Pattern Recognition Letters 62 (2015) 32–40

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Saliency-based keypoint selection for fast object detection and matching☆

Simone Buoncompagni[a], Dario Maio[a], Davide Maltoni[a], Serena Papi[b,*]

[a] DISI—Università di Bologna, Mura Anteo Zamboni 7-40126 Bologna, Italy
[b] CIRI ICT—Università di Bologna, Via Rasi e Spinelli 146, 47521 Cesena, Italy

ARTICLE INFO

Article history:
Received 5 August 2014
Available online 21 May 2015

Keywords:
Feature selection
Local descriptors
Fast keypoint detection
Keypoints saliency
Augmented reality

ABSTRACT

In this paper we present a new approach to rank and select keypoints based on their saliency for object detection and matching under moderate viewpoint and lighting changes. Saliency is defined in terms of detectability, repeatability and distinctiveness by considering both the keypoint strength (as returned by the detector algorithm) and the associated local descriptor discriminating power. Our experiments prove that selecting a small amount of available keypoints (e.g., 10%) not only boosts efficiency but can also lead to better detection/matching accuracy thus making the proposed method attractive for real-time applications (e.g., augmented reality).

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Keypoints and local descriptors are nowadays largely used for image classification, object detection and recognition, object tracking, image registration and multi-view vision. In the context of object detection/matching, the reference model of a given object can be created by extracting a set of keypoints (e.g., Harris [14], FAST [5], Shi-Tomasi [16], SUSAN [15], Difference of Gaussians [1]) and associating to each of them a local descriptor (e.g. SIFT [1], SURF [2], BRIEF [4]). When the object has to be searched/matched in/against a given test image, a new set of keypoints and related descriptors is extracted and the two sets are matched to consolidate a similarity score. This basic approach works well for some applications but it is not always applicable to real-time scenarios especially when low performance computing architectures are involved. In fact, even if time-efficient keypoint detectors and descriptors are used, the potential very large number of resulting keypoints can require heavy computational demands for the matching phase. Decreasing the number of keypoints reduces the matching complexity but this strategy is effective only if the retained keypoints are stable across different instances of the same object and at the same time the associated descriptors are sufficiently discriminant to avoid increasing matching errors.

In this paper we present a new approach to select keypoints based on their saliency for object detection and matching, under moderate viewpoint and light changes. Keypoint selection is performed through a training stage starting from one or multiple images (views) of the given object. Keypoint saliency is defined in terms of keypoint detectability and descriptor repeatability and distinctiveness. Even though the proposed approach is independent of the keypoint detector and local descriptor, FAST detector [5] and BRIEF descriptor [4] are here used to maximize efficiency thus enabling real-time applications.

The paper includes a summary of systematic experiments conducted on the proposed selection approach and a comparison with another commonly used method. We prove the effectiveness of our technique in a typical real-time augmented reality scenario, where keypoint matching is used to effectively recover the camera viewpoint.

The paper is organized as follows: in Section 2 the recent literature on the subject of this work is reviewed; an overall schema of the proposed approach is presented in Section 3; in Section 4 we provide a formal definition of keypoint saliency and introduce the keypoint selection scheme; Section 5 presents the experiments carried out to evaluate the approach and Section 6 proposes a case study; finally, in Section 7 we draw some conclusions.

2. Related work

The definition of keypoint saliency and the effectiveness of local descriptors have been widely investigated in recent years. Usually, the definition of keypoint saliency takes into account different aspects such as robustness with respect to deformations or descriptor distinctiveness. For example, Amit and Geman [19] define a training procedure to select special points which are likely to be found at certain places in the object but rarely in the background; however they

☆ This paper has been recommended for acceptance by J. Yang.

* Corresponding Author. Tel.: +39 0547338869; fax: +39 0547338890.
E-mail address: serena.papi@unibo.it (S. Papi).

http://dx.doi.org/10.1016/j.patrec.2015.04.019
0167-8655/© 2015 Elsevier B.V. All rights reserved.

do not consider the robustness of associated local descriptors. The concept of robustness of local object appearance represented as probability density function has been investigated by Fergus et al. [6] and Sim and Dudek [8]. Moreover, Pope and Lowe [7] attempt to give an estimation of descriptor detectability and distinctiveness by calculating how often it appears in the learning process. Another memory-based approach was proposed by Nelson [20] that relies on the combination of an associative memory with a Hough-like evidence technique. Ohba and Ikeuchi [9] propose an eigen-based selection of robust descriptors according to their variation with respect to deformations and suggest selection of unique descriptors by checking their distinctiveness compared to other descriptors extracted from training images. In the approaches proposed by Agarwal and Roth [10] and Weber et al. [11] a clustering algorithm is adopted to select patterns most often appearing during the training stage. Dorkó and Schmid [21] train a classifier for each object part and propose a selection of scale invariant feature descriptors to determine the most discriminant ones, whereas Zhang and Košecká [22] introduce a hierarchical approach where a refinement stage is adopted to select only the most discriminative SIFT features and a simple probabilistic model integrates the evidence from individual matches.

The previously cited approaches allow to effectively determine the most discriminant local appearances for a given object class. Nevertheless, most of them do not explicitly consider the intra-class saliency of local appearances in order to establish a ranking of them. A relevant work that quantitatively characterizes keypoint robustness has been proposed by Comer and Draper [17]: their approach tries to determine if a point is repeatable using a generalized linear model (GLM) which is able to predict which points will repeat according to 17 different attributes. The authors use different keypoint detectors such as Lowe's keypoint detector [1] and Harris–Affine keypoint detector [14,18]. Differently from [17] our saliency analysis is not based only on the evaluation of keypoint detection response but it also considers associated local descriptor discriminating power.

The literature contributions closest to our approach are the interesting (and inspiring) work introduced by Carneiro et al. [12,23] and the classification-based prediction of local descriptors matchability recently introduced by Hartmann et al. [33]. Analogously to our approach, [12] and [23] use a training phase where geometric and brightness transformations are used to estimate keypoint/descriptor robustness and to define their saliency. Furthermore, as in [33], our aim is to find out in advance best candidates for matching. However, our approach deviates from [12,23 and 33] in several directions:

- We propose a simplified saliency definition and a different matching schema to boost efficiency and enable real-time operation on low performance architectures; Carneiro and Jepson [12] and Hartmann et al. [33] use a keypoint classifier to filter out (also from the test image) unwanted keypoints and [12] uses a regressor to estimate probability values given in input to a keypoint-assignment validator. On the contrary we perform keypoint selection only during the model computation (training), and our approach relies on simple NN pairing followed by RANSAC consolidation.
- In [33] a training phase is carried out by learning the probability of a given local descriptors to exceed a matching threshold during NN search. On the contrary, our approach does not rely on matching probability against thresholds.
- We focus on recently introduced keypoint detection (FAST) and local descriptors (BRIEF) while experiments in [12] have been performed by using (more accurate but less efficient) SIFT (as well as [33]) and Phase-based local descriptor.
- In our definition of distinctiveness, with the aim of maximizing keypoint/descriptor inter-class diversity, we consider different keypoints of the same images, while in [12] "negative" examples are randomly selected from a separate set of images. We believe our choice is more effective to filter out from the object model those keypoints/descriptors of the object which are similar to each other, in order to reduce the probability of false assignments.

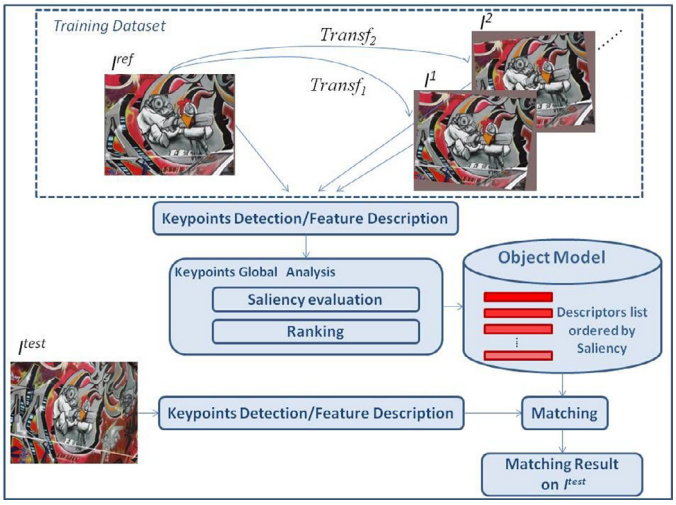

Fig. 1. Overview of the proposed detection/matching method: A preliminary training stage is performed for saliency-based keypoint ranking and selection. The most salient keypoint descriptors which form the object model are the only keypoints matched against test images.

3. Proposed approach: An overview

The proposed approach belongs to the field of object detection and matching by keypoint detection and local descriptors comparison. A full overview of the method is reported in Fig. 1. A training set is composed by a single reference image I^{ref} of the object acquired in neutral viewpoint and lighting conditions and by a set of N generated images $I^1, I^2, \ldots, I^N$ which depict the same object or scene under different conditions. A generic transformed image I^l is obtained by applying a transformation Transf_l, belonging to the set of transformations T, to the reference image as follows:

$$I^l = \text{Transf}_l(I^{\text{ref}}) \tag{1}$$

The nature of Transf_l function is strongly related to the type of transformations characterizing the target application and could be a 2D homography, a 3D projection, a light changing function or a combination of the previous ones.

The training phase starts with the keypoints detection on the reference image I^{ref}; each keypoint is then mapped on the transformed images (through the known Transf_l functions), thus obtaining a set of reference keypoints and their projections. Keypoint descriptors are then computed for all keypoints and a global analysis is performed to rank the keypoints by saliency and select the m-best ones to create the object model. Matching a test image I^{test} against an object model is carried out by detecting the keypoints and computing descriptors from I^{test} and matching them against the model keypoints.

Considering that we are interested in real-time detection/matching, recently proposed FAST [5] and BRIEF [4] have been adopted as detector and descriptor algorithms, respectively. FAST keypoint detection involves simple computations considering only the brightness of the surrounding pixels, whereas BRIEF binary descriptors can be easily extracted through straightforward brightness tests and efficiently matched by bitwise operators.

Saliency evaluation relies on the estimation of keypoint distinctiveness, repeatability and detectability properties:

- *Distinctiveness* quantifies the difference between a given keypoint descriptor and other keypoint descriptors of the same object. Note

that distinctiveness of a keypoint is strongly related to its local descriptor.

- *Repeatability* quantifies the difference between a given keypoint descriptor and corresponding descriptors of projected keypoint on transformed images. Hence, repeatability estimates invariance of local descriptor under different conditions (e.g. viewpoint and lighting). Here too, the repeatability of a keypoint is strongly related to its local descriptor.
- *Detectability* quantifies the aptitude of a given keypoint to be detected under various viewpoint and lighting changes. Unlike distinctiveness and repeatability, detectability of a keypoint is based only on its detection properties (independently of the associated descriptor). For the FAST algorithm this is expressed by a score estimating the corner strength.

A highly distinctive, repeatable and detectable keypoint is an excellent candidate for the matching phase. On the contrary, a point with a low saliency is poorly representative and it could lead to false positive matches. Therefore, focusing on the most salient keypoints not only reduces the computation load but can also improve keypoint matching accuracy.

As previously described, saliency evaluation exploits various images taken under different conditions of viewpoint and lighting. To overcome well-known problems of RGB color space when dealing with light changes, according to other authors ([24,31,32]) we propose to operate in the Opponent color space [24], defined as:

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix} \tag{2}$$

The intensity information is encoded by O_3 whereas the color information is represented by O_1 and O_2. Due to the subtraction in O_1 and O_2, such components are shift-invariant with respect to light intensity. Our experimental results confirm that Opponent color space (OCS in the following) is more effective than RGB space.

4. Saliency evaluation

In this section we introduce a quantitative characterization of keypoints saliency in terms of repeatability, distinctiveness and detectability. We define with $\boldsymbol{x}_i = (u_i, v_i) \in I^{\text{ref}}$ a keypoint selected by the FAST detection algorithm and with s_i the keypoint strength (i.e. amount of cornerness) computed by the FAST algorithm itself. The set of all keypoints of the reference image $K_d(I^{\text{ref}})$ is then:

$$K_d(I^{\text{ref}}) = \{(\boldsymbol{x}_i, s_i) : \boldsymbol{x}_i \in I^{\text{ref}}, i = 1, \ldots, J\} \tag{3}$$

For each $\boldsymbol{x}_i \in K_d(I^{\text{ref}})$, we define with descr : $(\Re^2, \Re^S \times \Re^S) \rightarrow \Re^L$ the function that computes BRIEF descriptor for a keypoint $\boldsymbol{x}_i$ according to the image patch $P(\boldsymbol{x}_i)$ of size S×S centered on $\boldsymbol{x}_i$. Given the nature of BRIEF, $\mathbf{b}_i = \text{descr}(\boldsymbol{x}_i, P(\boldsymbol{x}_i)))$ is a binary vector. Therefore, two binary vectors $\mathbf{b}_i$ and $\mathbf{b}_j$ are compared by using the Hamming distance $H(\mathbf{b}_i, \mathbf{b}_j)$ that can be computed very efficiently through a bitwise XOR operation followed by a bit count.

Distinctiveness: The distinctiveness $D(\boldsymbol{x}_i)$ of a keypoint $\boldsymbol{x}_i \in K_d(I^{\text{ref}})$ is proportional to the diversity among the $\boldsymbol{x}_i$descriptor and the descriptors of other keypoints $\boldsymbol{x}_j \in K_d(I^{\text{ref}}),\ j \neq i$ in the same image. Formally, we estimate the distinctiveness as follows:

$$D(\boldsymbol{x}_i) = \frac{1}{L \cdot (\#K_d(I^{\text{ref}}) - 1)} \sum_{\substack{x_j \in K_d(I^{\text{ref}}) \\ j \neq i}} H(\mathbf{b}_i, \mathbf{b}_j) \tag{4}$$

being L the descriptor length and $K_d(I^{\text{ref}}) > 1$ the total number of keypoints detected on I^{ref}. $D(\boldsymbol{x}_i)$ takes a value in the range [0,1], where 1 denotes maximum distinctiveness.

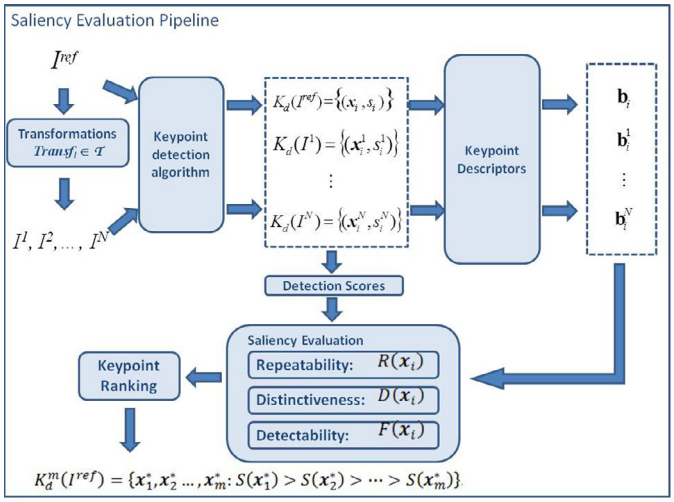

Fig. 2. Detectability, distinctiveness and repeatability are combined in order to obtain a final value of saliency for each keypoint.

Repeatability: The repeatability of a keypoint $\boldsymbol{x}_i \in K_d(I^{\text{ref}})$ is proportional to the similarity among its descriptor $\mathbf{b}_i$ and the descriptors of corresponding keypoints under a set of given transformations. It is defined as:

$$R(\boldsymbol{x}_i) = 1 - \frac{1}{L \cdot \#\mathcal{T}} \sum_{\substack{\boldsymbol{x}_i^l = \text{Transf}_l(x_i) \\ \text{Transf}_l \in \mathcal{T}}} H\left(\mathbf{b}_i, \mathbf{b}_i^l\right) \tag{5}$$

being $\#\mathcal{T}$ the cardinality of the set T and $\mathbf{b}_i^l = \text{descr}(\boldsymbol{x}_i^l, P(\boldsymbol{x}_i^l))$. $R(\boldsymbol{x}_i)$ takes a value close to 1 when a keypoint is highly repeatable.

Detectability: The detectability of a keypoint depends of the score values returned by the keypoint detection algorithm. If detection is performed with FAST, the score is the corner strength [5]. The detectability of a keypoint $\boldsymbol{x}_i \in K_d(I^{\text{ref}})$ is simply an average (normalized in the range [0,1]) over the scores of all keypoints in the original image and its transformed versions:

$$F(\boldsymbol{x}_i) = \frac{1}{\#\mathcal{T}} \sum_{\substack{x_i^l = \text{Transf}_l(x_i) \\ \text{Transf}_l \in \mathcal{T}}} s_i^l \tag{6}$$

where s_i^l is the strength score related to $\boldsymbol{x}_i^l$ and returned by a detection algorithm.

Saliency: Detectability, distinctiveness and repeatability are combined in order to determine the keypoint saliency, as shown in Fig. 2. More precisely, for each keypoint $\boldsymbol{x}_i \in K_d(I^{\text{ref}})$, its saliency is:

$$S(\boldsymbol{x}_i) = \omega_R R(\boldsymbol{x}_i) + \omega_D D(\boldsymbol{x}_i) + \omega_F F(\boldsymbol{x}_i) \tag{7}$$

where ω_R, ω_D and ω_F are weights assigned to repeatability, distinctiveness and detectability, respectively. Optimal values for the weights can be computed by trial and error during experimentations by using a validation set.

It is worth noting that even if detectability and repeatability can be quite correlated, both the contributions are important: the former is crucial to maximize the probability that the detection algorithm will find the keypoint of interest in new images, the latter to maximize the probability that two corresponding keypoints match under the metrics associated to the chosen descriptors. In other words, while detectability is related to keypoint stability under transformation, repeatability and distinctiveness are related to the discriminant power of descriptors. The keypoints of the image can

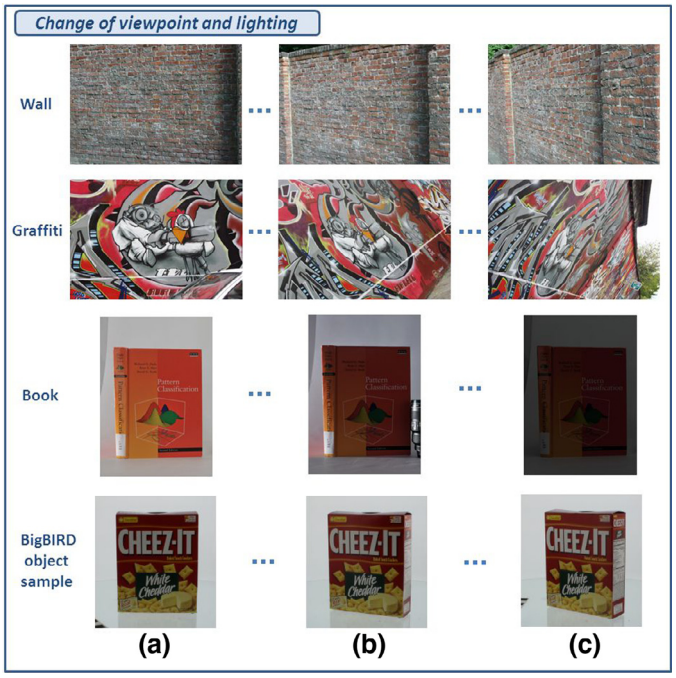

Fig. 3. Datasets used for matching accuracy evaluation under different conditions of viewpoint and lighting: Wall, Graffiti, Book and BigBIRD object example. Column shows for each dataset: (a) the reference images, (b) an image with moderate variation, and (c) an image with high variation.

be ranked according to saliency, resulting in the following ordered set $K_d^*(I^{\text{ref}})$:

$$K_d^*(I^{\text{ref}}) = \{\boldsymbol{x}_1^*, \boldsymbol{x}_2^* \ldots, \boldsymbol{x}_{\#K_d(I^{\text{ref}})}^* : S(\boldsymbol{x}_1^*) > S(\boldsymbol{x}_2^*) > \cdots > S(\boldsymbol{x}_{\#K_d(I^{\text{ref}})}^*)\} \tag{8}$$

Equivalently, we define the m most salient points as:

$$K_d^m(I^{\text{ref}}) = \{\boldsymbol{x}_1^*, \boldsymbol{x}_2^* \ldots, \boldsymbol{x}_m^* : S(\boldsymbol{x}_1^*) > S(\boldsymbol{x}_2^*) > \cdots > S(\boldsymbol{x}_m^*)\} \tag{9}$$

The best choice for the value m is related to the specific application in order to achieve at the same time good matching accuracy and speed, as properly shown in Sections 5 and 6.

5. Experimental results

Many experiments have been carried out on various real images acquired under different conditions of viewpoint and lighting. In particular:

- a first round of experiments has been carried out on small datasets with the aim of tuning parameters and selecting the most effective color space;
- a second round of experiments is then performed (without further adjusting parameters) on a larger dataset to validate results;
- our approach is then compared with other state-of-the-art techniques including the native ranking of FAST detector and the ranking method proposed in [33]; a direct comparison with the approach by Carneiro and Jepson [12] would be interesting too, but the authors report their results on proprietary datasets (currently not available) and the re-implementation of their method is quite complex and the risk of an inexact implementation is high.

5.1. Dataset and evaluation criterion

For the first round of experiments we focus our evaluation on public and commonly used datasets (see Fig. 3). In particular, Wall and Graffiti datasets [3,4,25,29] are typical benchmarks to evaluate keypoint robustness against viewpoint changes while Book dataset [26,27] is employed to analyze keypoint robustness against lighting changes.

For the second round of experiments we use the recently introduced BigBIRD [34] dataset, which contains a larger amount of objects acquired under different viewpoints and lighting conditions.

In the following we briefly describe these datasets and the associated ground truth information.

Wall and Graffiti datasets (see Fig. 3) consist of six images, where the first one is acquired under standard conditions and the remaining five images are acquired under different viewpoints (with increasing variation). According to other works, the first image of each dataset has been used as reference image whereas the remaining five images have been considered for matching. For each transformed image the ground truth homography matrix [25] corresponding to the transformation with respect to the reference image is given, thus allowing the ground truth keypoint correspondence to be easily computed. When working with Graffiti and Wall datasets, for each reference image we generated 80 artificial transformations to be used for the training phase: the variations considered are random homographic transformations within predefined parameter ranges.

Book dataset has been selected from Phos database [30] and contains 45 images taken under different conditions of natural light (overexposure, underexposure, directional).

The first image acquired in standard conditions (normal exposure) has been considered as reference image. Unlike for Wall and Graffiti, here generating artificial light changes (including shadows) for training is complex and could lead to produce unrealistic variations. Therefore the 44 images have been split in two sets: the first set (29 images) has been used for training, while the remaining 15 images for testing.

BigBIRD dataset [34] consists of 125 objects each of them acquired under different poses by varying the camera angle and the rotation plan. We selected a total of 100 objects by leaving out objects with very poor or no texture information; for each object we selected one frontal reference image plus five rotated images with increasing rotation angle. As described for Wall and Graffiti, also for each BigBIRD object we exploit ground truth homography matrices to map points between reference and rotated images. Moreover, for our training we use the same number of artificial transformations (80).

For each dataset, a training phase has been carried out to detect and rank keypoints for each reference image I^{ref} and the corresponding transformations. According to the notation introduced in Section 4, the m most salient keypoints are denoted by $K_d^m(I^{\text{ref}})$.

Then for each test image I^{test} we extract all keypoints $K_d(I^{\text{test}})$ and we evaluate the accuracy of our detection/matching approach based on average Hamming distance and recall.

The average Hamming distance is defined as:

$$H^{\text{avg}}(I^{\text{test}}) = \frac{1}{m} \sum_{x_i \in K_d^m(I^{\text{ref}})} H(\mathbf{b}_i, \tilde{\mathbf{b}}_i) \tag{10}$$

where $\tilde{\boldsymbol{x}}_i = \text{Transf}_{\text{test}}(\boldsymbol{x}_i)$ is the position corresponding to $\boldsymbol{x}_i$ in the test image according to the known transformation binding I^{test} to I^{ref} and $\tilde{\mathbf{b}}_i = \text{descr}(\tilde{\boldsymbol{x}}_i, P(\tilde{\boldsymbol{x}}_i))$.

The recall metrics is computed pairing $K_d^m(I^{\text{ref}})$ and $K_d(I^{\text{test}})$ by nearest-neighbor Hamming distance and by evaluating the amount of correct pairing. We associate each $\boldsymbol{x}_i \in K_d^m(I^{\text{ref}})$ to the keypoint $\boldsymbol{y}_j \in K_d(I^{\text{test}})$ such that:

$$H(\mathbf{b}_i, \hat{\mathbf{b}}_j) = \min_{y_k \in K_d(I^{\text{test}})} H(\mathbf{b}_i, \hat{\mathbf{b}}_k) \tag{11}$$

being $\hat{\mathbf{b}}_j = \text{descr}(\boldsymbol{y}_j, P(\boldsymbol{y}_j))$ the BRIEF descriptor of $\boldsymbol{y}_j$. Since the mapping function between reference and test images is known we can distinguish between correct and false matches. In particular, according to Mikolajczyk and Schmid [3], a match is considered correct if the spatial overlap between the regions covered by the two keypoint

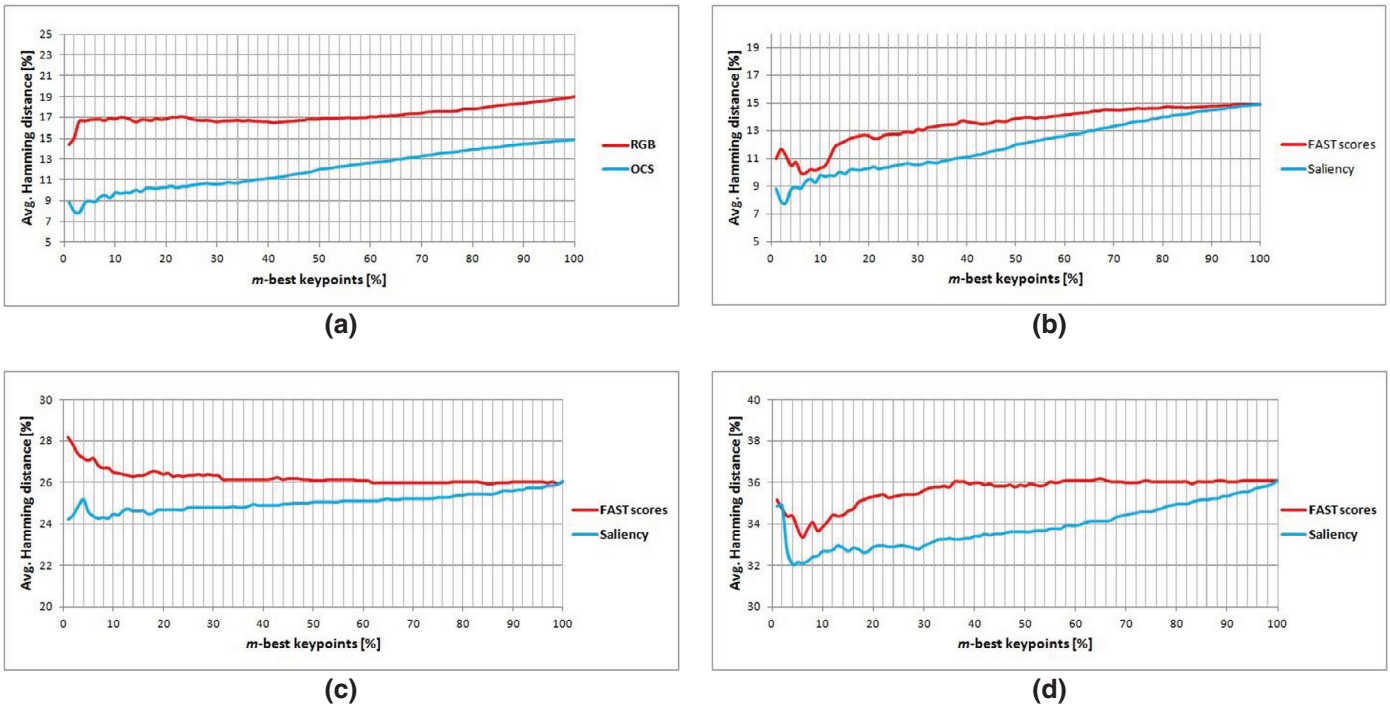

Fig. 4. Average percentage Hamming distance by varying the percentage of keypoints; (a) the graph refers to average values over all reference-test pairs of Book dataset when using RGB and OCS space; (b)–(d) the graphs refer to average values over all reference-test pairs of Book, Wall and Graffiti datasets when using our saliency-based ranking and FAST score ranking. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

descriptors is larger than a given threshold ε_s. Therefore, recall is defined as:

$$\text{recall} = \frac{\#\text{ correct matches}}{m} \tag{12}$$

It is worth noting that the two metrics used, even if related, consider different aspects. In fact, when computing the average Hamming distance we do not extract keypoint by FAST approach on I^{test} but we compute the descriptors at the positions corresponding to the projection of I^{ref} keypoints; so this metrics highlights the descriptor matching but discounts potential errors due to keypoint detection. On the contrary, when estimating recall, we compute correspondences by nearest neighbor between the two sets of keypoints (both detected by FAST) and this takes into account false detection and false pairing of keypoints.

Tests have been repeated for different values of m: from 1% to 100% of the total number of keypoints with step 1%. This allows to evaluate the effect of selecting different amounts of keypoints.

5.2. Saliency evaluation: OCS vs RGB

To evaluate the pros/cons of OCS, we used the Book dataset. Fig. 4 (a) shows the Hamming distance reported as percentage with respect to the descriptor length. For both RGB and OCS the increasing trend of the curves proves that the most salient keypoints are the most stable with respect to image variations. As expected, results confirm that by using OCS instead of RGB, BRIEF descriptors result more similar and therefore more invariant to light changes. The good performance obtained by using OCS induced us to use such color space for the rest of the experiments.

5.3. Saliency evaluation: Effectiveness of keypoint selection

In this set of experiments we compare our saliency-based ranking with respect to FAST score-based ranking (according to the notation used, FAST scores are the s_i values introduced in Section 4). Results obtained on Wall, Graffiti and Book datasets, reported in Fig. 4 (b)–(d) respectively, show an increasing trend of the Hamming distances thus proving that the most salient keypoints (those in the first positions of the ranking) are more likely to match under the image variations. Ranking keypoints according to their FAST scores leads to a somewhat analogous trend but in this case the curve is more flat and, especially for small values of m, the reduction in the average Hamming distance is less relevant. Of course, when 100% keypoints are used, ranking is irrelevant and the two curves converge. The preliminary results on Wall, Graffiti and Book allowed us to define optimal values for the basic parameters: $\omega_R = 1$, $\omega_D = 1$, and $\omega_F = 2$. For the rest of experiments we keep these values fixed.

The same type of experiment has been repeated for 100 BigBIRD objects (see Fig. 6). Here too we note that our ranking is more effective than FAST-score, and we can observe (for our approach) an increased trend of the distances after an initial decrease. It is worth noting that BigBIRD objects typically consist of small objects (i.e. canned food) with respect to Wall, Graffiti and Book images and the number of relevant keypoint is much smaller.

Fig. 5 shows the recall values on Book, Wall and Graffiti datasets. Here the decreasing trend proves that the most salient descriptors are detectable with more stability thus leading to a lower number of false matches. For example, for Wall dataset, working with the 10%-most salient keypoints results in a recall of about 45%, while using the whole set of keypoint reduces the recall to about 20%. It is worth noting that the recall values are quite different for the three datasets and in particular are higher for Book and lower for Graffiti. This is due to the relative difficulty of the dataset. In particular, Graffiti confirms to be challenging for BRIEF descriptors, leading to low average *recall.* In [4] a higher recall is reported on this dataset. However, the comparison of our result with [4] is misleading, since in [4] the recall value has been computed by pairing keypoints according to ground truth data.

5.4. Saliency evaluation: Comparison with the state-of-the-art

In this set of experiments we compare our saliency-based ranking against the FAST score-based ranking and against the ranking method

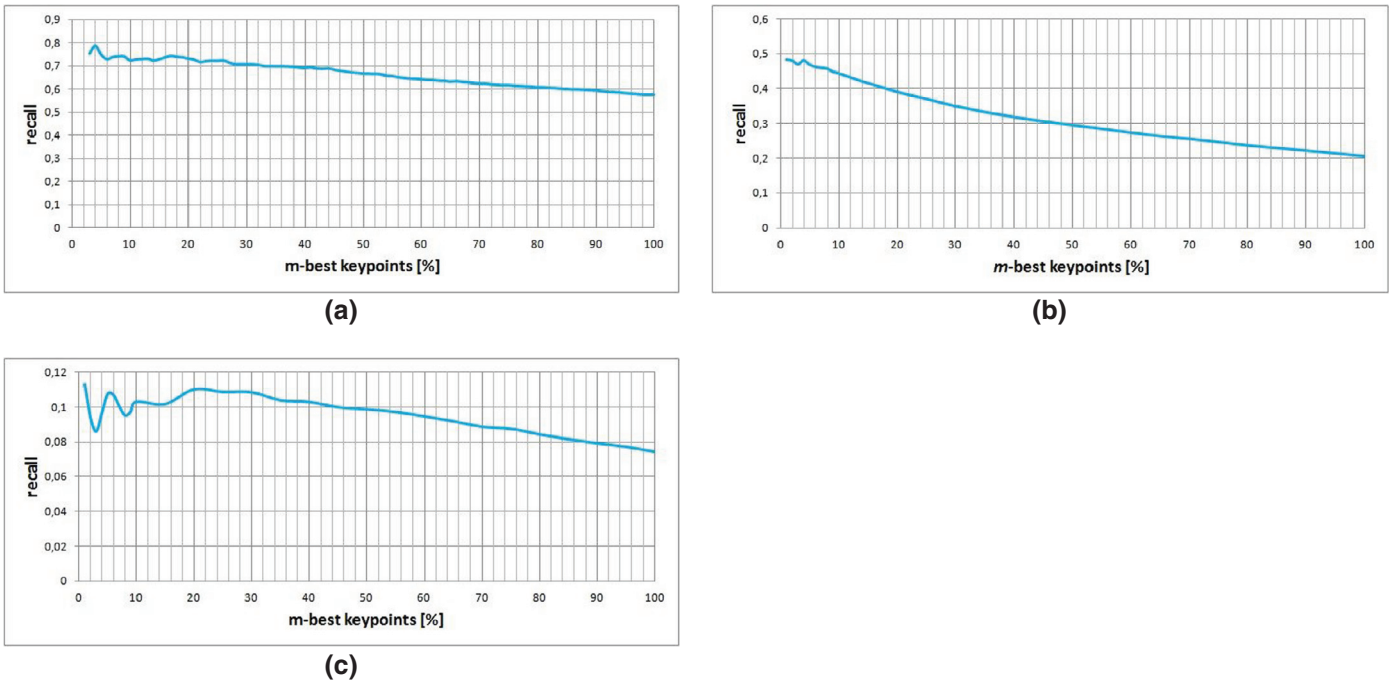

Fig. 5. Recall value averaged over all reference-test pairs for Book (a), Wall (b), and Graffiti (c) datasets, plotted as a function of keypoints percentage.

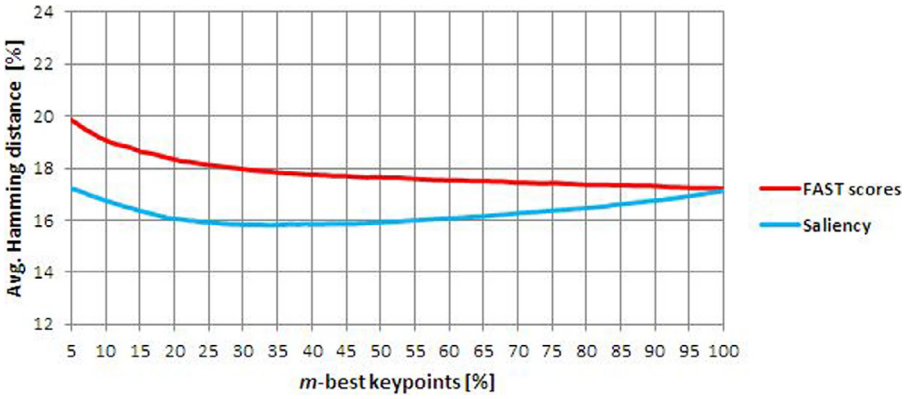

Fig. 6. Average percentage Hamming distance by varying the percentage of keypoints on 100 BigBIRD objects: the graph refers to average values over all reference-test pairs when using our saliency-based ranking and FAST score ranking. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

proposed in [33]. Hartmann et al. [33] train a random forest classifier able to return a list of keypoints ordered by matchability score of SIFT descriptors. We use such classifier to produce a list of keypoints for each BigBIRD object. Then, we compare the average Hamming distances of BRIEF descriptors computed for the matchability-based ordered list of keypoints with our saliency-based and FAST score-based ordered lists. To avoid implementation differences for [33] we used the code kindly made available by the authors. Results are reported in Fig. 7 and show that keypoints ranking according to our saliency yields a more stable set of descriptors.

As the reader can argue, curves in Fig. 7 are not convergent and trends of light blue and red curves are not the same as Fig. 6. This is due to different reasons: on one hand, [33] was originally designed to operate with SIFT detector thus leading to a different set of keypoints with respect to FAST. On the other hand in this experiment we were forced to truncate the saliency-based and FAST score-based lists in order to make them numerically comparable with the matchability-based list of [33], thus eliminating the tails of the light blue and red curves.

5.5. Saliency evaluation: Average processing time

In this section we briefly illustrate time performance of the training stage of our approach. For efficiency on test images please refer to Section 6. To carry out the training phase, the following operations are required: FAST detection, BRIEF computation and evaluation of detectability, repeatability and distinctiveness on 80 different transformed images. Considering that each BigBIRD object has an average of ~750 detected FAST keypoints, the required average processing time on a workstation Intel i7-2720QM 2.20 GHz with 8 GB of RAM to complete the training phase of a single object is ~23 s.

6. A real application: Pose estimation in Augmented Reality

In this section we apply the proposed approach to real-time pose estimation for vision-based Augmented Reality (AR) [13]. We are interested in detecting "natural" object markers in order to estimate the pose of the object with respect to the camera and superimpose information of interest to the captured image. As case of study we

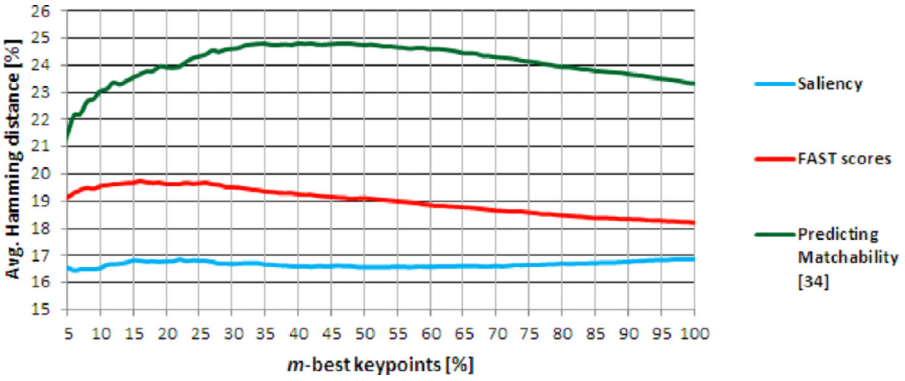

Fig. 7. Comparison with state-of-the-art: the graph refers to Hamming average distance values over all reference-test pairs of BigBIRD objects when using our saliency-based ranking, FAST score ranking and the very recently proposed ranking based on "matchability" properties of keypoints. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

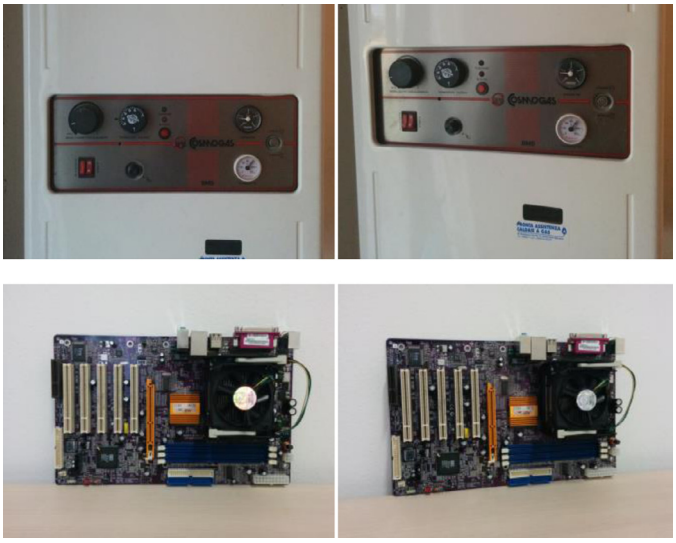

Fig. 8. Reference images (left column) and a test images (right column) of a water heater and a PC motherboard.

consider the AR-based maintenance task of two objects with substantial differences in local appearances and textures: a water heater and a PC motherboard (see Fig. 8).

In these applications the user is expected to interact with a mobile device (e.g., tablet or smartphone) whose camera takes live pictures of the object and useful pictorial information is superimposed (at the proper location) to guide maintenance. Since the viewpoint changes are moderate, our approach finds here an ideal application. In particular, the object of interest (i.e., the front panel of a water heater or a motherboard) can be considered as a simple planar object and therefore estimating its pose is equivalent to compute a homographic transformation. Given a set of keypoint correspondences we can extract the homography matrix through the RANSAC algorithm [28]. We compare three cases:

A. all FAST keypoints are considered;
B. only 15% *m*-best keypoints ranked according to FAST score are considered;
C. only 15% *m*-best keypoints ranked according to our saliency-based approach are considered.

In all the above cases, (initial) keypoint correspondences are found by nearest neighbor. In the C case a single reference image is used to generate 80 synthetic viewpoint changes and produce keypoints ranking. Figs. 10 and 11 compare the results on different test images.

Although RANSAC is somewhat robust against outlier, the advantages of using only robust keypoints is here evident in terms of precision of the recovered viewpoint transformation. We also note how our saliency-based ranking leads to consolidate a higher number of inliers and therefore a better viewpoint estimation with respect to the FAST-score based selection.

We repeated the previous test by random selecting 100 frames from a video taken by moving a tablet in front of the objects. The ground truth pose (for evaluation purposes) has been obtained by manually labeling the four panel corners. Then, automatic pose recovery has been carried out by approaches A, B and C, respectively. The resulting accuracy, here quantified in terms of average corner

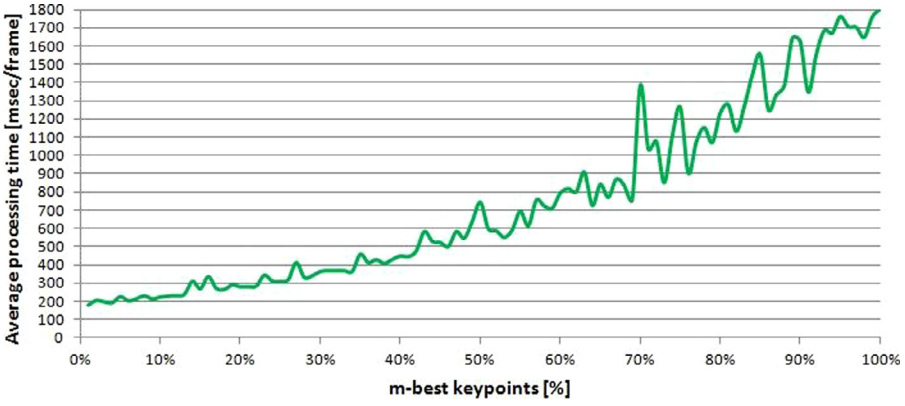

Fig. 9. Average processing time (milliseconds) required for frame analysis and pose estimation.

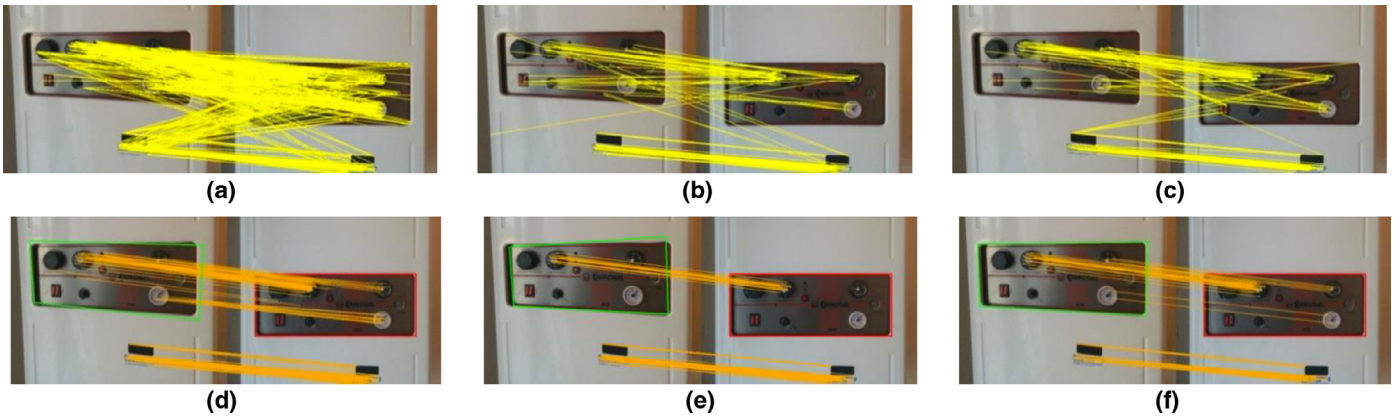

Fig. 10. Water heater transformation recovery through RANSAC algorithm by taking as input: (d) all FAST keypoints; (e) 15% *m*-best keypoints ranked according to FAST score, (f) 15% *m*-best keypoints ranked according to our saliency-based approach. Yellow segments (a), (b) and (c) denote initial keypoint pairing and orange segments (d), (e) and (f) final RANSAC inliers; the green rectangle denotes the homographic transformation inferred by RANSAC. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

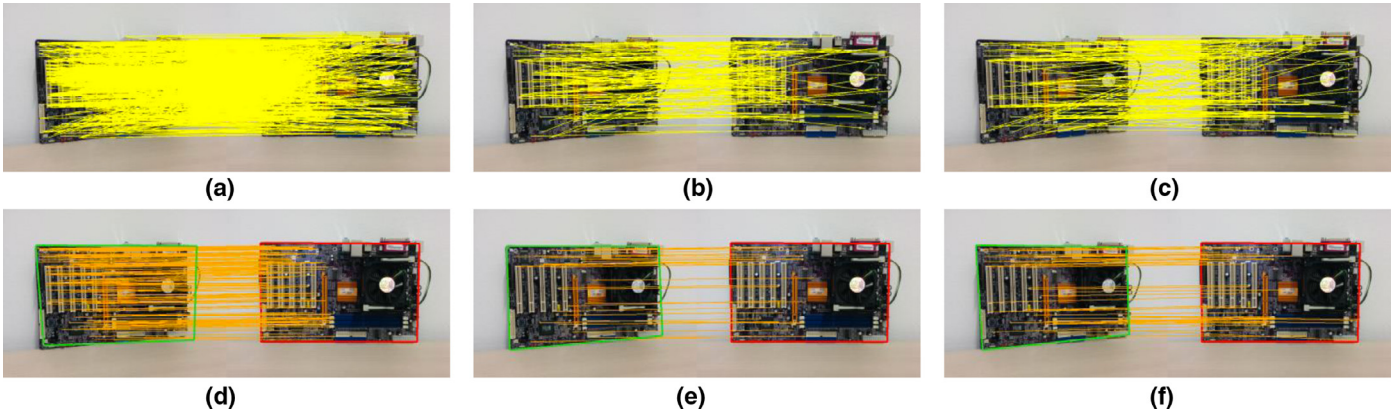

Fig. 11. PC motherboard transformation recovery through RANSAC algorithm by taking as input: (d) all FAST keypoints; (e) 15% *m*-best keypoints ranked according to FAST score, (f) 15% *m*-best keypoints ranked according to our saliency-based approach. Yellow segments (a), (b) and (c) denote initial keypoint pairing and orange segments (d), (e) and (f) final RANSAC inliers; the green rectangle denotes the homographic transformation inferred by RANSAC. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

distance with respect to the ground truth position, increases when using only the most salient keypoints. In particular, with respect to case A: the average distance is 32% lower in B and 56% lower in C for the water heater and 7% lower in B and 50% lower in C for the motherboard.

In terms of computation, the processing time needed to estimate pose for a single frame can be split in: (i) FAST detection, (ii) RGB to OCS conversion, (iii) BRIEF descriptors computation, (iv) Keypoints matching and (v) RANSAC homography estimation. The stages whose computing time depends on the number of keypoints are (iv) and (v). Fig. 9 shows the average processing time for a single frame analysis as function of the keypoints percentage. For this experiment we used a tablet device: Samsung ATIV Smart PC (Intel Atom Processor Z2760 1.5 GHz). Even if the code (written in C# for .Net) was not highly optimized, by selecting the 10%-best keypoints we can process about 5 frames per second. We are confident that with proper optimization we can significantly improve the frame rate. On the contrary, for this application, if all keypoints are used the efficiency would be about one order of magnitude worse.

7. Conclusions and future work

In this work we propose a method to quantitatively evaluate saliency of keypoints with the aim of selecting reliable keypoints for fast object detection and matching. Saliency is defined in terms of detectability, repeatability and distinctiveness. We prove that a simple training (often starting from a single image) is effective to select keypoints which are stable under moderate viewpoint and lighting variations. Experimental results show that our saliency-based ranking and selection of keypoints is effective in terms of both matching accuracy and processing speed thus making this approach feasible for real-time application such as Augmented Reality.

As a future work we plan to improve the proposed approach by allowing to train an object model starting from multiple (unregistered) images and to dynamically evolve the object model (by replacing the selected keypoints) after the initial training.

Furthermore, even if in this paper we mainly focus on object detection and pose estimation, we are confident that the keypoint selection we proposed can be useful for object recognition in many different applications.

References

[1] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[2] H. Bay, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Comput. Vis. Image Understand. 110 (3) (2008) 346–359.
[3] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 27 (10) (2005) 1615–1630.
[4] M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: Binary robust independent elementary features, in: European Conference on Computer Vision, Springer, Berlin Heidelberg, 2010, pp. 778–792.
[5] E. Rosten, R. Porter, T. Drummond, Faster and better: A machine learning approach to corner detection, IEEE Trans. Pattern Anal. Mach. Intell. 32 (1) (2010) 105–119.

[6] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2, IEEE, 2003 II-264.
[7] A.R. Pope, D.G. Lowe, Probabilistic models of appearance for 3-D object recognition, Int. J. Comput. Vis. 40 (2) (2000) 149–167.
[8] R. Sim, G. Dudek, Learning generative models of scene features, Int. J. Comput. Vis. 60 (1) (2004) 45–61.
[9] K. Ohba, K. Ikeuchi, Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects, IEEE Trans. Pattern Anal. Mach. Intell. 19 (9) (1997) 1043–1047.
[10] S. Agarwal, D. Roth, Learning a sparse representation for object detection, in: European Conference on Computer Vision, Springer, Berlin Heidelberg, 2002, pp. 113–127.
[11] M. Weber, M. Welling, P. Perona, Unsupervised Learning of Models for Recognition, Springer, Berlin Heidelberg, 2000, pp. 18–32.
[12] G. Carneiro, A.D. Jepson, The quantitative characterization of the distinctiveness and robustness of local image descriptors, Image Vis. Comput. 27 (8) (2009) 1143–1156.
[13] B. Furht, Handbook of Augmented Reality, vol. 71, Springer, New York, 2011.
[14] C. Harris, M. Stephens, A combined corner and edge detector, Alvey Vision Conference, vol. 15, Manchester, UK, 1988, p. 50.
[15] S.M. Smith, J.M. Brady, SUSAN - A new approach to low level image processing, Int. J. Comput. Vis. 23 (1) (1997) 45–78.
[16] J. Shi, C. Tomasi, Good features to track, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 1994, pp. 593–600.
[17] H.T. Comer, B.A. Draper, Keypoint stability prediction, Computer Vision Systems, Springer, Berlin Heidelberg, 2009, pp. 315–324.
[18] K. Mikolajczyk, C. Schmid, An affine invariant keypoint detector, in: European Conference on Computer Vision, Springer, Berlin Heidelberg, 2002, pp. 128–142.
[19] Y. Amit, D. Geman, A computational model for visual selection, Neural Comput. 11 (7) (1999) 1691–1715.
[20] R.C. Nelson, Memory-based recognition for 3-d objects, ARPA Image Understanding Workshop, Citeseer, 1996, pp. 1305–1310.
[21] G. Dorkó, C. Schmid, Selection of scale-invariant parts for object class recognition, in: IEEE International Conference on Computer Vision, IEEE, 2003, pp. 634–639.
[22] W. Zhang, J. Košecká, Hierarchical building recognition, Image Vis. Comput. 25 (5) (2007) 704–716.
[23] G. Carneiro, A.D. Jepson, The distinctiveness, detectability, and robustness of local image features, Comput. Vis. Pattern Recogn. 2 (2005) 296–301.
[24] K.E. Van De Sande, T. Gevers, C.G. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1582–1596.
[25] http://www.robots.ox.ac.uk/~vgg/research/affine/
[26] V. Vonikakis, D. Chrysostomou, R. Kouskouridas, A. Gasteratos, A biologically inspired scale-space for illumination invariant feature selection, Measure. Sci. Technol. 24 (7) (2013) 074024 (13pp).
[27] V. Vonikakis, D. Chrysostomou, R. Kouskouridas, A. Gasteratos, Improving the Robustness in Feature Detection by Local Contrast Enhancement, in: Proceedings of the IEEE International Conference on Imaging Systems and Techniques, IEEE, 2012, pp. 158–163.
[28] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
[29] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, L. Van Gool, A comparison of affine region detectors, Int. J. Comput. Vis. 65 (1–2) (2005) 43–72.
[30] http://utopia.duth.gr/~dchrisos/pubs/database2.html
[31] J.M. Geusebroek, R. Van den Boomgaard, A.W.M. Smeulders, H. Geerts, Color invariance, IEEE Trans. Pattern Anal. Mach. Intell. 23 (12) (2001) 1338–1350.
[32] J. Van De Weijer, T. Gevers, A.D. Bagdanov, Boosting color saliency in image feature detection, IEEE Trans. Pattern Anal. Mach. Intell. 28 (1) (2006) 150–155.
[33] W. Hartmann, M. Havlena, K. Schindler, Predicting matchability, in: Conference on Computer Vision and Pattern Recognition., IEEE, 2014.
[34] A. Singh, J. Sha, K.S. Narayan, T. Achim, P. Abbeel, BigBIRD: A large-scale 3D database of object instances, in: IEEE International Conference on Robotics and Automation, IEEE, 2014.