol '2' on the right), and C_t^{DOC} (the symbol '2' on the center). Their *voting vectors* are $\overrightarrow{V_i}$, $\overrightarrow{V_k}$, and $\overrightarrow{V_t}$. The *voting vectors* differ in norms to adapt for the size change. The circles in purple highlight the components' centroids.

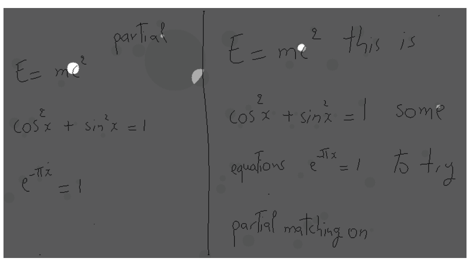


Fig. 5. The voting image I_V. Bright spots show regions of high voting scores.

Mat^i, 1-valued cells show candidate locations that have been voted for by similar $\{C_j^{DOC}\}_{j\leq N}$, and 0-valued cells show absence of voting. A voting operation is determined using the following information:

- The centroid c_j of C_j^{DOC}.
- The displacement vector $\overrightarrow{c_i c_Q}$ connecting the centroid c_i of C_i^Q and the centroid c_Q of the query (Fig. 3).
- The similarity score $S(i,j)$ of matching C_i^Q and C_j^{DOC}.

Next, a *voting vector* $\overrightarrow{V_j}$ is calculated. $\overrightarrow{V_j}$ originates from c_j parallely to $\overrightarrow{c_i c_Q}$ and its norm is as follows:

$$|\overrightarrow{V_j}| = |\overrightarrow{c_i c_Q}| \times \gamma \tag{3}$$

where γ is a scale normalization factor calculated using the radius of the enclosing circles of C_i^Q and C_j^{DOC}. $\overrightarrow{V_j}$ points to the *voting point*, that is the center of the candidate occurrence.

Then, the cells of Mat^i located in a circular region around the voting point are turned 1-valued. The choice of a circular region is in order to account for components displacement. The radius r of the circular region is calculated as follows:

$$r = r^Q \times \gamma \times S(i,j) \times \delta \tag{4}$$

where r^Q is the radius of the query's enclosing circle, and δ is a parameter to control the size of the voting region. δ depends on the image resolution. Small values of δ increase the number of true negatives, while large values increase the number of false positives. Fig. 4 illustrates a voting operation in Mat^i superposed on the document image.

After generating a *voting matrix* Mat^i corresponding to each query component C_i^Q, the matrix Mat^{mean} holding the average of values of votes $\{Mat^i(x,y)\}_{i\leq M}$ is calculated. Then, the entries of Mat^{mean} are mapped into grayscale intensities and used to produce the voting image I_V.

Fig. 5 shows an example of the voting image I_V superposed on the document image. I_V is the output of this step.

E. Candidate filtering

The voting image I_V has been produced using votes from groups of components C_j^{DOC}. In this step, *voting groups* are identified, extracted from the document image, and classified as relevant or irrelevant.

First, the centers of the voting regions $\{c_k^V\}_{k\leq K}$, where K is the total number of voting regions, are extracted from I_V by applying Distance Transform [27] and an intensity maxima detection algorithm.

Next, the components C_j^{DOC} which voting vectors (Eq. 3) point to a location inside the voting circular region centered in c_k^V and which radius is r_k (Eq. 4) are extracted and form a voting group G_k.

At this stage, K voting groups are extracted from the document image. Filtering of irrelevant groups is done through two procedures:

1) Checking the scale consistency of the group components.
2) Global matching of the group with the query image.

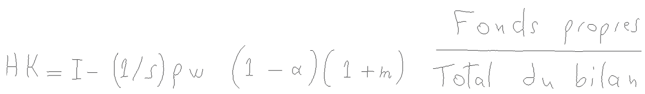
In the first procedure, the idea is that a relevant voting group should have components that hold nearly equal scale factor γ with their corresponding query components. This is insured by calculating the scale factor variance $\sigma(\gamma)$. If $\sigma(\gamma)$ is large, it means that the voting group is formed of components having inconsistent scales.

In the second procedure, the image formed by the voting group pixels is matched with the query using a global image matching algorithm called Support Regions Descriptor (SRD) [28][29]. Then, if the similarity score given by SRD is above a threshold ϵ, the voting group is considered as relevant. A small value of ϵ leads to increasing false positives, while a large value increases true negatives.

The output of this step is the detected relevant voting groups, which are the relevant occurrences of the query in the document image.

IV. Experimental Results

In this section, we present our preliminary experimental results. We aimed to evaluate the approach's performance when spotting handwritten queries in document images with challenging quality.



(a) Query 1 (b) Query 2 (c) Query 3

Fig. 6. Thinned queries used in the experiment.

Evaluation procedure

We prepared an image dataset by converting pages of the journal *Annales de l'insée (Numéro 40, Oct-Dec 1980)* [30] into document images in 200×200 dpi. The dataset contains 104 images that include text and mathematical calculations. The image resolution was 1110×1162.

Throughout the experiment, the parameters were set empirically as follows: The similarity threshold $\alpha = 0.4$, the dissimilarity threshold $\theta = 0.025$, the voting region size parameter $\delta = 0.3$, and the similarity threshold for SRD $\epsilon = 0.4$.

The evaluation was done using 3 queries that were scanned in 300×300 dpi and thinned using ATF (Fig. 6). Statistical analysis of the results was done by calculating *Precision*, *Recall*, and *Pixel Suppression Rate*, as follows:

$$Precision = \frac{Number\ of\ Relevant\ Pixels \times 100}{Number\ of\ Retrieved\ Pixels} \quad (5)$$

$$Recall = \frac{Number\ of\ Relevant\ Pixels \times 100}{Number\ of\ Total\ Relevant\ Pixels} \quad (6)$$

$$Pixel\ Suppression\ Rate = \frac{(N_{Original} - N_{Final}) \times 100}{N_{Final}} \quad (7)$$

where $Number\ of\ Total\ Relevant\ Pixels$ is known from the ground truth, $Number\ of\ Relevant\ Pixels$ and $Number\ of\ Retrieved\ Pixels$ are calculated after the spotting, $N_{Original}$ is the number of pixels in the document image, and N_{Final} is the number of remaining pixels after spotting.

Precision expresses the ability of the approach to find relevant occurrences. *Recall* expresses the ability to find all correct results. As for *Pixel Suppression Rate*, it shows the amount of image information filtered out by the approach.

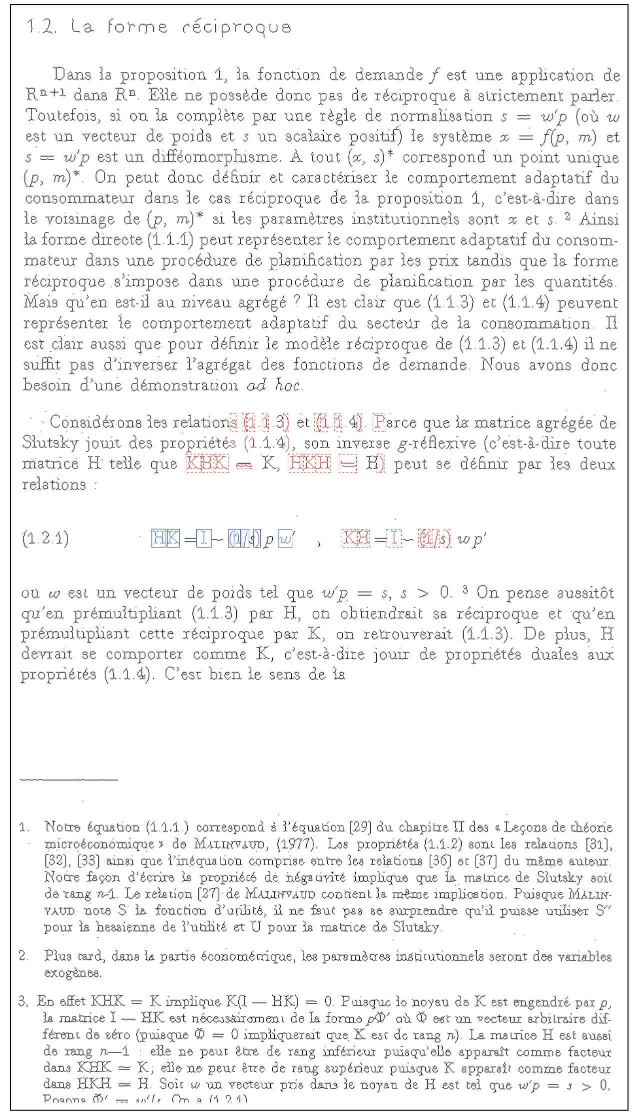
Results and discussion

Table I shows the average values of *Precision*, *Recall*, and *Pixel Suppression Rate* for the 3 queries in Fig. 6. The average *Recall* values show that the approach was able to spot more than 70% of the queries in the document images. The average *Precision* show that false positives are being spotted. However, the fact that *Pixel Suppression Rate* took large values indicates that the approach removes a large part of the document image and leaves only few false positives. In most spotting applications, finding occurrences of the query is the most important, and false positive detection is usually an expected part of the result, particularly in document images of challenging quality such as the ones used in our experiment.

Fig. 7 shows the result of spotting the query in Fig. 6(a) in a document image from the dataset. In this example, the approach correctly retrieved most of the query components, and removed much of the image irrelevant pixels. Other patterns of the document image that share similar components with the query have been detected as false positives.

TABLE I. AVERAGE VALUES OF *Precision*, *Recall*, AND *Pixel Suppression Rate* CORRESPONDING TO 3 QUERIES.

Query	Precision	Recall	Pixel Suppression Rate
Query 1	49.20%	71.30%	99.15%
Query 2	56.71%	71.78%	99.51%
Query 3	52.88%	73.56%	98.18%



1.2. La forme réciproque

Dans la proposition 1, la fonction de demande f est une application de R^{n+1} dans R^n. Elle ne possède donc pas de réciproque à strictement parler. Toutefois, si on la complète par une règle de normalisation $s = w'p$ (où w est un vecteur de poids et s un scalaire positif) le système $x = f(p, m)$ et $s = w'p$ est un difféomorphisme. A tout $(x, s)^*$ correspond un point unique $(p, m)^*$. On peut donc définir et caractériser le comportement adaptatif du consommateur dans le cas réciproque de la proposition 1, c'est-à-dire dans le voisinage de $(p, m)^*$ si les paramètres institutionnels sont x et s. [2] Ainsi la forme directe (1.1.1) peut représenter le comportement adaptatif du consommateur dans une procédure de planification par les prix tandis que la forme réciproque s'impose dans une procédure de planification par les quantités. Mais qu'en est-il au niveau agrégé ? Il est clair que (1.1.3) et (1.1.4) peuvent représenter le comportement adaptatif du secteur de la consommation. Il est clair aussi que pour définir le modèle réciproque de (1.1.3) et (1.1.4) il ne suffit pas d'inverser l'agrégat des fonctions de demande. Nous avons donc besoin d'une démonstration *ad hoc*.

Considérons les relations (1.1.3) et (1.1.4). Parce que la matrice agrégée de Slutsky jouit des propriétés (1.1.4), son inverse g-réflexive (c'est-à-dire toute matrice H telle que KHK = K, HKH = H) peut se définir par les deux relations :

$$HK = I - (1/s)\, p\, w' \quad , \quad KH = I - (1/s)\, w\, p' \qquad (1.2.1)$$

où w est un vecteur de poids tel que $w'p = s$, $s > 0$. [3] On pense aussitôt qu'en prémultipliant (1.1.3) par H, on obtiendrait sa réciproque et qu'en prémultipliant cette réciproque par K, on retrouverait (1.1.3). De plus, H devrait se comporter comme K, c'est-à-dire jouir de propriétés duales aux propriétés (1.1.4). C'est bien le sens de la

1. Notre équation (1.1.1.) correspond à l'équation [29] du chapitre II des « Leçons de théorie microéconomique » de MALINVAUD, (1977). Les propriétés (1.1.2) sont les relations [31], [32], [33] ainsi que l'inéquation comprise entre les relations [36] et [37] du même auteur. Notre façon d'écrire la propriété de négativité implique que la matrice de Slutsky soit de rang n-1. La relation [27] de MALINVAUD contient la même implication. Puisque MALINVAUD note S la fonction d'utilité, il ne faut pas se surprendre qu'il puisse utiliser S'' pour la hessienne de l'utilité et U pour la matrice de Slutsky.

2. Plus tard, dans la partie économétrique, les paramètres institutionnels seront des variables exogènes.

3. En effet KHK = K implique K(I — HK) = 0. Puisque le noyau de K est engendré par p, la matrice I — HK est nécessairement de la forme $p\Phi'$ où Φ est un vecteur arbitraire différent de zéro (puisque $\Phi = 0$ impliquerait que K est de rang n). La matrice H est aussi de rang n—1 : elle ne peut être de rang inférieur puisqu'elle apparaît comme facteur dans KHK = K; elle ne peut être de rang supérieur puisque K apparaît comme facteur dans HKH = H. Soit w un vecteur pris dans le noyau de H est tel que $w'p = s > 0$. Posons $\Phi' = w'/s$. On a (1.2.1)

Fig. 7. Result of spotting the query in Fig. 6(a) in a document image from the dataset: All the black pixels are suppressed after spotting ($Pixel\ Suppression\ Rate = 97.92\%$) and only colored pixels remain. The blue pixels inside blue rectangles are relevant spotting results, and the red pixels inside dashed red rectangles are incorrect spotting results. Here, $Precision = 21\%$ and $Recall = 61.53\%$.

The pruning step using the query auto-correlation matrix removes around 50% of the document image content. Instead of direct comparison between components, the advantage of this pruning method lies in comparing between patterns of similarity and dissimilarity between a document image component and a set of query components. By doing so, it attenuates the effect of handwriting and standard font variations.

The results indicate promising performances since most of the image irrelevant information is suppressed and most of the query components are spotted correctly. As for the spotting precision, the results suggest that a post-processing that further filters out false positives should be conceivable. Additionally, a trade-off between *Precision* and *Recall* is possible by adjusting the similarity threshold α, the dissimilarity threshold θ, the voting region size parameter δ, and the similarity threshold for SRD ϵ. These parameters can be tuned to reject false positives from the early steps of the approach.

To the best of our knowledge, available spotting methods do not offer application-independence and are often based on segmentation. Therefore, direct comparison of our approach with available methods is not adequate.

V. Conclusion and Future Work

We reported our ongoing research on an application-independent and segmentation-free approach for spotting multipart queries in document images, that has been built on an earlier work reported in [1]. The proposed approach finds occurrences of a query in a document image through 5 steps that remove irrelevant pixels by means of feature matching and pruning, and use the relative locations of the query components to vote for candidate occurrences in the document image.

Preliminary experimental results show promising performances and possibility of further improvement. Our next direction is to make the parameters automatic and adaptive to the query and document image, and to conduct large scale experiments with various queries.

Being application-independent and segmentation-free makes the proposed approach radically different than existing methods. At the same time, having a modular aspect makes the approach configurable to specific applications and able to incorporate a-priori knowledge. For instance, the number of components in the query can be of a critical importance (e.g. automatic evaluation of student assignments) and hence it can be used as an additional condition in the Candidate Filtering step. Whereas in other applications such a condition does not hold the same priority (e.g. query spotting in distorted old historical documents).

References

[1] H. Chatbri and K. Kameyama, "Towards making thinning algorithms robust against noise in sketch images," in *International Conference on Pattern Recognition (ICPR'12)*. IEEE, 2012, pp. 3030–3033.

[2] H. Chatbri and K. Kameyama, "Using scale space filtering to make thinning algorithms robust against noise in sketch images," *Pattern Recognition Letters*, vol. 42, no. 0, pp. 1–10, 2014.

[3] S. Marinai, B. Miotti, and G. Soda, "Digital libraries and document image retrieval techniques: A survey," in *Learning Structure and Schemas from Documents*. Springer, 2011, pp. 181–204.

[4] I. Siddiqi and N. Vincent, "A set of chain code based features for writer recognition," in *International Conference on Document Analysis and Recognition (ICDAR'09)*. IEEE, 2009, pp. 981–985.

[5] N. Chen and D. Blostein, "A survey of document image classification: problem statement, classifier architecture and performance evaluation," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 10, no. 1, pp. 1–16, 2007.

[6] Z. Shi and V. Govindaraju, "Historical document image enhancement using background light intensity normalization," in *International Conference on Pattern Recognition (ICPR'04)*, vol. 1. IEEE, 2004.

[7] M. I. Shah and C. Y. Suen, "Word spotting techniques in document analysis and retrieval-a comprehensive survey," *Handbook of Pattern Recognition and Computer Vision*, vol. 4, pp. 353–376, 2010.

[8] R. Zanibbi and L. Yu, "Math spotting: Retrieving math in technical documents using handwritten query images," in *International Conference on Document Analysis and Recognition (ICDAR'11)*, 2011.

[9] R. Manmatha, C. Han, and E. M. Riseman, "Word spotting: A new approach to indexing handwriting," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, 1996.

[10] M. Bokser, "Omnidocument technologies," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1066–1078, 1992.

[11] A. Kae and E. G. Learned-Miller, "Learning on the fly: Font free approaches to difficult ocr problems," in *ICDAR'09*, 2009.

[12] K. Tamura, T. Yoshikawa, and T. Furuhashi, "A study on document retrieval system based on visualization to manage ocr documents," in *Human-Computer Interaction. Interaction Modalities and Techniques*. Springer, 2013, pp. 740–749.

[13] D.-R. Lee, W. Hong, and I.-S. Oh, "Segmentation-free word spotting using SIFT," in *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*. IEEE, 2012, pp. 65–68.

[14] J. Ha, R. M. Haralick, and I. T. Phillips, "Understanding mathematical expressions from document images," in *International Conference on Document Analysis and Recognition (ICDAR'95)*, vol. 2. IEEE, 1995.

[15] R. Zanibbi, A. Pillay, H. Mouchere, C. Viard-Gaudin, and D. Blostein, "Stroke-based performance metrics for handwritten mathematical expressions," in *International Conference Document Analysis and Recognition (ICDAR'11)*. IEEE, 2011, pp. 334–338.

[16] R. Zanibbi and D. Blostein, "Recognition and retrieval of mathematical expressions," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 15, no. 4, pp. 331–357, 2012.

[17] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 9, no. 2-4, pp. 139–152, 2007.

[18] D. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD workshop*, vol. 10, no. 16, 1994, pp. 359–370.

[19] Y. Lu and C. L. Tan, "Word spotting in chinese document images without layout analysis," in *International Conference on Pattern Recognition (ICPR'02)*, vol. 3. IEEE, 2002, pp. 57–60.

[20] T. Sari, A. Kefali *et al.*, "A search engine for arabic documents," in *Dixième Colloque International Francophone sur l'Ecrit et le Document*, 2008, pp. 97–102.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, 2004.

[22] G. Zhu, Y. Zheng, D. Doermann, and S. Jaeger, "Signature detection and matching for document image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, 2009.

[23] H. R. M. Ha, Jaekyu and I. T. Phillips, "Recursive XY cut using bounding boxes of connected components," in *International Conference on Document Analysis and Recognition (ICDAR'95)*, vol. 2, 1995.

[24] R. Zanibbi and B. Yuan, "Keyword and image-based retrieval of mathematical expressions," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2011.

[25] F. Zhang, Y.-s. Wang, C.-y. Gao, S.-c. Si, and J.-q. Xu, "An improved parallel thinning algorithm with two subiterations," *Optoelectronics Letters*, vol. 4, pp. 69–71, 2008.

[26] X. Shu and X.-J. Wu, "A novel contour descriptor for 2D shape matching and its application to image retrieval," *Image and vision Computing*, vol. 29, no. 4, pp. 286–294, 2011.

[27] G. Borgefors, "Applications using distance transforms," *Aspects of Visual Form Processing*, pp. 83–108, 1994.

[28] H. Chatbri, P. Kwan, and K. Kameyama, "Sketch-based image retrieval by size-adaptive and noise-robust feature description," in *International Conference on Digital Image Computing Techniques and Applications (DICTA'13)*. IEEE, 2013, pp. 469–476.

[29] H. Chatbri and K. Kameyama, "Sketch-based image retrieval by shape points description in support regions," in *International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2013, pp. 19–22.

[30] Annales de l'insée (numéro 40, Oct-Dec 1980). [Online]. Available: http://www.jstor.org (Accessed: December 2013)