Spotting Graphical Symbols in Camera-Acquired Documents in Real Time

Marçal Rusiñol, Dimosthenis Karatzas and Josep Lladós
Computer Vision Center, Dept. Ciències de la Computació
Edifici O, Univ. Autònoma de Barcelona
08193 Bellaterra (Barcelona), Spain.

***Abstract*—In this paper we present a system devoted to spot graphical symbols in camera-acquired document images. The system is based on the extraction and further matching of ORB compact local features computed over interest key-points. Then, the FLANN indexing framework based on approximate nearest neighbor search allows to efficiently match local descriptors between the captured scene and the graphical models. Finally, the RANSAC algorithm is used in order to compute the homography between the spotted symbol and its appearance in the document image. The proposed approach is efficient and is able to work in real time.**

I. Introduction

The problem of locating and recognizing specific graphical symbols within documents images has received the attention of the graphics recognition community for many years. Being a mature enough research topic [1], many different approaches can be found in the literature [2].

Usually, ad-hoc hand-crafted descriptor features [3] together with time consuming classifiers [4] or distances [5] are often proposed to solve the graphics recognition problem. However, most of the existing methods might not scale well to large scenarios mainly for their computational complexity [6] and their lack of generality. However, we strongly believe that recent proposals from other computer vision communities such as object recognition, scene classification or information retrieval at large can be easily adapted to the graphics recognition domain resulting in much generic and scalable methods. Although in the last years several works can be found that already used such techniques [7] in the graphics recognition context, we are surprised of seeing proposals that still completely disregard such progresses.

Following this idea, we present in this paper a graphics spotting application that work in real time with camera-acquired images of documents from various sources. For that purpose we have used a quite recent off-the-shelf lightweight object matching framework combined with an also well-known approximate nearest-neighbor method that allows the detection and the classification to be performed in real time.

Specifically, the detection and classification of graphical symbols is performed by matching ORB features with FLANN indexing. A final RANSAC step is used in order to obtain the homography. In a first test scenario we printed several invoices from different providers, cropped their logotypes and tried to detect and classify those in a video feed from a standard webcam. The implemented method is able to work at real time with images at 1280x1024 resolution.

In the next sections we will briefly review the used key-point detector and descriptor together with the matching scheme we use. In the results section we will present several application scenarios in which such framework can be applied.

II. Feature Extraction

Graphical symbol recognition within images being a particular case of the object recognition problem, we used the well-known and popular framework of matching local descriptors extracted over interest key-points [8]. But not all of the available key-point matching strategies are efficient enough to be used in real time. We have thus used a set of selected methods that, although they might not present the better performances, they are fast to compute.

In order to extract interest key-points from a webcam feed, we have used the FAST key-point detector [9]. FAST is a high-speed corner detector that use machine learning techniques in order to speed-up the response of whether a candidate pixel is in fact a corner. We can see in Figure 1 the obtained corner points that will be used to extract the local descriptors. In our experiments we appreciated that such corners are robust and stable despite their efficient extraction.

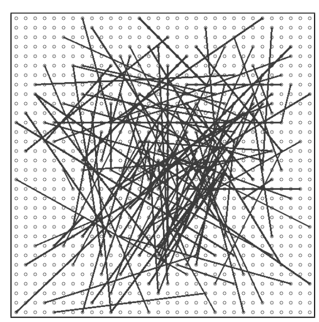


Fig. 2. Example of the binary test distribution from the BRIEF descriptor, extracted from [10].

Once the key-points are extracted, local descriptors are computed over them. A large amount of local descriptors have been proposed in the recent object recognition literature [11], however, most of them are not efficient enough to be used in real-time applications. One of the most promising local descriptors is the efficient BRIEF descriptor [10]. BRIEF being a binary descriptor, aims at quickly comparing local features while requiring few amounts of memory.

The BRIEF descriptor outputs a set of bits obtained by comparing intensities of pairs of pixels within the local key-

Fig. 1. Example of the extracted keypoints.

region. Such pairs of pixels in which the binary tests are performed are sampled from an isotropic Gaussian distribution since the pixels at the patch center ought to be more stable than in the patch perimeter. We can see an example of such binary test distribution in Figure 2, extracted from [10].

However, the original FAST and BRIEF detector and descriptor presented some flaws. On the one hand FAST does not produce multi-scale features so we might have some problems when trying to find graphical symbols in a scene at a different scale from their model. On the other hand, in BRIEF, since the binary tests were performed over a static point distribution, the method is not rotation invariant. Although a recent contribution by the authors propose how to achieve such invariances [12], in our work we used a slight modification of the BRIEF descriptor known as ORB (Oriented FAST and Rotated BRIEF) presented by Rublee et al. in [13].

The ORB framework propose to compute corners by applying the FAST method over a scale pyramid of the image. By using such pyramid, the obtained key-points are extracted at different scales. In addition, a measure of the corner orientation is extracted by means of the intensity centroid [14]. Then, a rotation-aware version of the BRIEF descriptor is applied to the oriented and scaled key-regions. The ORB framework is then able to match key-points from objects under scale and orientation changes while performing much faster than well-known local detectors and descriptors such as SIFT or SURF.

III. Symbol Matching

In order to perform object recognition by matching local key-points at real time, we need some nearest neighbor matching algorithm efficient enough. Although the BRIEF framework outputs a binary descriptor that can be compared with Hamming distances, there is a number of approximate nearest neighbor methods that allow an important speedup of the matching step. In this paper we used the FLANN library [15] which provide an interface to several approximate nearest neighbor algorithms such as randomized kd-trees, hierarchical k-means trees, local sensitive hashing (LSH), etc. and which automatically tunes the parameters of such methods in order to obtain the desired precision degree. In our scenario, since we deal with binary features we decided to use the LSH algorithm [16] in order to index our data. We can see in Figure 3 an example of the matched local key-points described by ORB features using the FLANN matcher.

Fig. 3. Example of the local key-point matching.

Having several model graphical symbols in our dataset that we want to find in the video feed, the matching step is applied separately to each model. The model which receives more votes is the one that is considered as being present in the scene.

In a final step, we used the well-known RANSAC [17] algorithm in order to filter out the outliers matches and to find

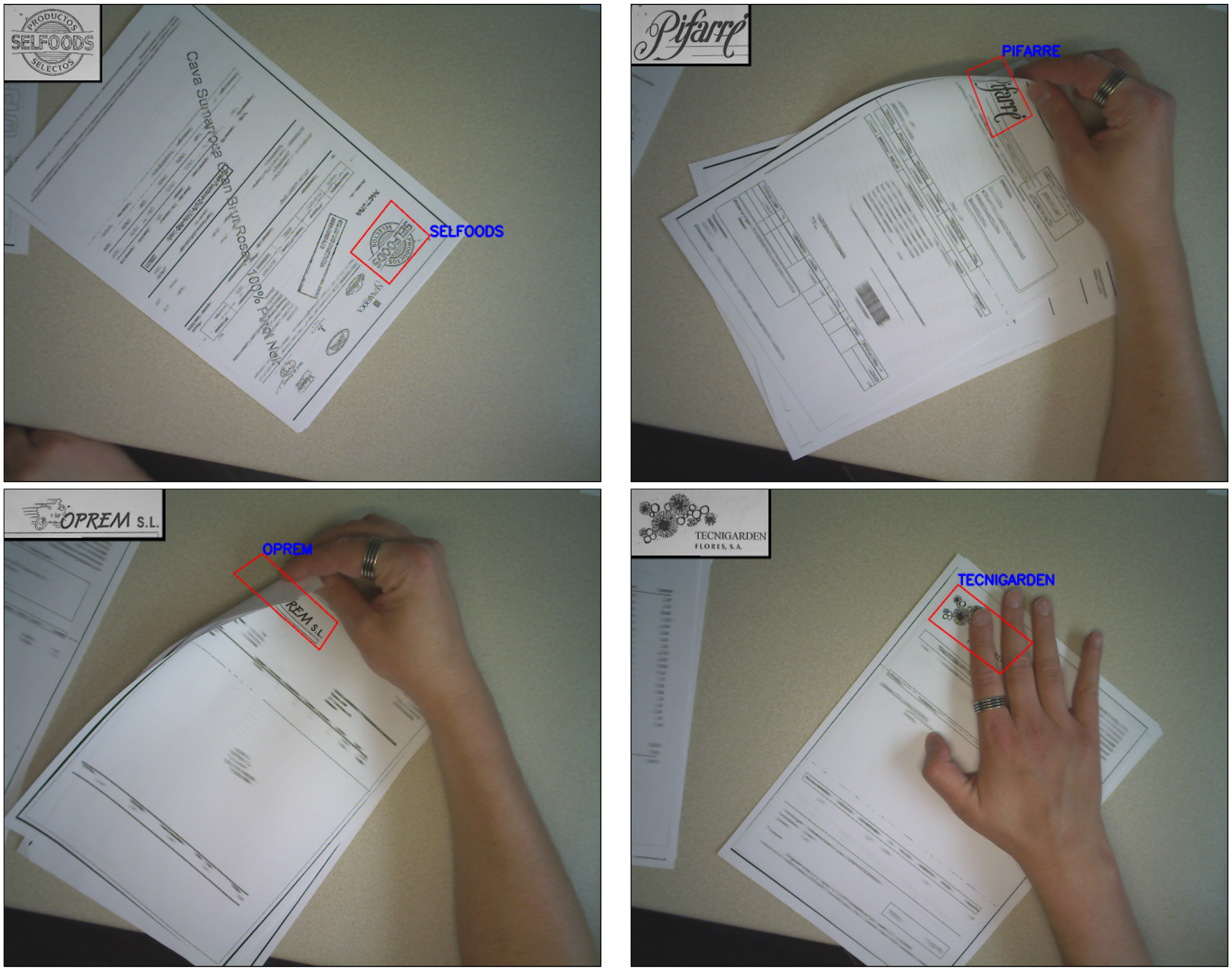


Fig. 4. Screen-shots of the logo spotting in invoice images.

the homography between the recognized model symbol and its instance appearing in the scene.

IV. Experimental Results

Since in this paper we have used off-the-shelf methods that have already been thoroughly evaluated in their respective original papers, rather than providing an exhaustive evaluation of their behavior we preferred to qualitatively assess their performances in the context of graphics-rich documents.

Our main test scenario was to acquire administrative documents, such as invoices, with a webcam and on-the-fly spot the graphical logos that appeared in them. We can see in Figure 4 some screen-shots of this spotting procedure. We refer the reader to our website[1] where the complete video demos are available. We can appreciate that the used framework is able to tolerate well rotation and scale changes, perspective changes, occlusions and illumination changes.

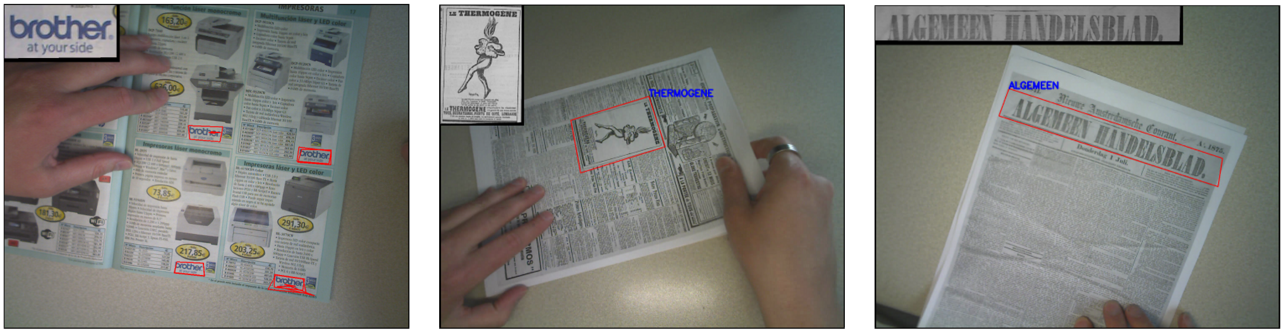
We also tested other application scenarios such as spotting logos in color catalogs or graphical advertisements published in old newspapers. We also tested to spot non-graphical information such as the headlines of those old newspapers also appreciating good performances. We can find some screen-shots of such applications in Figure 6.

Finally, we tested to spot graphical symbols appearing in line-drawing images. In those experiments we found out that really simple symbols represented by low-textured graphics were quite difficult to spot by matching local descriptors, since the BRIEF descriptor proved not to be very discriminant in such scenario. However, for a little bit more complex symbols, such as the ones presented in Figure 5, the method performs well as well.

It is worth to mention that by using the RANSAC method we are assuming that we only look for a single instance of a model symbol in the scene under analysis. In order to be able to spot several symbols in the same image, we divided the image by a multi-resolution overlapping grid. Then the RANSAC method was applied locally and independently at each image cell, allowing to spot several symbols as shown in Figure 6a) and Figure 5.

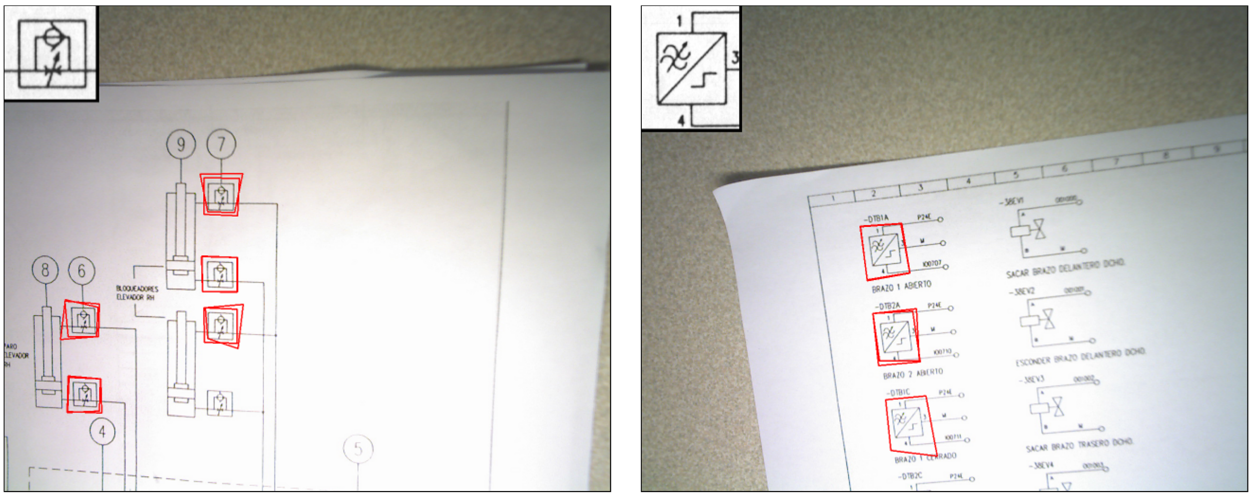
Both ORB features and FLANN indexing library are integrated within the OpenCV library. A Python + Opencv 2.4 version of the source code of the presented method is available through our website.

[1] http://www.cvc.uab.es/~marcal/demos/spotting.html



a) b) c)

Fig. 5. Screen-shots of the symbol spotting in a) color catalogs, b) ads and c) headlines in old newspapers.



a) b)

Fig. 6. Screen-shots of the symbol spotting in line-drawing images.

V. Conclusions

In this paper we have presented how state-of-the-art techniques from the object recognition field can be successfully applied to the problem of symbol spotting. Not only the proposed techniques are performant, but they are also quite efficient, making them very suitable for either real-time applications or applications embedded in smartphones.

We strongly believe that the graphics recognition community should not only focus on ad-hoc recognition processes specifically designed to work in our domain, but rather take advantage of the latest progresses in other computer vision areas.

Acknowledgment

This work has been partially supported by the European 7th framework project FP7-PEOPLE-2008-IAPP: 230653 ADAO, and the Spanish Ministry of Education and Science under projects TIN2009-14633-C03-03, TIN2011-25606 and TIN2012-37475-C02-02.

References

[1] T. Syeda-Mahmood, "Indexing of technical line drawing databases," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 737–751, August 1999.

[2] M. Rusiñol and J. Lladós, *Symbol Spotting in Digital Libraries: Focused Retrieval over Graphic-rich Document Collections*. Springer-Verlag London Limited, 2010.

[3] L. Wenyin, W. Zhang, and L. Yan, "An interactive example-driven approach to graphics recognition in engineering drawings," *International Journal on Document Analysis and Recognition*, vol. 9, no. 1, pp. 13–29, March 2007.

[4] M. Luqman, T. Brouard, and J. Ramel, "Graphic symbol recognition using graph based signature and bayesian network classifier," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2009, pp. 1325–1329.

[5] J. Lladós, E. Martí, and J. Villanueva, "Symbol recognition by error-tolerant subgraph matching between region adjacency graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1137–1143, October 2001.

[6] S. Yang, "Symbol recognition via statistical integration of pixel-level constraint histograms: A new descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 278–281, February 2005.

[7] T. Nguyen, S. Tabbone, and A. Boucher, "A symbol spotting approach based on the vector model and a visual vocabulary," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2009, pp. 708–712.

[8] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.

[9] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, ser. Lecture Notes on Computer Science, 2006, vol. 3951, pp. 430–443.

[10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *European Conference on Computer Vision*, ser. Lecture Notes on Computer Science, 2010, vol. 6314, pp. 778–792.

[11] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, October 2005.

[12] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, July 2012.

[13] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of the International Conference on Computer Vision*, 2011, pp. 2564–2571.

[14] P. Rosin, "Measuring corner properties," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 291–307, February 1999.

[15] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2009, pp. 331–340.

[16] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Communications of the ACM – 50th anniversary issue: 1958 – 2008*, vol. 51, no. 1, pp. 117–122, January 2008.

[17] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.