Novel Line Verification for Multiple Instance Focused Retrieval in Document Collections

Hongxing Gao*‡, Marçal Rusiñol*, Dimosthenis Karatzas*, Josep Lladós*,
Rajiv Jain†, David Doermann†
*Computer Vision Center, Univ. Autònoma de Barcelona, Spain.
†Language and Media Precessing Laboratory, University of Maryland, U.S.A.
‡Hangzhou Hikvision Digital Technology Co. Ltd., China.

***Abstract*—Spatial verification is typically employed to check the spatial consistency among matched local features and to remove outliers. However, when looking for multiple instances of the query within a target image, RANSAC algorithms which are widely applied in many one-to-one matching applications might fail due to the large proportion of "outliers" - correct matches corresponding to other instances. On the other hand, geometrical verification methods are more robust to outliers but usually suffer from high computational costs. In this paper, we introduce a novel two-step line verification method which is more flexible than existing methods and leads to lower computational complexity especially when multiple instances of a query are sought. We study this approach within an information extraction scenario, where the objective is to locate document structures indicative of certain type of information (e.g. different records on invoices).**

I. Introduction

Local keypoint matching based methods such as SIFT, SURF and MSER have achieved great success in many computer vision applications. However, local features might be mismatched if the feature description is not discriminative enough. Spatial verification is usually applied as a post processing technique to filter out the matches that are inconsistent with the dominant transformation between the query and target images. Such a refinement on matched local features has been widely employed for image retrieval [1], object detection [2] and image stitching [3] for example.

The RANSAC algorithm [4] has met great success because it efficiently converges to the dominant spatial transformation for various applications especially for one-to-one matching scenarios (e.g. image stitching, duplicate image detection etc.) However, since RANSAC computes the spatial consistency among the matched features in a global manner, the algorithm might fail when multiple instances of the query are present in the target image. In such cases, a large number of outliers, corresponding to correct matches with instances other than the dominant one, is inevitable. This scenario is of particular importance to document image analysis, as in encompasses a number of information extraction tasks, ranging from word and symbol spotting to the retrieval of structurally similar areas in document collections. In all such applications, the existence of multiple instances of the query on the same page is to be expected.

An alternative to RANSAC like algorithms for computing the spatial consistency among matched local features, is geometrical verification techniques resulting in much higher flexibility. They generally consider local geometry by taking a specific combination of local matched points such as lines and triangles. The transformation relations are then computed based on local geometry while the final transformations are determined by votes in parameter space. Geometrical verification methods allow different transformations and thus are more favourable for multiple instance retrieval scenarios. However, since the geometrical (line/triangle) verifications compute any combination of two or three local matched points, their computation cost (square or cubic) is usually much higher than RANSAC.

In this paper, we propose a two-step line verification strategy for the multiple instance matching scenario. We apply this strategy to structural based focused retrieval in an administrative document retrieval scenario. The matched points are first divided into groups according to the transformation estimated by each pair of points in linear time. In the second step, we employ a line verification algorithm to check the spatial consistency for each group of matched points. In such a two-step manner, only the lines related to the two points from the same group are employed for the transformation estimate while the lines linking points from different groups are ignored. Consequently, for multiple matching scenarios, the two-step line verification holds lower computational complexity than the conventional one and is more robust than RANSAC since the transformations are estimated by local lines.

The rest of this document is organized as follows. Section II will introduced the current algorithms to verify spatial consistency among the matched correspondences while a novel two-step strategy for line verification is proposed in Section III. Section IV will explain an pair-wise retrieval framework to obtain the matched local feature points. The proposed verification method is tested on an invoice dataset in two different retrieval scenarios in Section V. The conclusions of our work are presented in Section VI.

II. Spatial Verification

A spatial verification process is usually employed to estimate the transformation relations or to filter out any "bad" matches (outliers) that are inconsistent with the estimated transformation. Since invoice images employed in our research are clean and upright, shearing is seldom observed between images. Consequently, we only consider transformations in 2-D space. In such cases, the transformation relations generally can be formulated as a transformation matrix as follows.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha\cos\theta & -\alpha\sin\theta & t_x \\ \alpha\sin\theta & \alpha\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x\prime \\ y\prime \\ 1 \end{bmatrix} \qquad (1)$$

where α, θ and (t_x, t_y) represent the scale change, rotation and translation respectively.

The spatial verification process takes n points pairs $P = \{(p_1, p_1\prime), (p_2, p_2\prime), \ldots, (p_n, p_n\prime)\}$ as inputs, whereas $(p_1, p_2, \ldots, p_n)$ and $(p_1\prime, p_2\prime, \ldots, p_n\prime)$ represent the location of the matched points for the query and target images respectively and n denotes the number of the matched points. It estimates the parameters for the transformation matrix shown in Equation 1. Various methods of spatial verification, such as the RANSAC and geometrical verification, have been introduced for many applications to filter out the matches that are inconsistent with the estimated transformation.

A. *RANSAC algorithm based method*

RANSAC is a generic algorithm that deals with outliers. It randomly samples a specific number of matched points as an initial census/pool. It then iteratively performs the two following steps: 1) estimate the value of parameters α, θ and (t_x, t_y) for the transformation matrix using the current census; 2) update the census by adding more points from location pairs P if they are consistent with the estimated transformation and remove the ones that are not consistent with the new transformation from the previous census. The two steps are repeated until the algorithm converges or the given maximum iterations is reached.

Generally, the RANSAC algorithm makes an initial assumption on the transformation, adds new “inliers” from all the observations given the assumption and updates the assumption according to the updated group of “inliers”. Hence, through the two iterative steps, RANSAC is capable of reaching a stable point in parameter space through iterative “minor” revisions. However, when a large proportion of outliers appear in given location pairs, the RANSAC algorithm might fail to obtain the right transformation estimate. The reason is that RANSAC checks the spatial consistency for all matches in a global manner and thus the right solution could easily be hidden by a large amount of noise.

B. *Geometrical Verification*

Geometrical verification refers to an alternative set of methods for verifying the spatial consistency of the matched points. Unlike the global approach employed in RANSAC, geometrical verification checks the spatial consistency based on a small amount of matched points. Limited by the computation complexity, the number of local points is usually set to 2 or 3 which corresponds to a line or a triangle verification respectively.

As shown in Figure 1, triangular geometrical verification compares the corresponding triangles from query and target images. The majority estimations are returned as the final transformations between the query and target images. When there is only a small number of matched points, all combinations of 3 matched points are employed to generate the triangles. However, the computational cost of such a triangular verification process is $O(n^3)$ where n is the number of the matched points. Hence, it becomes infeasible to apply when n grows.

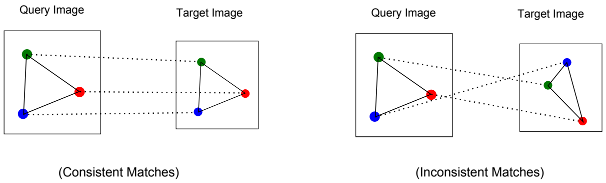


Fig. 1. Geometrical verification by checking the consistency between triangles.

Common strategies to reduce the computation time are: 1) sample a subset with small size from all the matched points. 2) find a reference matched point that appears in all the target images and only compute the triangles related to the reference point. Thus the computational cost is reduced to $O(n^2)$. For example, the number of effective matched points is limited to 25 in [5] while other matches with large distance are not employed. However, as shown in Figure 2, we seek to retrieve all the similar image parts from the dataset while multiple instances are expected within the same target image. Hence, it is not feasible in our case to limit the number of matches into a proper range. In [6], the computational complexity of line verification is reduced to $O(n)$ by first finding a reference point and only considering the lines related to the reference point. Nevertheless, in a focused retrieval scenario where the query corresponds to a specific area of the images, a reference point can not be determined at the page level. Consequently, neither the strategy discussed above serves our situation since there might not exist a stable reference point that appears in all the counterparts.

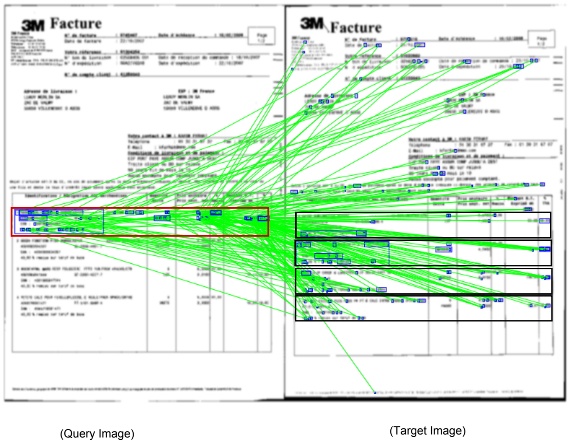
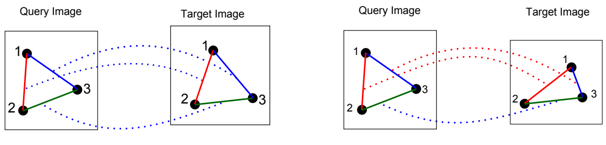

3M Facture
3M Facture
(Query Image)
(Target Image)

Fig. 2. The example of matched points (key-region pairs in our case). The blue bounding boxes correspond to the key-region pairs while the big red and black ones indicate the focused query and its expected matches respectively. Only 20% of the overall matches are shown here for simplicity.

Line verification is a cheaper option for geometrical verification. Like triangular verification, it takes matched X-Y locations and locally estimates the transformation. The difference between these two methods is that the line based method computes the estimation based on lines between points. As shown in Figure 3, the triangle employed in triangular verification boils down to three independent lines which are used to estimate three separate transformations $(\{\alpha_i, \theta_i, t_{xi}, t_{yi}, i = 1, 2, 3\})$ for the matched points. At the end, the transformations are determined by finding the ones that are supported by a large number of lines (observations).

Compared to triangle verification, line verification is less expensive since its computational complexity is $O(n^2)$, while it is more flexible than triangle verification. Assume the transformation shown in Figure 3(a) is the expected one but in reality the location of matched point 1 is shifted as shown in Figure 3(b). In such a situation, triangle verification would not lead to the expected transformation. However, line verification still can give one supporting evidence (line connecting point 2 and 3) to the expected transformation even though the transformations estimated by the other two lines are not consistent.



(a) Expected Transformations

(b) Real Transformations

Fig. 3. Geometrical verification by check the relations between lines and estimate the transformation for the matched points. The dotted lines show the mapping relation of lines. The blue dotted lines correspond to expected transformation estimation and the red dotted line indicate the inconsistent ones.

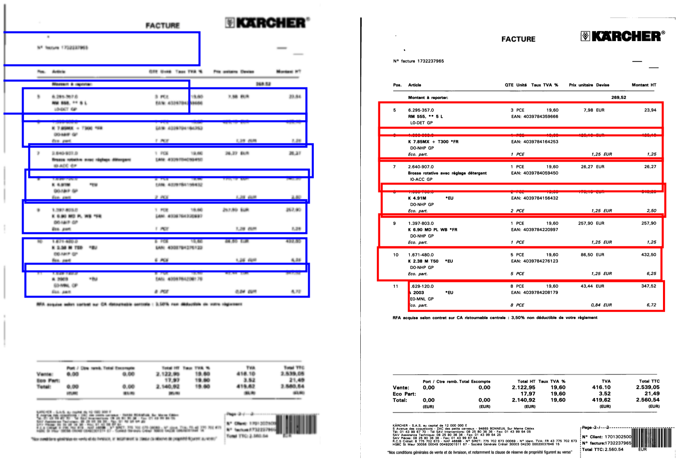
III. Two-step Line Verification

Line verification is much cheaper than triangle verification. However, it is still unaffordable when multiple instances are expected from the same target image which usually leads to a large number of matched points.

In this section, we will introduce a two-step strategy for the line verification method. As shown in Figure 2, when multiple instances are expected in the target image, there is a large number of correct matched points that will positively contribute to the line verification process. However, it is not necessary to employ all the combinations of the points to generate the lines. For instance, the line that links one point from the first bounding box in the target image and another point from another bounding box will not lead to any valid transformation estimation. Consequently, we propose to perform line verification in a two step manner as follows.

- **Step 1**: we first identify the tentative bounding boxes (shown in Figure 4 as the blue rectangles) to divide the matched points into several groups. The tentative bounding boxes are obtained by transforming the query bounding box into target image. Each pair of matched points is employed to compute the corresponding transformation parameters while the scale $\alpha = Area_i / Area_{i'}$ and the rotation $\theta = Orient_i - Orient_{i'}$ where $Area_i$ and $Orient_i$ represent the scale and the orientation of the ith matched point in the query and $Area_{i'}$ and $Orient_{i'}$ correspond to the counterparts in the target image. The translation is determined by the location of the two corresponding matched points. The estimated transformations that at least fit $Thre1$ points are selected as tentative bounding boxes ($Thre1$ is experimentally set to 10.)
- **Step 2**: line verification is employed to precisely check the spatial consistency among the points inside each tentative bounding box. Those bounding boxes are updated by computing the transformation that fits the largest number of matched lines (inliers). As with step 1, we set a threshold $Thre2$ on the number of consistent lines (inliers). When $Thre2 < 3$, the estimated transformation is removed due to insufficient inliers. We also test the performance on repeating step 2 one time more to refine the result.

Compared to conventional line verification, the two-step version is much cheaper with average computational complexity $O(n + k * (\frac{n}{k})^2)$ where k denotes the number of tentative bounding boxes found in step 1. The exclusion of lines outside the bounding boxes leads to a more robust estimation of the transformation due to fewer incorrect lines



Tentative Bounding Boxes

Verified Bounding Boxes

Fig. 4. Example of two-step line verification results.

IV. Retrieval Pipeline

To obtain the matched points, we follow the retrieval framework presented in [7]. The DTMSER algorithm is employed to interpret each document as a key-region tree $T(R, E, l)$ where $R = \{r\}$ is a set of tree nodes corresponding to key-regions that roughly correspond to the structural elements of the document (letters, words, paragraphs). $E = \{e\}$ is a set of directed edges in the tree and represents *inclusion* relations between structural key-regions where $e = (r\_i, r\_j)$ and $r\_i, r\_j \in R$. Each key-region$r$ is described by two different visual features: geometrical feature $f_c(r)$ and SIFT feature $f_g(r)$. A consecutive quantization process$l(r) = l_g(f_g(r)) \times l_c(f_c(r))$ is employed to assign a numerical label to each key-region while l_g and l_c represent the quantization function for the two types of visual features respectively. Afterwards, all the key-regions are stored into a spatial database where spatial index are built to facilitate querying the spatial relation (e.g. *inclusion, top/left of* etc.) between key-regions.

During query time, based on each edge $e = (r_i, r_j) \in E$ of the key-region tree, we cast a pair-wise key-region retrieval process and the spatial database returns the matched pairs $\{(r'_{ik}, r'_{jk}), k = 1, 2, \ldots, n\}$ whereas $l(r'_{ik}) = l(r_i)$ and

$l(r\prime_{jk}) = l(r_j)$ and meanwhile $inclusion$ relations is observed in $(r\prime_{ik}, r\prime_{jk})$ as well.

V. Experimental Results

A. Dataset

We tested the proposed method on the focused document retrieval scenario based on an invoice dataset consisting of 4109 images from 249 providers. The queries correspond to specific areas (e.g. address blocks, headlines, shopping item records etc.) rather than the full-page of the document image. We define 20 focused queries that correspond to 2 groups (10 in each group) according to the expected similarity: *structure-focused* retrieval and *visual-focused* retrieval.

- *Structure-focused* queries – queries aim at retrieving all the structurally similar parts while their exact textual content may change. For example, in our experiment, the invoice items (see Figure 5) that were employed to simulate this type of queries may vary in item name, quantity and price while the structural similarity remains.
- *Visual-focused* queries – queries that aim to exact visual matches where the objects to be matched are exactly reproduced in the target images and thus do not allow content change. In the experiment, we take the invoice headline (see Figure 6) as query and search the matches in the dataset.

We manually created ground truth by specifying all the bounding boxes that corresponding to *structure-focused* queries or *visual-focused* queries. At query time, the transformations are estimated according to the location of the points matched by RANSAC and the proposed line verification. The query bounding box highlighting the focused area is then transformed into the target image and the overlap area ratio criteria between the transformed bounding box and ground truth bounding box is employed to determine if the retrieved matches are correct or not. Inspired by the protocol of PASCAL [8], we set the threshold for this criteria to 50%. We employ Mean Average Precision (MAP) to evaluate the performance of the considered methods.

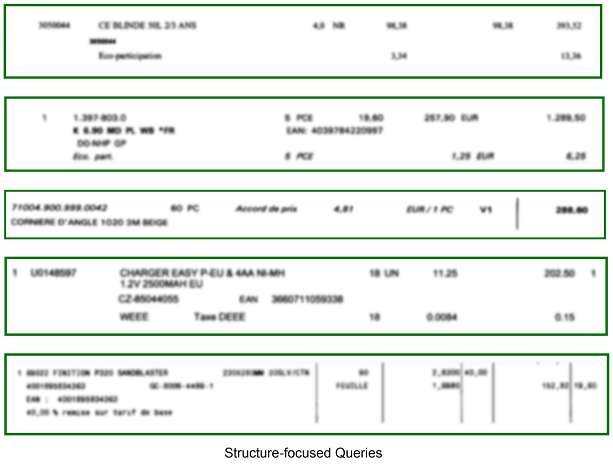

CHARGER EASY P-EU & 4AA NI-MH
Structure-focused Queries

Fig. 5. Samples of the structure-focused queries.

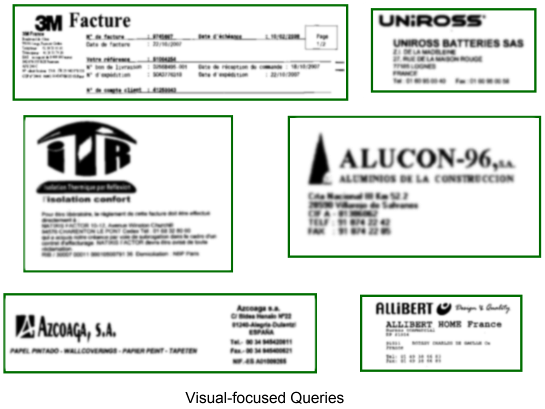

3M Facture
UNiROSS
UNIROSS BATTERIES SAS
ALUCON-96, S.A.
ALUMINIOS DE LA CONSTRUCCION
AZCOAGA, S.A.
ALLIBERT
ALLIBERT HOME France
Visual-focused Queries

Fig. 6. Samples of the queries that seek for visual-focused matches from the collection.

B. Experimental Results

As explained previously, line verification consists of two steps: tentative bounding box estimation by points and line verification based on the lines within each tentative bounding box. We also tested the effect of applying one extra step of the line verification process to further refine the verified bound boxes. *LineVeri1* and *LineVeri2* stand for performing line verification for one and two times respectively while $RANSAC$ corresponds to the performance achieved by RANSAC algorithm.

TABLE I. MAP performance of spatial verification methods on *structure-focused* queries.

	{100,1}	{25,4}	{10,10}	{4,25}	{1,100}
RANSAC	0.6771	**0.6805**	0.6062	0.5780	0.5519
LineVeri1	0.7383	**0.7970**	0.6959	0.7234	0.6110
LineVeri2	0.7400	**0.8243**	0.6887	0.7246	0.6076

We also applied the parameter verification process to find the optimal configuration on the number of centroids for the two type of visual features while the overall codebook size is fixed to 100. The first row of the table corresponds to configurations of $\{n_{geom}, n_{des}\}$ that represent the number of codewords of the types of visual features respectively. As shown in Table I, the optimal parameters ($n_{geom} = 25, n_{des} = 4$) achieve the best performance for both line verification and RANSAC method. For *structure-focused* queries, giving more importance (more codewords) to SIFT feature generally results in worse performance since SIFT feature is very discriminative and not robust on content change.Giving more importance to geometrical features which is robust on content variation would lead to less discriminative power of the visual features and thus in turn let the structural relations between key-region play more important role.

Compared with RANSAC, line verification generally achieves $6 \sim 14$ percent better performance. The reason for such an improvement is that RANSAC computes the transformation in a rigid global manner and thus fails to retrieve the true positives when a large proportion of outliers appear. By contrast, line verification separately estimates the transformations by each of the local lines and hence is more robust to outliers.

To better understand the behaviour of the methods, we analyze the precision and recall separately. As shown in Figure 7, RANSAC would generally achieve higher precision but significantly lower recall than the two line verification methods. Comparing $LineVeri1$ and $LineVeri2$, a remarkable improvement on precision is observed when one extra verification process is applied (see the precision of the query #1, #4, #6, #9, #10). Meanwhile, as shown in 8, such a refinement does not necessarily result in a notable decrease in recall performance. Regarding to the recall of the retrieved result, line verification methods consistently achieve much higher recall performance than RANSAC. Such significant enhancement on recall leads to around 14% improvement on MAP performance.

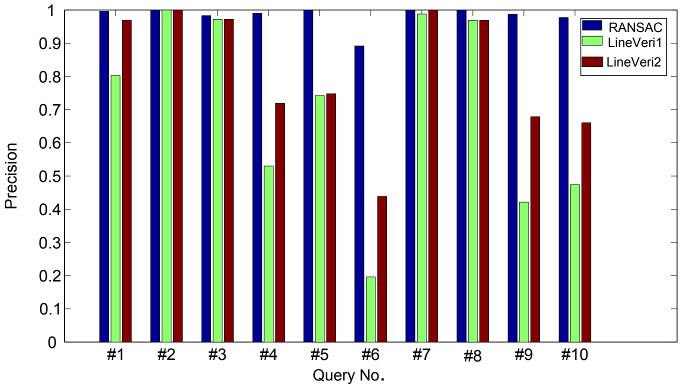


Fig. 7. The precision performance comparison of spatial verification methods for the structure-focused queries.

Apart of the *structure-focused* queries, we also test the performance of the methods for retrieving the *visual-focused* matches. As shown in Table II, the performance of the retrieval with RANSAC has already reached a near-perfect state (the MAP is 0.9938). Line verification makes a small but significant improvement over RANSAC(from 0.9938 to 0.9999).

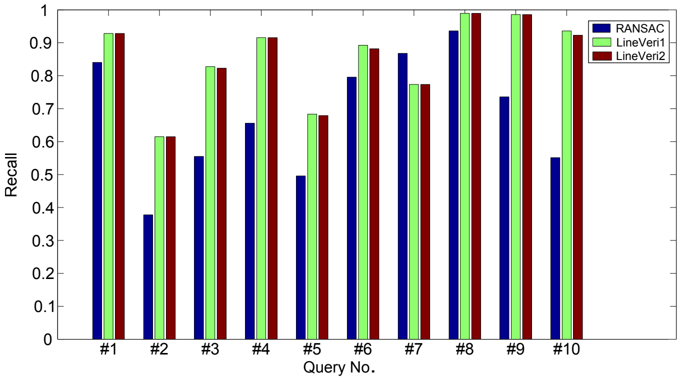


Fig. 8. The recall performance comparison of spatial verification methods for the structure-focused queries.

VI. Conclusions

TABLE II. MAP performance over the focused queries that concentrate on both visual and structural features (visual-focused).

	{100,1}	{25,4}	{10,10}	{4,25}	{1,100}
RANSAC	0.9253	0.9698	**0.9938**	0.9831	0.9921
LineVeri1	0.9828	0.9828	**0.9996**	0.9994	0.9961
LineVeri2	0.9873	0.9889	0.9998	**0.9999**	0.9964

In this paper, we proposed a two-step line verification method to verify the spatial consistency of the matched points. We demonstrated that the RANSAC algorithm might fail to find the right transformations when searching for multiple instances due to a large proportion of outliers. Comparing with the global manner employed in the RANSAC algorithm, line verification verifies the spatial consistency locally and is more flexible in dealing with the "outliers" caused by the location variation of the matched points and thus achieves remarkable improvement especially for structure-focused queries. We demonstrated that the line verification methods achieves much higher recall performance with reasonable loss on precision.

The main disadvantage of line verification is that its cost is $O(n^2)$. However, our system seeks for multiple instance search in single target images and thus leads to a larger number of matched points. Even though we managed to reduce its cost to $O(n + k(\frac{n}{k})^2)$ through a two step strategy, it is still much more expensive than RANSAC. Hence, in the future, it would be important to work on more efficient algorithms with a computational cost closer to $O(n)$. One possible way is to identify a reference points (e.g. geometrical center of the tentative bounding box) and only employ the lines related to the selected reference points.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Education and Science under projects TIN2014-52072-P, and TIN2012-37475-C02-02, by the People Programme (Marie Curie Actions) of the Seventh Framework Programme of the European Union (FP7/2007-2013) under REA grant agreement no. 600388, and by the Agency of Competitiveness for Companies of the Government of Catalonia, ACCIÓ, and China Scholarship Council grant (No.2011674029).

References

[1] Y. Ke and R. Sukthankar, "Efficient near-duplicate detection and subimage retrieval," in *In ACM Multimedia*, 2004, pp. 869–876.

[2] D. G. Lowe, "Object recognition from local scale-invariant features," in *The 7th IEEE International Conference on Computer Vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[3] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.

[4] M. Zuliani, "RANSAC for dummies," 2008.

[5] R. Jain and D. S. Doermann, "Logo retrieval in document images," in *The 10th International Workshop on Document Analysis Systems*, 2012, pp. 135–139.

[6] T. Matsuda, M. Iwamura, and K. Kise, "Performance improvement in local feature based camera-captured character recognition," in *The 11th IAPR International Workshop on Document Analysis Systems*. IEEE, 2014, pp. 196–201.

[7] H. Gao, M. Rusiñol, D. Karatzas, and J. Lladós, "Fast structural matching for document image retrieval through spatial databases," in *Document Recognition and Retrieval XXI*, 2014, pp. 90 210N–90 210N–10.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.