Comparison and Optimization of Image Descriptors for Real-Time Detection of Abandoned Objects

Florentin Kucharczak, Allan F. da Silva, Lucas A. Thomaz, Gustavo Carvalho,
Eduardo A. B. da Silva, Sergio L. Netto
PEE/COPPE, Federal University of Rio de Janeiro, RJ, Brazil.
{kucharczak.florentin, allan.freitas, lucas.thomaz, gustavo.carvalho, eduardo, sergioln}@smt.ufrj.br

Abstract - This paper presents a detailed study of four image descriptors (SURF, SIFT, BRISK and FREAK) in the context of real-time detection of abandoned objects using a moving camera. In this scenario, captured frames are compared to a reference video, and noticeable differences among the two videos are associated to an abandoned object. The image descriptors allow a simple and robust image representation, retaining most relevant features for proper registration and alignment, enabling a comparison between two video frames corresponding to the same scene. Performances of these fours schemes are assessed in terms of processing time and detection efficiency considering the OpenCV implementations of the methods. A modification is also considered for the image descriptors, which restricts the correspondences between the two images to the same representation scale. Experiments on three pairs of videos show the improvements achieved by the proposed system configuration.

Keywords: object detection, automatic surveillance, cluttered environment, moving camera, image descriptors, SIFT, SURF, FREAK, BRISK.

I. Introduction

Technology needs in terms of security, control and speed have increased strongly in the last few decades. In response to this challenge, many computer-vision solutions for production process control, object detection and automatic guidance systems have been developed. These allow, among others, risk reduction when operating in hazardous environments, decreased costs and an overall performance improvement. Among the many factors that contributed to this technology leap, we may cite the continued growth of computing and data-storage capabilities, as well as faster and more accurate algorithms for these specific purposes.

One of the main challenges in automatic video surveillance is the excess of data being processed, particularly when one considers the real-time aspect to the application at hand. This obstacle is usually overcome by using the so-called image descriptors that identify a limited number of keypoints within a video frame. Ideally, the set of keypoints constitutes a compact representation with the most significant information from the original image. These keypoints may then be employed, for instance, to determine a transformation between two corresponding frames in the target and reference videos, allowing proper video registration before the difference-detecting stage. Finding the optimal transformation between two images is usually a computationally intensive operation. Therefore, in order to perform real-time detection, an efficient algorithm for this step is paramount. This work investigates the performance of different image descriptors in the context of abandoned object detection using a moving camera. It explores the various parameters and specificities of these four descriptors, searching for the best compromise between detection accuracy and speed.

The main focus of this work is the comparison of four image descriptors found in the literature in the context of abandoned-object detection, namely: scale-invariant feature transform (SIFT, [1]); speed-up robust features (SURF, [2]); binary robust invariant-scale keypoints (BRISK, [3]); fast retina keypoint (FREAK, [4]). In addition, since the nature of the surveillance setup guarantees that the camera views the two scenes to be matched at the same scale, we investigate the use of scale restriction when matching the keypoints in order to reduce the number of false matches. To present the proposed contributions, this paper is organized as follows: Section II describes the object-detecting system where the image descriptors are employed. Section III summarizes the main characteristics of the four image descriptors here evaluated. Section IV details the comparative analysis of the image-descriptor performances in terms of computational cost and detection efficiency. Section V investigates the system performance when the proposed fixed-scale constraint is imposed to the image registration stage and Section VI concludes the paper emphasizing its main contributions.

II. Object-Detection System

The surveillance system employed in this work is described in [6]. The system uses a high-definition camera mounted on a robotic platform that performs a linear back-and-forth motion to increase the camera range. The video obtained in a first passage, after proper validation, is used as a reference to which all subsequent videos are compared in search of a new video event, such as the appearance of an abandoned object. In this scenario, the newly acquired (target) and reference videos can only be compared after precise time- and space-alignment procedures are performed.

The system employed here differs from others found in the literature [8]–[10] by not relying on any external trigger signal to enforce synchronization between the videos. Instead, points of interest (keypoints) are extracted from both videos and a search for correspondences is performed between adjacent frames for each video, allowing one to estimate the camera displacement up to a constant offset. Then, the temporal alignment between the two videos is estimated as the delay that maximizes the correlation between the two motion models.

Ideally, synchronism between videos should provide sufficient conditions to allow a sample-by-sample measurement of similarity among the target and reference videos. However, a few imperfections such as vibration or friction cause the camera motion to be different from footage to footage, thus requiring an additional spatial registration stage between the two videos. For this purpose, keypoint correspondences in synchronized frames are used to generate a geometric transformation (homography) that maps one camera view into the other. In this process, the random sample consensus (RANSAC) algorithm [9] is employed to eliminate correspondence outliers that may generate an improper geometric transformation. Due to the linear nature of movement (the camera moves along a straight rail), outlier removal is also performed by imposing a maximum 1° angle with the horizontal axis for all keypoint correspondences.

After the time- and space-alignment procedures, the target and reference videos are compared through the normalized cross correlation (NCC) function [9], and a threshold is used to generate a binary mask that indicates possible abandoned objects. However, as this calculation is independent of the intensity values the pixels of the original image, there may be a large number of false-positive regions. Thus, the NCC computation is restricted to regions where the absolute difference between the frames is above a certain threshold. To further reduce the effects of false positives the binary mask undergoes a temporal voting process, where each mask pixel must appear a minimum number of times in a given time interval.

III. Image Descriptors

An image descriptor represents relevant visual features of an image through a limited number of keypoints. Such features may be

used to compare images by finding corresponding points between them. Among the image descriptors available, the most used are SIFT [1], SURF [2], BRISK [3] and FREAK [4], which were devised with similar functionalities. Generally speaking, these descriptors work similarly by performing the four steps described below:

• **Scale-Space Representation**: Objects in real world have characteristics in various levels of details, such as contours at low level and texture at high level. An algorithm for automatic extraction of image features must obtain information about the different aspects of each object, since it does not have in general any prior knowledge about the level of detail that should be used to interpret a given image. Thus, most image descriptors use a scale-space framework that generates a one-parameter family of images derived from the original, with each member representing well a different detail level. This step ensures that objects with different sizes or levels of degradation can be recognized, as they present similar features at different scales.

• **Interest Point Localization**: In this stage salient points are extracted from the scale-space representation, generating a compact description of most important content features within the original image. The extraction is performed by detecting the most representative local maxima and minima in intensity. These will correspond to points with a consistent relative position, such that similar objects in different positions or sizes generate a similar set of salient points even when in different images.

• **Orientation Assignment**: This step estimates the orientation (direction of maximum luminosity variation) for each keypoint identified in the previous stage. This feature allows one to identify similar objects with different relative orientations in multiple images.

• **Keypoint Descriptor**: Finally, a descriptor is determined for each keypoint previously identified, providing a compact representation of the most important image characteristics around that point. Thereby, descriptors obtained from different images can be compared, allowing the identification of similar features, which are associated to point/region correspondences between the two images.

In the sequel we analyze the four descriptors used in this work according to the above steps.

A. SIFT

The SIFT [1], [14] algorithm was one of the pioneers of the scale-invariant image descriptors. It works through a 4-step feature extraction for building the image descriptor, and its main characteristics are:

• **Scale-space extrema detection**: The scale-space is generated by applying a sequence of difference-of-Gaussian filters through multiple scales, thus generating a sequence of filtered-and-downsampled versions of the original picture.

• **Keypoint localization**: The SIFT keypoints are obtained by searching the local maxima in a 9-point vicinity in the scale space, excluding local maxima along image edges in order to avoid keypoint mismatching.

• **Orientation assignment**: The orientation is assigned based on a histogram of oriented gradients computed around each keypoint using 36 bins representing the 360-degree scale.

• **Keypoint descriptor**: The SIFT descriptor uses 4 orientation histograms built over a 16 x 16 pixel area around the keypoint.

B. SURF

The SURF [2] algorithm is a widely-used scale- and rotation-invariant descriptor which aims to work faster than SIFT while being more reliable. It is similar to SIFT, with the main differences as presented below:

• **Scale-space representation**: The SURF scale-space uses a bank of pre-designed filters that can be applied in parallel, as opposed to the cascade of Gaussian filters employed by the SIFT algorithm.

• **Interest point localization**: The maxima localization is done with a speeded-up method called 'fast-Hessian', which uses an algorithm know as integral Image to calculate the Hessian matrix, whose determinant will characterize a local maximum.

• **Interest-point description and matching**: The keypoint orientation is estimated with a Haar wavelet transform in both x and y directions followed by a Gaussian (interpolation) filter centered at the keypoint.

• **Keypoint descriptor**: The descriptor is obtained by summing the wavelet coefficients taken in 4 directions around the keypoint.

C. BRISK

The BRISK [3] algorithm, as the name suggests, is a binary image descriptor that was designed to be as robust as SURF, but using much less computational power. The main features of the method are:

• **Scale-space representation**: The scale-space representation is obtained by downsampling the original image, and organizing the resulting images into octaves and intra-octaves.

• **Keypoint detection**: The keypoint detection metric for the BRISK algorithm is the same as the one from FAST (Features-from-Accelerated-Segment-Test) [16] algorithm, which detects local maxima or minima as points that are brighter or darker than an arc of contiguous pixels around it.

• **Keypoint description**: The BRISK descriptor consists of a 512-bit array obtained by comparing point-to-point the intensity of samples taken along a rotated circular pattern around the keypoint. If the intensity of the first point is higher than the one of the second, then the corresponding bit is set to 1; otherwise, it is set to 0.

D. FREAK

The FREAK [4] algorithm is a binary image descriptor inspired in the human visual system. Its main properties are described below:

• **Scale-space representation and keypoint detection**: The method essentially uses the same keypoint-detection scheme as the BRISK algorithm.

• **Keypoint descriptor**: The most significant FREAK feature is the retina-based sampling pattern, which is much denser around the keypoint as opposed to the uniform BRISK pattern. The FREAK descriptor is then built through a 1-bit difference-of-Gaussian method applied to the samples around the keypoint.

E. General Comparison

Among the 4 descriptors presented above the SIFT was the first one to be developed, quickly becoming the reference method to which most subsequent proposals were compared. The SURF algorithm was later introduced in an attempt to speed up the SIFT processing while sustaining (and sometimes improving upon) the SIFT's capability for providing a compact image description. The BRISK algorithm is a more recent method which uses a binary descriptor, thus saving both storage space and time taken for a proper descriptor match. The FREAK algorithm works as an evolution of BRISK, by providing an even shorter binary descriptor, also reducing memory space and simplifying the comparison stage, due to its human-retina-inspired sample pattern around the keypoint.

For our object-detection application, the most important image-descriptor characteristics are the processing speed (due to the real-time nature of the system) and the final detection robustness. Therefore, the performances of the 4 image descriptors above are assessed according to speed and robustness in the subsequent sections of this paper.

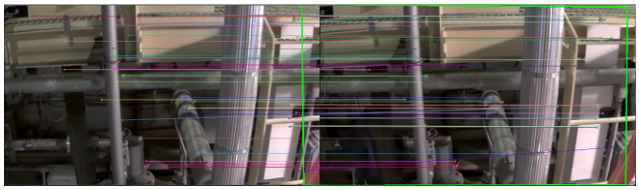
IV. Performance Assessment of Image Descriptors

The original detection system, described in [6], employed both the SIFT and SURF methods with the default setup given by the OpenCV library. Such configurations, however, were shown to be too complex computationally for the system to operate in real time. The main objective of this paper is to explore the various parameters and specificities of the four methods presented in Section III in

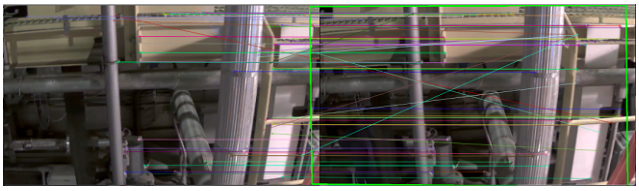
order to improve the compromise between detection accuracy and computational speed.

In an initial experiment, the 4 image descriptors were tested on a short 500-frame target-reference video pair, allowing a general comparison with respect to the number and accuracy of keypoint matches, as depicted in Fig. 1. From this figure, one readily notices that the SURF and BRISK algorithms generated much less improper (inclined) keypoint correspondences than the SIFT and FREAK algorithms, although all 4 methods yielded a significant amount of proper (horizontal) keypoint matches. A similar result was observed for the vast majority of frame pairs of the target and reference videos employed here. In practice, the detection system can remove all false matches by discarding the correspondences with an angle larger than 1 degree, as the camera movement is considered to be strictly horizontal.

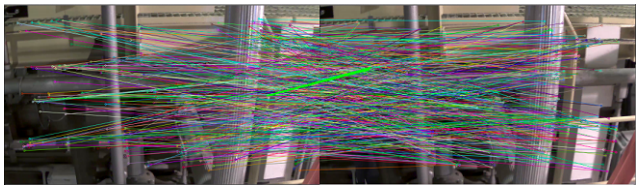
(a) SIFT



(b) SURF



(c) BRISK



(d) FREAK

Figure 1: Comparison of keypoint matching with SIFT, SURF, BRISK and FREAK descriptors.

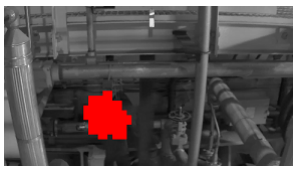
A second experiment considered one complete target-reference pair of videos, with the target video showcasing one abandoned ('coat') object. The initial performance results were obtained using the default parameters given by the OpenCV manual for the 4 descriptors. In all cases the processing time did not allow real-time detection, and some parameter adjustment was performed to speed up the descriptors while sustaining the system's ability to detected the object of interest. Table I summarizes the performances of the 4 descriptors with respect to the number of proper (horizontal) keypoint matches and processing time [1], which takes into account the detection of keypoints, descriptors, matching descriptor vectors and eventual corrections as that of discarding the correspondences with an angle larger than 1 degree. The table includes results for the SIFT and SURF setups employed in the system from [6] along with the ones from the 4 modified descriptors proposed in this paper. From these results one can infer that, for the same overall object detection performance (as validated in 2 other 1-object target videos), the original system was able to detect more keypoints at the cost of a higher processing time. After the fine tunning of the parameters from the 4 keypoint detectors, we observe that the keypoint correspondence and processing time can be significantly reduced, without any negative impact on the system's performance. This is particularly true in the case of the FREAK algorithm which was able to reduce the processing time to 1/3 of its original value.

It is important to notice that all these results were obtained using the implementations of the keypoint detection methods from OpenCV 2.4.8. This version contains a series of improvements in the SIFT algorithm to speed it up [7]. These improvements bridged the gap betweeen the processing times from SIFT and SURF that is in general reported in the literature. Suprisingly, the BRISK algorithm showed to run more slowly than usually reported in the literature. This can also be explained by its particular implementation in OpenCV 2.4.8, not necessarily meaning it is slower than the other methods. Some tests were made aiming to speed-up the BRISK algorithm by using different parameter settings. However, one could just obtain gains in speed of about 3%, but at the cost of the generation of only a prohibitively low number of keypoints.

Table I: Average performance results of Image Descriptors per frame.

	Keypoint pairs	**Processing Time (ms)**
SIFT	187,18	276,0
SURF	181,42	246,1
SIFT Optimized	170,56	168,5
SURF Optimized	80,43	99,5
BRISK	82,9	298,9
FREAK	35,03	72,8

A subjective evaluation of the overall detection system with each of the 4 descriptors is illustrated in Figure 2 for a single frame of the 'coat' target video. In this case, the marked region corresponds to where the algorithm detects the abandoned object. Thus, good detection implies a uniform marked region, centered on and covering most of the object, without extending beyond its borders. One can note that this is achieved for all 4 descriptors.



(a) SURF.

(b) SIFT.

(c) BRISK.

(d) FREAK.

Figure 2: Detection results using the 4 different descriptors in a single frame of the 'coat' target video.

V. Detection Improvement by a Scale Restriction

As mentioned in Section II, the object-detection system compares the target and reference videos in search for distinct regions which are associated to a possible abandoned object. An important characteristic of this setup is that all objects present in both videos should be at the same scale. Therefore, we propose to discard any keypoint match arising from different scales/octaves, thus increasing the system robustness (by reducing the chances of false correspondences) and processing speed (by reducing the search range for the keypoint match). We refer to it as the Scale-Dependend (SD) implementation.

[1]Considering the C++ descriptor implementations available in the OpenCV 2.4.8 library [5], using a 4-GB RAM MacBook Air, with the 10.9.2 IOS version and a 1.3-GHz Intel i5 processor.

Using this scale restriction, we also investigated the elimination of the 1-degree restriction to the keypoint correspondence inclination (referred to as the rotation invariant - RI - implementation). This is interesting because the robot vibrations can cause some camera oscillation above this threshold. Another desirable characteristic of getting rid of the angular restriction is that it also works with a more generic camera movement, other than the back-and-forth horizontal motion.

The proposed scale-dependent (SD) and rotation-invariant (RI) modifications were implemented in the FREAK detector, both separetely and together. The FREAK detector was chosen because in the tested OpenCV implemention of the methods it was the fastest and was able to generate good object detection. The performance of the resulting schemes was assessed with respect to the resulting number of keypoint matches and processing time, with a similar procedure as Table I, and is summarized in Table II. From this table, one concludes that neither modification affects the processing time in a significant manner. As expected, however, the RI-FREAK method generates a larger number of undesirable matches, leading to an efficiency drop on the detection system, as observed in Figure 3. In contrast, the SD-FREAK is able to avoid undesirable correspondences, thus reducing the total number of correspondences by about 20% while sustaining the detection performance. The RISD-FREAK algorithm synergizes the two modifications (namely, increased SD robustness by avoiding false correspondences and increased movement flexibility associated to the RI scheme) and yields a more robust system performance, as also illustrated in Figure 4.

Table II: Average performance results of FREAK methods per frame.

	Keypoint pairs	Processing Time (ms)
Original FREAK	35,03	72,8
RI FREAK	52,73	74,0
SD FREAK	29,74	72,9
RISD FREAK	40,37	73,1

(a) Original FREAK detector (b) RI FREAK detector

(c) SD FREAK detector (d) RISD FREAK detector

Figure 3: Detection of the different methods using FREAK.

(a) Original FREAK. (b) SD FREAK.

Figure 4: Detection of both FREAK methods for the same frame.

VI. Conclusions

A comparative analysis of some of the most widely used image descriptors (SIFT, SURF, BRISK and FREAK) is provided in the context of real-time object detection in a cluttered environment.

Preliminary subjective results based on the OpenCV 2.4.8 implemention indicated that a similar detection performance can be achieved by all 4 methods, with the FREAK descriptor presenting the fastest processing time. Two variations were then proposed to the FREAK algorithm in order to increase its robustness in the described system for abandoned object detection, by enforcing a scale-dependent matching system. The scale restriction had the added advantage of providing more flexibility on the camera movement, since it allowed the elimination of the 1-degree restriction on the matching stage.

When compared to the original FREAK method, the proposed SD-FREAK scheme obtains similar qualitative results on the object-detection application, at about 1/3 of the processing time for the original system without imposing any movement-restriction on the robotic platform.

These results allow real-time abandoned object detection, since they make it possible to process about 15 frames per second while performing keypoint detection, a sufficient frame rate for the proposed detection application.

VII. Acknowledgements

This work was developed with the support of Statoil Brazil, Petrobras, ANP, CAPES and FAPERJ.

References

[1] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *Int. Jrnl. Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[2] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, T., "SURF: Speeded-up robust features," *Computer Vision Image Understanding*, vol. 110 no. 3 pp. 346–359, 2008.

[3] Leutenegger, S., Chli, M., and Siegwart, R., "BRISK: Binary robust invariant scalable keypoints," In: *Proc. Int. Conf. Computer Vision*, pp. 2548–2555, 2011.

[4] Alahi, A., Ortiz, R., and Vandergheynst, P., "FREAK: Fast retina keypoint," In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012.

[5] Laganière, R., *OpenCV 2 Computer Vision Application Programming Cookbook*, PacktPuv, Birmingham, UK, 2011.

[6] Carvalho, G., de Oliveira, J. F. L., et al., "Um sistema de monitoramento para detecção de objetos em tempo real empregando câmera em movimento," In: *Proc. Simp. Brasileiro de Telecomunicações*, Fortaleza, Brazil, Sept. 2013.

[7] OpenCV Change Logs, In: *http://code.opencv.org/projects/opencv/wiki/ChangeLog*. Last accessed in: May 28, 2014.

[8] Zhou, D., Wang, L., et al., "Detection of moving targets with a moving camera," In: *Proc. IEEE Int. Conf. Robotics and Biomimetics*, pp. 677–681, 2009.

[9] Kong, H., Aaudibert, J.-Y., and Ponce, J., "Detecting abandoned objects with a moving camera," *IEEE Trans. Image Processing*, vol. 19, no. 8, pp. 2201–2210, Aug. 2010.

[10] Kundu, A., Jawahar, C. V., and Krishna, K. M., "Realtime moving object detection from a freely moving monocular camera," In: *IEEE Int. Conf. Robotics and Biomimetics*, pp. 1635–1640, 2010.

[11] Lipton, A. J., Fujiyoshi, H., and Patil, R. S., "Moving target classification and tracking from real-time video," In: *Proc. IEEE Workshop Applications of Computer Vision*, pp. 8–14, 1998.

[12] Mikolajzyk, K., and Schmid, C., "A performance evaluation of local descriptors," In: *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[13] Aarthy, V., Mythili, R., and Venkatalakshimi, S., "Using SIFT algorithm identifying an abandoned object by moving camera," *Int. Jrnl. Engineering Research and Applications*, vol. 2, no. 6, pp. 347–353, 2012.

[14] Lowe, D., "Object recognition from local scale-invariant features," *Int. Jrnl. Computer Vision*, vol. 2 pp. 1150–1157, 2004.

[15] Viola, P.A., and Jones, M. J., "Rapid object detection using a boosted cascade of simple features," *Computer Vision and Pattern Recognition*, vol. 110, no. 3 pp. 511–518, 2001.

[16] Rosten, E., and Drummond, E., "Machine learning for highspeed corner detection," In: *Proc. European Conf. Computer Vision*, 2001.

[17] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., "ORB: An efficient alternative to SIFT or SURF," In: *Proc. IEEE Int. Conf. Computer Vision*, pp. 2564–2571, 2011.