Journal of Intelligent & Robotic Systems (2018) 92:33–63
https://doi.org/10.1007/s10846-017-0762-8

Performance Evaluation of Feature Detectors and Descriptors Beyond the Visible

Tarek Mouats[1] · Nabil Aouf[1] · David Nam[1] · Stephen Vidas[2]

Received: 28 June 2017 / Accepted: 12 December 2017 / Published online: 8 February 2018

Abstract
Feature detection and description algorithms represent an important milestone in most computer vision applications. They have been examined from various perspectives during the last decade. However, most studies focused on their performance when used on visible band imagery. This modality suffers considerably in poor lighting conditions and notably during night-time. Infrared cameras, which noticed a considerable proliferation in recent years, offer a viable alternative in such conditions. Understanding how the building blocks of computer vision applications behave in this modality would help the community accommodating them. For this reason, we carried out a performance analysis of the most commonly used feature detectors and descriptors beyond the visible. A dataset accounting for the various challenges on these algorithms has been generated. In addition, challenges inherent to the thermal modality have been considered, notably the non-uniformity noise. A comprehensive quantitative investigation into the performance of feature detectors and descriptors is therefore presented. This study would serve to filling the gap in the literature as most analyzes have been based on visible band imagery.

Keywords Performance evaluation · Feature detectors · Feature descriptors · Thermal imagery · Infrared imagery

1 Introduction

Recent advances in infrared technology meant that smaller, relatively cheaper yet reliable thermal cameras are available. Their domains of application witnessed a significant shift within the research community. Infrared cameras have applications in motion detection [37], more specifically thermal cameras are being used in a variety of applications such as driver assistance systems, object recognition and image registration to name a few [12, 36, 38, 39]. The most appealing property of thermal cameras is the imagery itself. Indeed, in contrast to visible cameras, they capture temperature variations within the imaged scene. One direct implication is that thermal cameras do not exhibit the same problems as visible cameras (e.g. illumination changes). Therefore, infrared cameras may replace visible cameras in situations where these are disadvantaged (e.g. poor visibility conditions, night time, . . .). However, using image processing tools specifically designed for visible band imagery without adaptation may prove unsuccessful. In this context, we undertook a performance analysis of some computer vision algorithms and investigated how they behave when used with thermal images. Similar evaluations have been previously carried out in the literature for similar algorithms with visible band image sequences [17, 18]. The findings of these surveys cannot be generalised to other imaging modalities such as the thermal imagery. The main reason is the relatively large difference between the different modalities in terms of image content. The most reported shortcomings of infrared imagery are low image resolution and relatively low signal-to-noise-ratios (SNRs). These may render some computer vision algorithms unusable. For instance, optimal settings of feature detection

✉ David Nam
d.nam@cranfield.ac.uk
Tarek Mouats
t.mouats@cranfield.ac.uk
Nabil Aouf
n.aouf@cranfield.ac.uk
Stephen Vidas
svidas@ntu.edu.sg

[1] Defence Academy of the United Kingdom, Centre for Electronic Warfare, Information and Cyber, Cranfield University, Shrivenham, SN6 8LA, UK

[2] Intelligent Robotics Laboratory, Nanyang Technological University, Singapore, Singapore

pipelines were designed for visible-band images and their performance may degrade severely if they are not properly tuned and adapted for infrared modality [33]. Therefore, choosing the right image processing tools to operate in an exclusively infrared-based imagery environment can prove very challenging.

2 Related Works

Feature extraction and matching is an essential prerequisite for most vision-based navigation techniques. Various interest point extraction and description algorithms have been used for this purpose in visual odometry and simultaneous localization and mapping (SLAM). Generally, classical feature detectors are used in the case of standard visible band cameras ([9, 25, 29]). When it comes to thermal imagery, and due to the reasons discussed in Section 3, these algorithms may not prove satisfactory.

There are few works in the literature that study the performance of computer vision algorithms in the infrared spectrum. In contrast, far more efforts were dedicated to surveys in the visible band. Interest point detectors evaluations go back to a decade ago ([17, 18, 27]). Various investigations followed up such as the study of popular feature detectors and descriptors in the context of 3D inference from two views [6], visual tracking [8] and robot navigation [28]. However, the latter based their study on a single indoor trajectory of 20 m corresponding to a laboratory environment. It is not representative of outdoor conditions and therefore the findings cannot be generalised without caution. However, larger data-sets such as, [5] (covering a campus scenario over the course of 15 months) and [30] (obtained via a train) have been collected. Krajník et al. [10] proposed and tested a descriptor which copes well with seasonal changes. This was evaluated on outdoor images and compared against other descriptors, over the course of a year. Gauglitz et al. [8] generated a larger database encompassing various camera motion and image transforms for visual tracking. The only downside in their work is the data generation itself where they "... fabricated a precisely milled acrylic glass frame which holds the planar texture..." [8]. The dataset corresponds to video streams of images held in a glass frame and therefore reflects more a controlled environment than real-world scenes. With regard to the far-infrared modality, relatively fewer investigations were dedicated to this end. This disparity could be explained by the relatively recent introduction of thermal cameras (compared to visible-band sensors) in the field of computer vision. Vidas et al. [33] investigated the performance of feature detectors on thermal images using a similar protocol to the one introduced by [18]. However, the authors did not consider feature descriptors. Another study was carried out by [23] where only descriptors were examined and their performance compared to the visible band. The recent binary algorithms ORB [26], BRIEF [4], BRISK [13] and FREAK [2] were included in their study. The downside in their work is that no decisive conclusions were made regarding the performance of the descriptors. In addition, the image sets used in the evaluation, which come partly from our dataset (ref. [20]), did not include ground truth homographies. Therefore, objective quantitative comparisons were not conducted. Note also that the studied transformations are *artificially* added to the images after acquisition and do not represent real conditions e.g. image blur was generated using Gaussian filtering which does not necessarily correspond to motion blur inherent to camera motion. For these reasons, we carried out a series of experiments, adapting the protocols introduced by [17, 18] in order to study the behaviour of popular feature detection/description algorithms in the infrared modality. All the steps are detailed in Section 4 where the considered feature detection/description algorithms are enumerated and briefly explained.

The remainder of the paper is organised as follows. Section 3 introduces some challenges inherent to thermal imagery. The analysis framework is described in Section 4 where we present the analysed feature detection and description algorithms, the image sequences used for the comparison, the different metrics as well as the evaluation process. The experimental results are then presented and discussed in Section 5. Section 6 concludes the paper with discussions and highlights into future work.

3 Thermal Imagery Background and Challenges

Here, we provide some insights into the challenges inherent to thermal-infrared imagery. For more details, the reader is kindly referred to our previous work [20]. To make this paper self-contained, we included a brief summary.

Any object which has a temperature greater than absolute zero is a source of thermal radiation (even cold objects) as it emits heat in the infrared spectrum [21]. Infrared radiation lies between the visible and microwave bands of the electromagnetic spectrum. It is traditionally divided into three sub-bands: near, mid and far-infrared. However, only portions are of interest as most of the radiation is absorbed by water and carbon dioxide molecules present in the atmosphere. The sub-band of interest in this study is the far infrared or thermal (8-14μm). In the far infrared band, heat reflectance of observed objects hardly contributes to the captured image. Instead, it is composed mainly of the objects' thermal radiation. Thermal imagery exhibits a number of challenges compared to visible imagery [14]

namely: (1) high noise and low spatial resolution (2) history effects (3) image saturation.

More importantly, thermal images are known for their relatively low SNRs. This issue is inherent to thermal imagery and is largely due to non-uniformities in the sensor elements. Indeed, this problem is due to differences in the photo-response of each detector in the focal plane array [11]. If the imaged scene (environment) exhibits relatively low thermal variation, non-uniformities will be more dominant and the captured image will have lower SNR. The effects of non-uniformities can be corrected in two ways: calibration-based and scene-based techniques. The latter are out of the scope of this study and the reader is kindly referred to [11] for more information. Most thermal cameras have a Non-Uniformity Correction feature (NUC) also referred to as Flat Field Correction. This is usually done automatically by the IR camera, by periodically presenting a uniform temperature (a flat field) to every detector element. While imaging the flat field, the camera updates the correction coefficients, resulting in a more uniform array output. However, even immediately after such an NUC operation, noise may be present in the image.

Vidas et al. [33] studied the effect of non-uniformities on the repeatability property of feature extractors. A very slight decay in performance over several minutes was reported in the results for most detectors. However, as the time since the last NUC operation increases, the image intensities collectively drift from their true values. Therefore the accumulation of non-uniformity noise may have a more significant influence on later stages of computer vision algorithms such as feature description and matching. We investigate this aspect of the problem in our work as it represents an important milestone in vision-based navigation applications such as visual odometry or SLAM.

4 Performance Analysis Framework

In this Section, we detail the analysis framework used in our work. More specifically, we introduce the studied detection & description algorithms, the image sequences used for that purpose, the comparison metrics and the various evaluations that were carried out.

4.1 Studied Feature Detectors/Descriptors

4.1.1 Feature Detectors

Harris Harris built on the work of [19] to extract features from images. He introduced an autocorrelation function that determines variations in image intensities in a neighbourhood window Q of a point $p(u, v)$:

$$E(x, y) = \sum_{Q} w(u, v)\,[I(u + x, v + y) - I(u, v)]^2 \quad (1)$$

where (x, y) represent a given shift; $w(u, v)$ is the window patch at position (u, v); I is the intensity. Using Taylor approximation, Harris formulated the correlation function at a pixel as:

$$E(x, y) = \begin{bmatrix} x & y \end{bmatrix} M \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

where M is a 2×2 symmetric function given by

$$M = \begin{bmatrix} \sum_Q {I_u}^2 & \sum_Q I_u I_v \\ \sum_Q I_u I_v & \sum_Q {I_v}^2 \end{bmatrix} \quad (3)$$

I_u, I_v represent the spatial gradients of the image. This matrix characterises the structure of the gray levels around a given point. Let λ_1 and λ_2 be the eigenvalues of M. The local shape of the neighbourhood can be classified according to the values of λ_1 and λ_2:

- If both are small, the window neighbourhood is of approximately constant intensity;
- If $\lambda_1 > \lambda_2$, this gives an indication that there is an edge;
- If both are large, then this indicates a corner.

However, computing the eigenvalues is a computationally expensive task (at that time). Harris introduced an innovative concept to avoid it, named *the cornerness measure* (C_m) and given by

$$C_m = \det(M) - k\,.\,tr^2(M) = \lambda_1\lambda_2 - k\,(\lambda_1 + \lambda_2) \quad (4)$$

where k is chosen heuristically and takes values in the range [0.04-0.15]. Therefore, instead of computing the eigenvalues, it is sufficient to evaluate the determinant and trace of M to find corners.

Good Features To Track - GFTT building on Harris' work, Shi and Tomasi suggested that the cornerness scoring function can be computed as $\min(\lambda_1, \lambda_2)$ because under certain assumptions the detected corners are more stable for tracking [29]. If the minimum eigenvalue is larger than a predefined threshold, the corresponding candidate feature is considered a corner.

Difference of Gaussians - DoG detecting local extrema using differences of Gaussians was introduced by [15] in his SIFT algorithm. The descriptor part is described in Section 4.1.2. In order to provide invariance against scale changes, a pyramid of images is built through convolution with Gaussian kernels at different scales followed by a subtraction operation (Difference of Gaussian). A set of

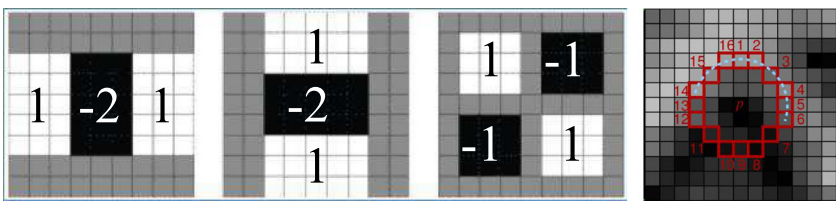
candidate features are extracted by looking at the maxima of the DoG images. Features are identified by comparing the value of a pixel to its 8 neighbours in the same scale and the 18 pixels in the two neighbouring scales (i.e. above and below). If the pixel value corresponds to a maxima in this neighbourhood then it is labelled as feature. Once candidate features are extracted, a clean-up has to be carried out: (i) low contrast features are removed by means of thresholding (ii) features lying on edges are eliminated by analysing the curvature of the area surrounding the keypoints using Harris' approach [9].

Fast-Hessian Fast-Hessian was introduced by [3] as the detection part of their SURF algorithm. Chronologically, Fast-Hessian was proposed after DoG as a faster, yet robust alternative. Features are extracted based on the Hessian operator which takes advantage of integral images to find local scale space extrema locations. The value of an integral image at a position $x = (u, v)$ is represented by the sum of all pixels in the image I contained within a rectangular window from the origin to the point x and given by $I_{\sum} = \sum_{i=0}^{i<u} \sum_{j=0}^{j<v} I(u, v)$. In order to avoid convolution with second order derivatives, Bay et al. suggested an approximation using simple box filters (Fig. 1a) computed on integral images at constant time cost. As for the DoG detector, a $3 \times 3 \times 3$-neighbourhood non-maximum suppression and sub-pixel refinement are carried out. In addition, candidate features with low scores s are discarded:

$$s(x, y, \sigma) = D_{xx}(\sigma).D_{yy}(\sigma) - \left(0.9D_{xy}(\sigma)\right)^2 \quad (5)$$

where $D_{xx}(\sigma)$, $D_{yy}(\sigma)$ and $D_{xy}(\sigma)$ are the convolution results of the filters illustrated in Fig. 1a.

Features from Accelerated Segment Test FAST is a known feature extractor in the computer vision community for its computational efficiency. To detect features in an image, FAST uses a circle of 16 pixels around the pixel of interest p numbered from 1 to 16 clockwise (Fig. 1b). Let I_p be the intensity of the pixel p and $thresh$ a threshold value. If a set of N contiguous pixels in the circle are all brighter than $I_p + thresh$ or all darker than $I_p - thresh$, then p is labelled corner. N is usually chosen to be equal to 12. A high speed test was introduced to accelerate the process by testing only 4 of the 16 surrounding pixels. However, it resulted in some



(a) (b)

Fig. 1 Fast-Hessian box filters and FAST test pattern. **a** box filters used by Fast-Hessian to approximate the Hessian matrix. Grey regions have a weight zero whereas the black regions' weights are annotated. **b** FAST test pattern for corner detection. Images adapted from [3, 25]

Fig. 2 Censure bi-level filters. From left to right: the ideal circle, octagon, hexagon and box. Image adapted from [1]

shortcomings that were tackled using machine learning principles as well as non-maximum suppression. The latter is necessary as multiple features adjacent to each other are returned by the detector.

Centre Surround Extrema Features - CenSurE as its name indicates, CenSurE is based on centre-surround filters to select feature extrema across scale and space. Simplified bi-level kernels are used to lower the computation times. In contrast to Lowe's approximation of the Laplacian (DoG), Agrawal et al. suggested a simpler alternative using the bi-level filters i.e. with values of -1 and 1. In order to reduce computational complexity of the symmetric circular filter, the authors proposed three approximations: octagon, hexagon and box filters (Fig. 2). As indicated above (Section 4.1.2), box filters can be efficiently computed using integral images. Agrawal et al. suggested the use of slanted integral images for octagon and hexagon filters. These can be decomposed into trapezoids and thus computed efficiently. The last steps in the selection requires a non-maximum suppression in a $3 \times 3 \times 3$-neighbourhood and the suppression of features lying along edges. This is in essence similar to the DoG and Fast-Hessian algorithms. The variants of CenSurE were compared by [1] where it was concluded that the octagon filter-based detector (CenSurE-OCT) provides the best compromise between repeatability and speed.

4.1.2 Feature Descriptors

Scale Invariant Feature Transform - SIFT Lowe proposed the following framework to compute a descriptor for the extracted features. In order to achieve rotation invariance, each feature f is assigned one or multiple orientations θ_f based on the local gradient information. The magnitude and direction of the gradient are calculated in the neighbourhood of the feature such that an orientation histogram with

36 bins is formed. The latter is weighted by a Gaussian window around f. The peak of the histogram corresponds to the orientation of the feature. In addition, if other peaks corresponding to at least 80% of the main peak exist, then other features are created with identical position and scale and different orientations. The feature scale σ_f and orientation θ_f are used in the descriptor computation steps as a local coordinate system. The descriptor is computed using the gradient magnitude and orientations in a 16 × 16 window around the feature (rotated according to θ_f). These are stacked in 8-bin histograms formed in 4 × 4 sub-regions and weighted by a Gaussian window. This yields a descriptor vector of 128 entries (default parameters).

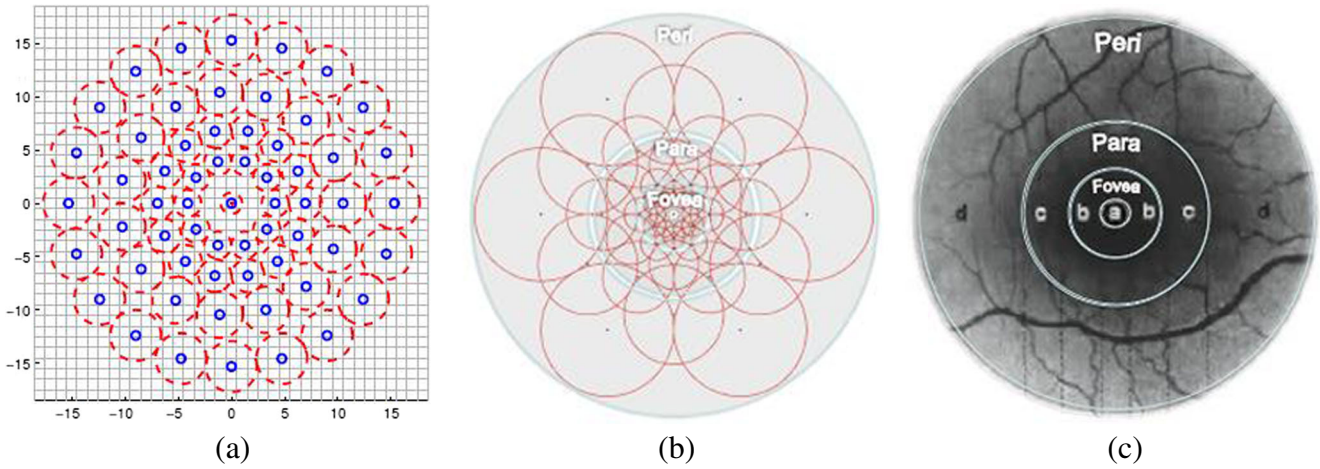
Speeded-Up Robust Features - SURF Similarly to SIFT, the first step in SURF consists in an orientation assignment. This is carried out by computing Gaussian weighted Haar-wavelet responses over a circular region of size 6 × scale around the keypoint (at the selected scale). Once an orientation is assigned, the descriptor is computed based on a square region (size 20 × scale) centred on the feature and oriented accordingly (similar to SIFT). This region is further divided into 4×4 sub-regions. Here again, horizontal and vertical Haar-wavelets responses are computed (and weighted with a Gaussian) at fixed sample points and summed up in each sub-region. Information about the polarity of intensity changes is also incorporated into the descriptor by means of the sum of absolute values of the horizontal and vertical responses. Hence, each sub-region has a descriptor vector containing 4 entries yielding a 64-element SURF descriptor (65 if the sign of the Laplacian is included).

Oriented FAST and Rotated BRIEF - ORB similarly to SIFT and SURF, ORB is a feature detector and descriptor algorithm. The main difference is that it belongs to the recently introduced algorithms named *binary*. Features are extracted using FAST with a circular radius of 9 pixels (FAST-9). Harris cornerness measure is used to retain a certain number of detected features. As FAST does not provide scale information, the authors use a pyramidal representation of the image and detect features at each scale. The other addition to the FAST detector is the computation of features' orientation by means of the intensity centroid [24]. The centroid c is computed using the moments of the patch centred at the pixel of interest p. The orientation is obtained by constructing a vector $\overrightarrow{pc}$ and taking the enclosed angle. ORB descriptor is based on BRIEF which is a bit string description of an image patch constructed from a set of binary intensity tests. The other difference between ORB and BRIEF, in addition to the orientation invariance, is the fact that ORB learns the optimal sampling pairs whereas BRIEF uses randomly chosen sampling pairs for the binary intensity tests. The output of these tests forms the descriptor elements of an extracted feature.

The advantage with ORB and the binary descriptors (in general) is two fold: (i) computing the descriptor can be accelerated to more than 40 folds compared to SURF especially when using the POPCNT instruction from SSE4.2 [4]. (ii) The matching process is also considerably fast using the efficient binary XOR instruction to compute the matching score.

Binary Robust Invariant Scalable Keypoints - BRISK similarly to ORB, this algorithm belongs to the binary descriptors family. Features are extracted using a variant of the AGAST algorithm (Adaptive and Generic Accelerated Segment Test) [16], which is an extension of FAST. The detector goes a step beyond ORB in order to achieve scale invariance. For this purpose, the algorithm searches for maxima in the image plane and the continuous scale space (via quadratic fitting). In contrast to ORB (and inherently BRIEF), BRISK uses a handcrafted sampling pattern composed of concentric circles centred at the feature (Fig. 3a). Gaussian smoothing corresponding to the distance of the circle centre to the feature is applied before the tests to account for aliasing effects. Sampling pairs are distinguished into short and long pairs depending on whether the distance between them is below a certain threshold d_{max} or greater than d_{min}. The long pairs are used to compute the local gradient (of the patch) that defines the orientation of the feature. The short pairs are then rotated accordingly to obtain the sought rotation invariance and used to compute the binary descriptor by means of intensity tests.

Fast Retina Keypoint - FREAK in contrast to ORB and BRISK, FREAK was proposed as a binary feature descriptor only. The descriptor corresponds to a bit string encoding a series of intensity tests around the pixel of interest. FREAK is similar to BRISK with regard to the sampling pattern used for the tests as well as the orientation mechanism. It is also similar to ORB by using machine learning techniques to learn the optimal set of sampling pairs. FREAK suggests using a retinal sampling grid (circular) having higher density of points near the centre and dropping exponentially away from the centre (Fig. 3a–c). As the name of the descriptor suggests, the sampling pattern corresponds to the distribution of receptive fields in the human retina. The consequence is that the test pairs naturally form a coarse-to-fine approach. The first pairs compare sampling points in the outer rings of the pattern whereas the last pairs correspond to the inner rings. The descriptor matching procedure also takes advantage of the coarse-to-fine approach in order to speed up the process. The first 128 bits of the descriptor are compared (coarse) and if they fall within a predefined threshold then the



(a) (b) (c)

Fig. 3 Sampling patterns of the binary descriptors. **a** BRISK pattern **b** FREAK pattern. The red and brown circles indicate the size of the Gaussian kernel used to smooth each sampling point for BRISK and FREAK, respectively. **c** Receptive fields of the human retina which inspired FREAK descriptor. Images adapted from [13]

remaining bits are tested. It was claimed that this coarse-to-fine approach allows to discard 90% of the candidates therefore accelerating the matching process [2].

Local Intensity Order Pattern - LIOP similarly to FREAK, LIOP also is a feature descriptor only [35]. As its name suggests, it is based on local intensity order pattern to encode the local ordinal information of each pixel. The overall ordinal information is used to divide the local patch into sub-regions, which are used for computing the descriptor. The first step to calculate the descriptors is a Gaussian smoothing of the image since the relative order is sensitive to noise. Instead of computing a dominant rotation of the considered patch as in SIFT, the descriptor is computed in an orientation independent fashion. In order to make order patterns rotation invariant, the neighbourhood of samples around a pixel x is taken in a rotation-covariant manner: the points are sampled anticlockwise on a circle of radius r around x. The considered image region is divided into N sub-regions (bins) based on the local ordering of the pixels. Histograms of order patterns are computed for each bin and then combined to form the final descriptor using a weighting scheme. The latter is claimed to be, at the same time, distinctive and invariant to monotonic intensity changes as well as image rotations.

4.2 Comparison Metrics

Our evaluation encompasses the performance of both feature extraction and description algorithms. We used the repeatability and matching scores [18] to measure the stability of the extracted features between images. In addition, we used the *recall*/1-*precision* curves to evaluate the descriptors [17].

Repeatability is computed from the corresponding regions between two images using a known homography. In theory, the maximum number of correspondences that can be computed between two images is defined by the number of features in the common region of the two images. Correspondence is defined in terms of the overlap error (6) which is the error in the image area covered by the regions. Two regions (a and b) correspond if their overlap error is sufficiently small:

$$1 - \frac{R_{\mu_a} \cap R_{(H^{\mathrm{T}}\mu_b H)}}{R_{\mu_a} \cup R_{(H^{\mathrm{T}}\mu_b H)}} < \epsilon_0 \tag{6}$$

where R_μ represents the elliptic region defined by $x^{\mathrm{T}}\mu x = 1$ and H is the homography linking the two images. $R_{(H^{\mathrm{T}}\mu_b H)}$ refers to the ellipse of feature b reprojected on the first image (using H). The union of the regions is $R_{\mu_a} \cup R_{(H^{\mathrm{T}}\mu_b H)}$ and their intersection is $R_{\mu_a} \cap R_{(H^{\mathrm{T}}\mu_b H)}$. The areas of the intersection and union of the regions are computed numerically.

The repeatability score is then computed as the ratio between the number of correspondences (C^+), for a given error threshold, and the number of features that are present in the first image C.

$$repeatability = \frac{|correspondences|}{|f_{\mathrm{A}}|} = \frac{C^+}{C} \tag{7}$$

Note that [27] use $\min(|f_{\mathrm{A}}|, |f_{\mathrm{B}}|)$ in the denominator of Eq. 7 which could be misleading in some cases as reported by [8]. Indeed, for instance, if a detector computes 50 features in the first image and only 1 in the second image, where this single feature belongs to the 50 previously detected points, a repeatability score of 1 is returned. We ran our experiments using both versions of Eq. 7 where very subtle variations were observed. We chose to show the

results using the modified version given in Eq. 7 for all the experiments.

It is to be noted that different feature extraction algorithms yield features/regions of various sizes. Using the original extracted regions would favour detectors with large regions. A solution to this issue, as suggested by [18], would be to normalise the size of the originally detected regions to a fixed size prior to computing the overlap error. The influence of this parameter is studied in Section 5.1.

The matching score provides a good evaluation of the algorithm's distinctiveness. It is computed as the ratio between the number of correct matches and the number of detected regions in the first image. Correct matches (defined as C^*) are determined by comparing the descriptors in a nearest neighbour way. The Euclidean distance is used for the distribution based descriptors whereas the Hamming distance is used for the binary descriptors. To consider a region match correct, the geometric overlap error needs to be below a defined threshold. The matching score can therefore be written as

$$matchingscore = \frac{C^+ \cap C^*}{C} \tag{8}$$

As suggested by [18], this score provides an idea on distinctiveness of the extracted features. If the matching score results do not follow the repeatability scores for a particular feature type, this means that their distinctiveness differs from the distinctiveness of the other detectors.

Finally, we used the $recall/1 - precision$ curves when testing the performance of the descriptors [17]. It is based on the number of correct matches and false matches between features detected in an image pair. Two features A and B are labelled as matches if the distance between their descriptors D_A and D_B is below a threshold μ. For this, each descriptor from image A (reference) has to be compared with all descriptors from the transformed image B allowing to count the numbers of correct and false matches. The *recall*/1-*precision* curves are obtained by varying the value of μ. Recall is computed as the ratio of the correct matches M^+ to the number of correspondences between a pair of images:

$$recall = \frac{|correctmatches|}{|correspondences|} = \frac{M^+}{C^+} \tag{9}$$

Precision is the ratio of correct matches M^+ to the number of total matches M^* (false matches and correct matches). It can be written as

$$precision = \frac{|correctmatches|}{|totalmatches|} = \frac{M^+}{M^*} \tag{10}$$

In order to label matches as correct (M^+), we need to verify whether the matched features are also the correspondences provided by the ground truth homography.

4.3 Thermal Dataset

A dataset consisting of sequences of far-infrared images was used for the experiments. We used two thermal cameras: (i) a Thermoteknix Miricle[1] 307K, with a spatial resolution of 640 × 480 pixels, and a temperature resolution of roughly 40 bits per degree Centigrade and (ii) a FLIR TAU2[2] with a similar spatial resolution and thermal sensitivity lower than 50mK. Lens distortion from the Miricle camera was removed using a calibration pattern specifically designed for the thermal modality introduced by [34]. The same process was carried out for the images acquired using the TAU2 camera. Images were captured from a range of environments with a variety of degrees of thermal-contrast. Lower thermal-contrast (meaning a smaller range of true observed surface temperatures) has the effect of reducing the SNR of the images.

4.3.1 Ground Truth

In order to compute the comparison metrics presented above, ground truth information is needed. For 3D scenes, this would require accurate 3D models or laser scanners. However, homographies relating two views can be computed for planar scenes. In our dataset, we captured various scenes where a dominant plane was present. This enables to compute the homography $\mathbf{H}$ which relates a point $\mathbf{x}$ in the first image to $\mathbf{x}'$ in the second image:

$$\mathbf{x}' = \mathbf{H}.\mathbf{x} \tag{11}$$

where $\mathbf{x} = (x, y, 1)^{\mathrm{T}}$ and $\mathbf{x}' = (x', y', 1)^{\mathrm{T}}$ are in homogeneous coordinates.

To compute the homography relating two views, we combine two methods: (i) the feature-based image alignment and (ii) the enhanced correlation coefficient (ECC) based forward additive algorithm introduced by [7]. This choice was mainly motivated by two facts: (i) it was reported by [31] that combining both methods is more effective than each method alone (ii) the ECC based algorithm showed better performance under noisy images which (noise) is inherent in thermal imagery [7].

Firstly, random sample consensus (RANSAC) is used to estimate an initial homography $\hat{\mathbf{H}}$. Features are extracted from both images and correspondences are computed. RANSAC is then used to filter out the outliers and compute the optimum (initial) homography $\hat{\mathbf{H}}$. Secondly, $\hat{\mathbf{H}}$ is used to initialise the forward additive ECC algorithm [7] to compute a refined and more accurate homography $\mathbf{H}$. Figure 4 shows

[1] http://www.thermoteknix.com/products/oem-thermal-imaging/miricle-thermal-imaging-modules/.

[2] http://www.flir.com/cores/display/?id=54717.

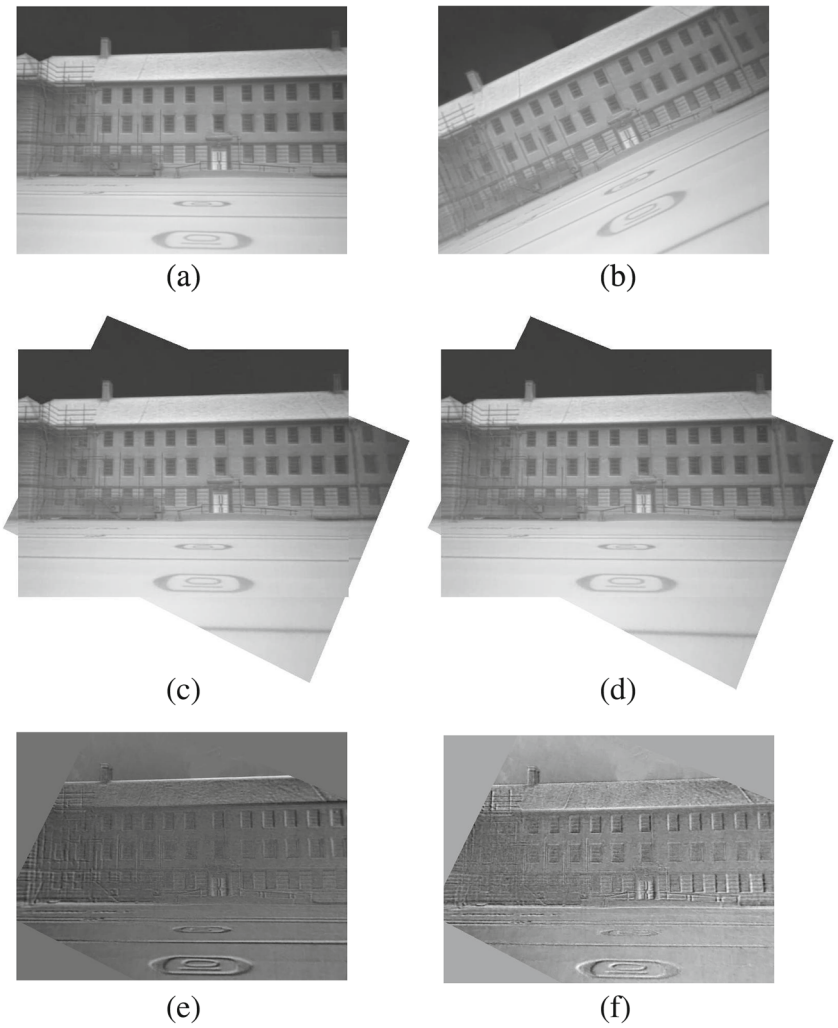


Fig. 4 Example showing mosaics generated using the computed homographies **a** original image **b** rotated images **c** mosaic from the initial homography $\hat{\mathbf{H}}$ **d** mosaic from the final homography **H** **e** **f** corresponding errors. Note that the contrast of both error images has been similarly stretched for better visibility

the mosaics that were created using the initial homography $\hat{\mathbf{H}}$ (Fig. 4c) and the refined homography **H** (Fig. 4d) with the corresponding errors. We can see that the final homography provides a better mosaic.

4.3.2 Dataset

The dataset consists of 64 video sequences capturing various *real-world* scenes (Fig. 5) and accounting for different image transforms which are representative of *realistic* applications (Figs. 6 and 7). This is in contrast to the dataset proposed by [8] which was captured using pictures held in a glass support.

In total, the dataset comprises 3654 images. These sequences were selected to cover diverse types of scenes, with varying SNR values. As suggested in [22], environments with low SNR may prove difficult to exploit for matching. This is notably the case for environments where the temperature difference between the imaged objects is relatively low (e.g. *office*). In such cases, the SNR may drop below a certain threshold (30/1) yielding a significant decrease in the percentage of matches.

The effects of eight image transformations (listed below) were explored for the considered feature detection/description algorithms. Varying severity levels were adopted to investigate the stability of the algorithms. The evaluation can therefore use the images in consecutive order to reflect navigation applications where continuous motion and video feed is assumed. Alternatively, sampling the sequences differently allows testing against larger image transformations.

Image transformations

- *Rotation*: the camera rotates around the optical axis from 0° to 90° which results in an in-plane rotation (Fig. 6a).
- *Panning*: the cameras pans sideways causing changes in viewpoint. This reflects a moving camera which undertakes a horizontal turn inducing viewpoint variations (Fig. 6c).

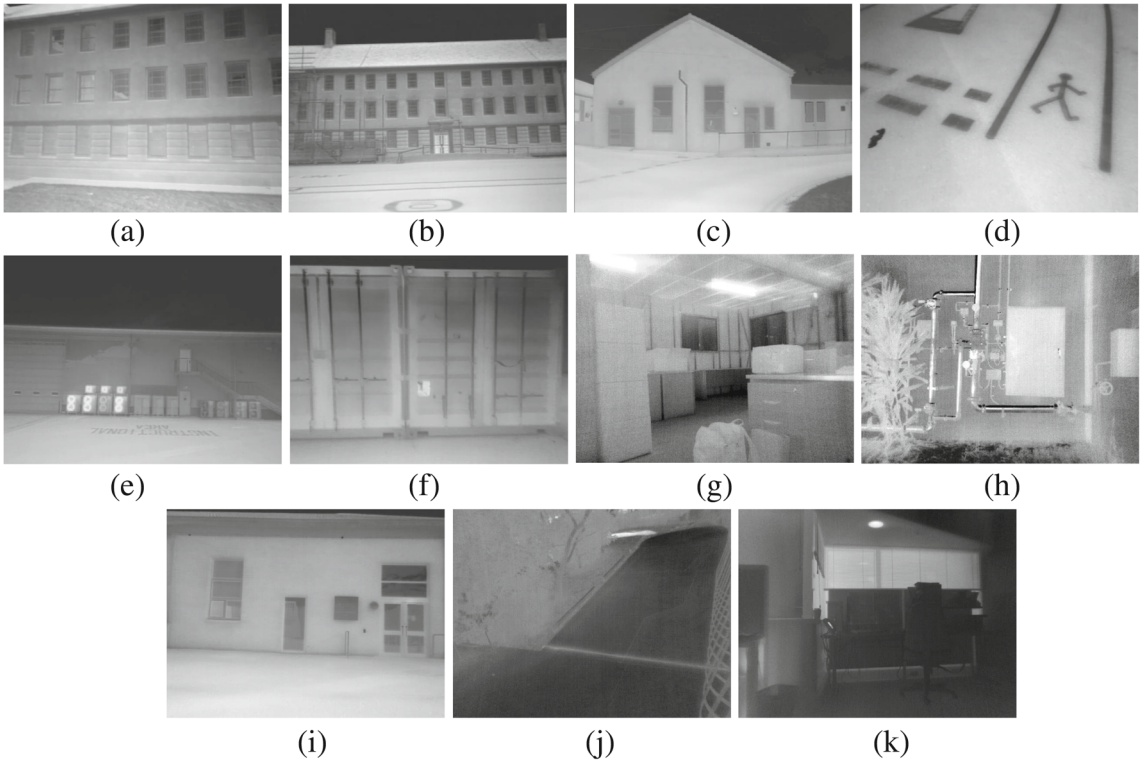


Fig. 5 Captured environments **a** *building1* **b** *building2* **c** *building3* **d** *fan* **e** *ground* **f** *container* **g** *office* **h** *pipes* **i** *wall* **j** *driveway* **k** *desk*

- *Zoom*: here, the camera moves towards or far from the imaged scene. This reflects a scale change between the frames (Fig. 6e).
- *Motion blur*: the camera pans sideways with varying speeds (Fig. 7e). We considered three variants from normal to fast as multiples of 4 pixels per frame (4, 8 and 12).
- *Non-uniformity noise*: here images were acquired at intervals of one minute after the NUC was performed. The total sequence corresponds to a 5-minute camera operation time yielding an accumulation of additional non-uniformity noise in the image (Fig. 7c).
- *Time-of-day*: in contrast to visible-band imagery, thermal sensors do not suffer from the change in lighting conditions unless it corresponds to a lasting phenomenon which induces temperature variations. For this reason, we did not include a lighting variation sequence in the dataset. Instead, we considered the generation of the time-of-day sequences. The latter were captured at different times of the day (on the hour) from fixed locations (Fig. 7c).

Note that all sequences contain additional non-uniformity noise that builds up during the acquisition period. Moreover, the various motion sequences contain amounts of jitter and camera blur due to the manual operation of the camera. This should not affect the evaluation process as these effects are common in applications based on moving cameras and apply for all algorithms.

4.4 Evaluation

The evaluation process can be divided into two distinctive parts for feature detectors and descriptors.

Feature Detectors we test the repeatability of the feature extractors as well as the number of geometric correspondences. In addition, we compute the matching scores and the number of matches using LIOP for the considered detectors. LIOP was chosen as it has been demonstrated in the literature to perform reasonably well in different scenarios. In addition, it is independent from all the detectors considered in this study so that to avoid bias towards a specific detector. The aim here is to study the distinctiveness of the extracted features regardless of the native descriptor. We also compute the matching scores of the DoG and Fast-Hessian using their native descriptors i.e. SIFT and SURF, respectively. This was intended to compare the behaviour of the native implementations to their combination with LIOP.

Feature Descriptors we tested the descriptors performance using the same feature extraction algorithm, namely

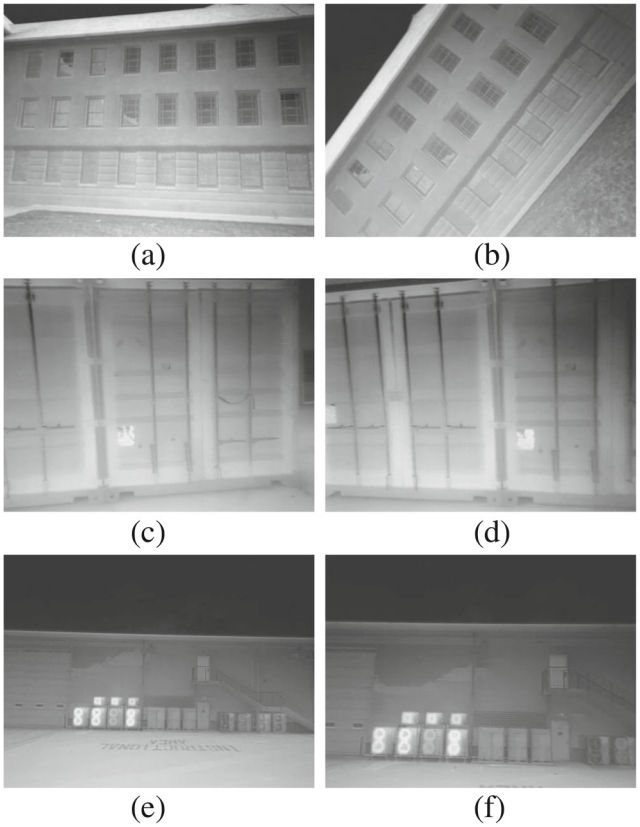


Fig. 6 Sample images from our dataset illustrating the studied image transforms. Left: original image. Right: transformed image. **a**, **b** rotation **c**, **d** panning **e**, **f** zoom

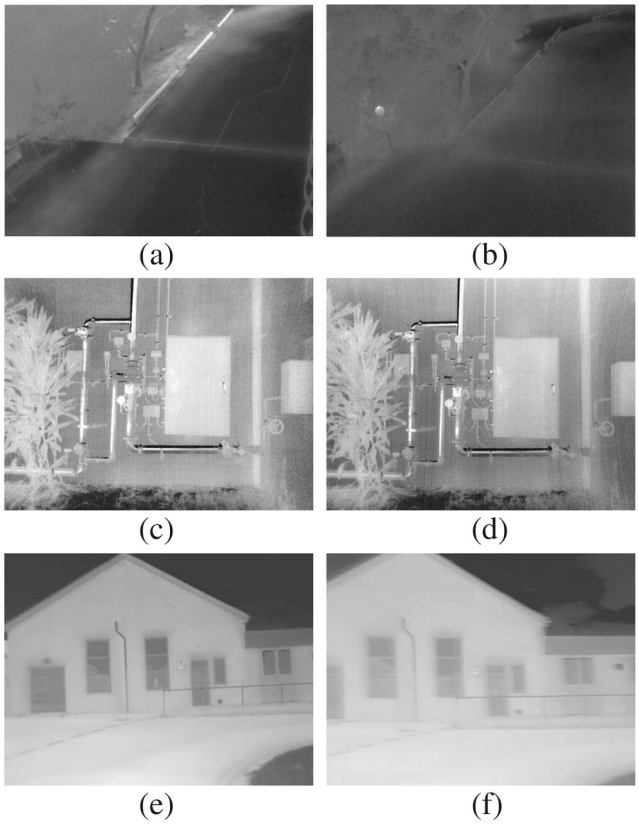


Fig. 7 Sample images from our dataset illustrating the studied image transforms. Left: original image. Right: transformed image. **a**, **b** time-of-day **c**, **d** non-uniformity noise **e**, **f** motion blur

Fast-Hessian and DoG. The latter were selected as common extractors from the analysis of the outcomes of the previous tests. Using the same features for all the descriptors algorithms provides a better insight into the performance of the descriptors using *recall*/1-*precision* curves. A summary of the studied descriptors is given in Table 1.

We also tested the extended SURF descriptor (SURF-128), which has a length of 128 instead of 64. The aim was to investigate whether extending the descriptor size would benefit the matching performance of SURF. We found out that, on average, SURF provided better results than SURF-128. This corresponds to the findings of Bay et al. where they stated that "... we discovered that the extended descriptor loses with respect to recall, but exhibits better precision. Overall, the effect of the extended version is minimal" [3]. For this reason, we kept only the more commonly used version of SURF in the comparisons (64D). Also, one needs to bear in mind that this extension comes at an additional computational description and matching cost that might prohibit its usage in certain applications.

4.5 Experimental Variations

The dataset described in Section 4.3.2 allows to undertake two types of tests: (i) simulate continuous camera operations and compute the comparison metrics consecutively (inter-frames) or (ii) simulate large image transforms and evaluate the robustness of the algorithms against these large baselines.

Table 1 Investigated description algorithms

Algorithm	Type	Size	Matching metric
SIFT	Vector of floats	128	Euclidean distance
SURF	Vector of floats	64	Euclidean distance
LIOP	Vector of floats	144	Euclidean distance
ORB	Binary string	256 bits	Hamming distance
BRISK	Binary string	512 bits	Hamming distance
FREAK	Binary string	512 bits	Hamming distance

Therefore, for the in-plane rotation for example, we would evaluate the performance of the detectors as the rotation angle increases with the same interval of 10° for the first test. This is usually the case in navigation applications e.g. visual odometry where the inter-frame motion is not expected to exceed small rotation angles. In the second evaluation, we use the first image as reference and we compute the metrics for increasing angles (10°- 20°- ...- 90°). This provides an insight on the robustness of the algorithms to large rotations. It is relevant in some applications such as image mosaicing and loop closure in visual odometry.

4.6 Implementation

Here we provide information regarding the implementation of the evaluated algorithms. We used the code provided by the original authors for LIOP[3] and Fast-Hessian+SURF.[4] In contrast, the original SIFT (DoG) implementation is only available as binary. This does not allow the tuning of the default parameters which is crucial in our benchmark. We tried two publicly available implementations from OpenCV (based on the code of Rob Hess[5]) and Andrea Vedaldi.[6] For the reasons discussed in Section 5.2.1 we opted for the former (OpenCV). There are various implementations for FAST[7] where we chose to use the OpenCV code suggested by the original author. It was reported by [8] that Rosden's implementation of Harris detector was equivalent to the one provided in OpenCV. The latter was therefore used in our benchmark as well as the code for Shi and Tomasi's GFTT. FREAK, BRISK and ORB original implementations have been integrated to OpenCV. We therefore used these codes for the evaluation. For CenSurE, we adopted an OpenCV implementation coined STAR which is a modified version of the original algorithm proposed by [1] for increased speed and stability.

5 Experimental Results and Discussion

In this Section, we detail the steps undertaken in the investigation of the algorithms' performance in the thermal band. We based our evaluation, in part, on the state-of-the-art feature detector performance analysis (in the visible band) proposed by [18]. However, several modifications had to be made to the protocol.

[3] http://zhwang.me/publication/liop/index.html.

[4] http://www.vision.ee.ethz.ch/~surf/download.html.

[5] http://robwhess.github.io/opensift/.

[6] http://www.vlfeat.org/overview/sift.html.

[7] http://www.edwardrosten.com/work/fast.html.

Indeed, as reported by [33], the original protocol presented limitations that made it particularly unsuitable for performance analysis in the thermal modality. The **first** limitation relates to the utilisation of feature detectors with default parameters which may lead to relatively low numbers of features on thermal images. The sensitivity threshold of a given algorithm has to be adequately tuned to obtain decent numbers of features when used on far-infrared images. The main reason behind it is that these values were initially set (by their developers) to provide sufficient performance in the visible-band modality. As we shift in the spectral domain from the visible band to far-infrared, the content of the images vary considerably making the original settings unusable. This is illustrated in Fig. 8. It shows the behaviour of the DoG detector using the same sensitivity threshold on a pair of visible-band and thermal images capturing the same scene. We can clearly observe the difference in the number of extracted interest points in both images (thermal: 125, visible: 954). For this reason, we had to tune the sensitivity thresholds of the studied algorithms to obtain a fairly similar amount of features. This operation had to be carried out for each sequence used in the experiments to account for the different environments. A somewhat arbitrary fixed value has to be selected (either sensitivity, or feature count) and applied to each sequence. It was our view that when implementing a feature detector in practice, sensitivity is usually adjusted to achieve an approximate desired number of features, and for a particular application this is likely to be a similar number regardless of the environment. Using a fixed sensitivity across high SNR and low SNR environments could result in excessive or negligible feature counts for such various environments.

The **second** limitation is independent from the authors as it relates to the time of the benchmark publication (2005). Indeed, different techniques have emerged since 2005 e.g. Fast-Hessian & SURF, CenSurE and the binary algorithms - ORB, BRISK and FREAK. The **third** limitation relates to the dataset used in the evaluation. It was reported that using a low number of images may cause an over-fitting problem where the conclusions might not reflect the performance of the algorithms in real-world applications.

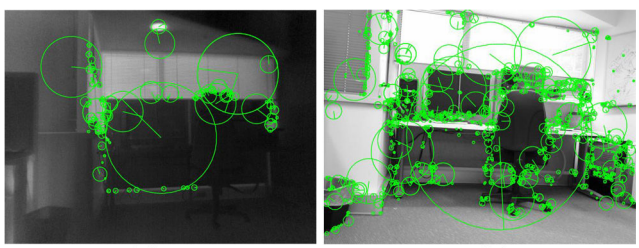


Fig. 8 Example showing DoG features for a pair of thermal and visible-band images using identical settings. **a** 125 features detected on the thermal image **b** 954 features detected on the visible image

Note that we also experimented with the affine algorithms proposed in [18]. Despite tuning the affine detectors, the number of returned features was very low. For instance, using the same sensitivity threshold, Harris returns the required number of features (e.g. 600) while Harris-Affine returns only 2 keypoints. As stated in [32], the estimation of affine shape can be applied to any initial point given that the determinant of the second moment matrix is larger than zero and the SNR is sufficiently large. This would explain the low number of extracted features in thermal imagery where the SNR is usually low. Moreover, the computation times are considerably higher than the more popular algorithms which might prohibit their usage in navigation applications.

5.1 Initial Results

First, we provide some insights into the different benchmarking parameters that can influence the performance of a given feature detector/descriptor algorithm. We chose a pair of images corresponding to the original and a transformed image from all sequences to carry out these tests. The plots in Fig. 11a–c are averaged over all the sequences for each feature detector. Figures 9 and 10 show the extracted interest points using the studied detectors on sample images

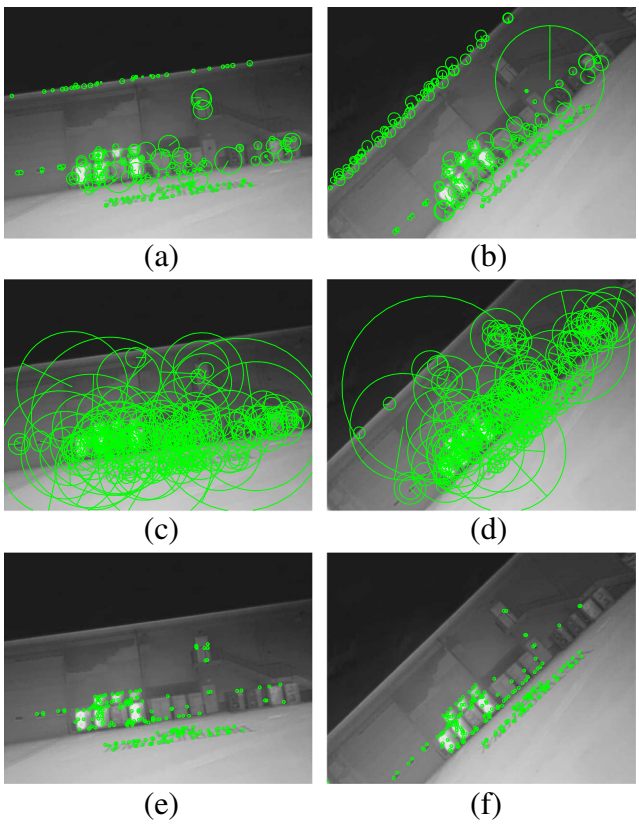


Fig. 9 Example of detected features on sample images. Left: reference image. Right: transformed image. **a**, **b** DoG **c**, **d** Fast-Hessian **e**, **f** Harris

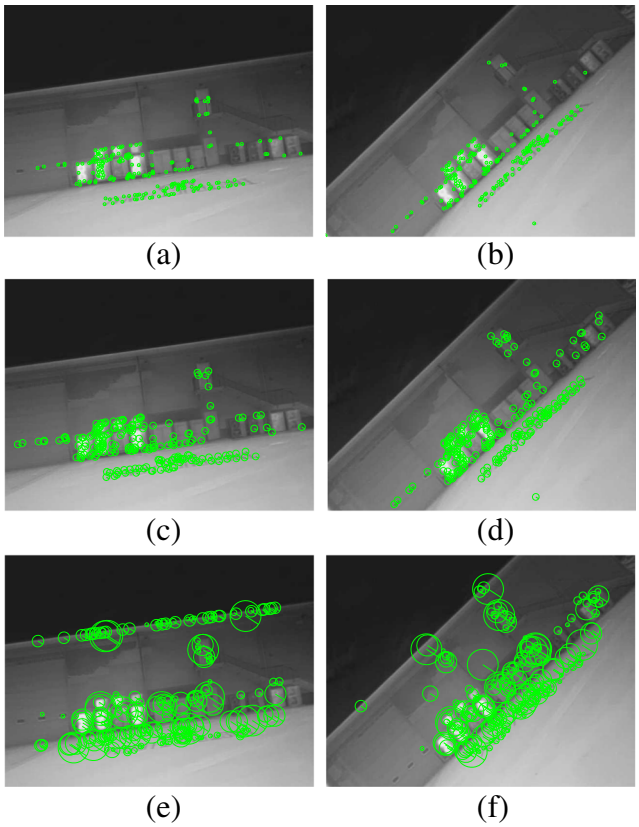


Fig. 10 Example of detected features on sample images. Left: reference image. Right: transformed image. **a**, **b** GFTT **c**, **d** FAST **e**, **f** CenSurE

containing rotation transformation. The keypoints are plotted on both frames corresponding to the original and the transformed image to illustrate the variation of their location with respect to the image transform. Note that we plot only 200 features for clarity.

Features' Accuracy here we test the effect of the overlap error threshold. Figure 11a shows the repeatability score as a function of the overlap error. Naturally, as the threshold is increased, larger numbers of features are considered as *correspondences* leading to an increase in the repeatability scores. The relative ordering of the detectors remains almost unchanged except for FAST and Fast-Hessian after 50% and 60% overlap error, respectively. DoG and CenSurE behave similarly by continuously increasing their repeatability scores with growing overlap errors. Likewise, Harris and GFTT have similar curves. In contrast to the other algorithms, their scores seem to saturate beyond an error of 40%. Choosing an overlap error threshold of 30% would therefore limit the bias in the ranking of the evaluated feature detectors.

Normalised Region Size the effect of the normalised feature size on the repeatability score is illustrated in Fig. 11b. This

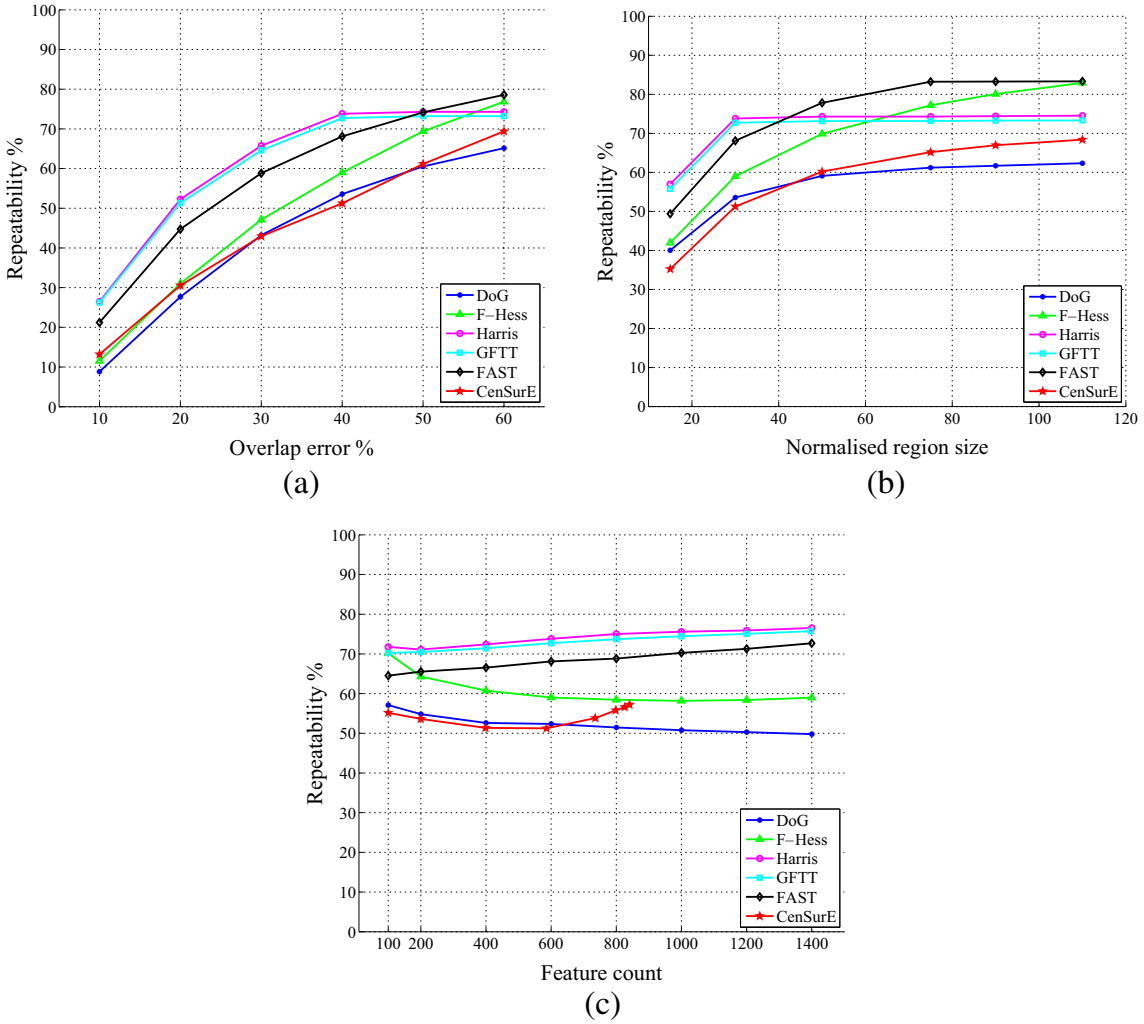


Fig. 11 Repeatability scores for various benchmark settings. **a** Repeatability scores vs overlap error **b** repeatability scores vs normalised region size **c** repeatability scores vs region density

test was carried out with an overlap error threshold fixed at 30%. Similarly to the previous test, the relative ordering of the algorithms remains the same. This indicates that the used benchmarking setup is not very sensitive to this parameter. It is also to be noted that beyond a certain normalised size, we can notice a flattening of the curve for most algorithms.

Region Density here we show the effect of increasing the number of extracted features on the repeatability scores of the different algorithms. Extracting less or more features can be accommodated by varying the value of one parameter i.e. the sensitivity threshold. This test illustrates the effect of varying such parameter on the repeatability of the detectors. This way, we may remove the bias towards dense responses as different algorithms extract disparate numbers of features.

As shown in Fig. 11c, FAST, Harris and GFTT exhibit the same behaviour: their repeatability score increases slightly with growing numbers of extracted regions. In contrast, DoG and Fast-Hessian scores are relatively higher for very low feature counts (<300) and remain levelled for numbers higher than 400 features. These high scores correspond to relatively high thresholds which in some cases lead to very few correspondences. This may impede the use of such detectors in applications where a minimal number of features are required. Similarly, CenSurE showed relatively steady repeatability scores. Despite the tuning of its sensitivity threshold, the number of extracted features could not be increased beyond 1250. An average of 800 interest points are extracted as shown in Fig. 11c.

5.2 Evaluation of Interest Point Detectors

In this first set of experiments, we evaluate the repeatability and matching scores of the studied detectors. The ideal detector would exhibit a 100% repeatability score with

large numbers of correspondences regardless of the image transformation. The number of correspondences is always displayed with the relative repeatability scores as they allow to appreciate their absolute value.

We considered an overlap error of 30%, a normalised region size of 30 pixels and fixed numbers of extracted features for each detector (600). We tuned the sensitivity thresholds of the evaluated extraction algorithms to have a similar number of features. The idea here is to address the issue related to the bias towards dense responses (see Section 5.1).

The results of these experiments are shown in Figs. 12, 13, 14, 15, 16, 17, 18, 19, 20 and 21. In general, two plots are provided: (i) the first is for the consecutive image transforms, which are more likely to happen in navigation applications and (ii) the second reflects the performance of the detectors against large image transforms (rotation, zoom, etc.). Typically, in the second case, the left hand side of the repeatability scores curves correspond to small transformations and indicate the performance with respect to small variations. The absolute number of (geometric) correspondences drops more rapidly than the repeatability score. Indeed, as we move right in the curve, larger transformations occur between the images resulting in either lower image quality or less overlap between the images leading to fewer correspondences (detected features).

Rotation Variation Figures 12–13 show the effect of varying in-plane rotation on the repeatability and matching scores. Harris, GFTT and FAST provide the highest repeatability. However, their matching scores are lower than DoG, CenSurE and Fast-Hessian. Indeed, the ranking is reversed when looking at Fig. 12a and b. This said, the performance of all detectors is not affected by the continuous inter-frame rotations. Interestingly, Fast-Hessian gets better matching score combined with LIOP than with the native SURF descriptor (Fig. 12b). From Fig. 13, we can note that all detectors are sensitive to larger variations. Indeed, both metrics seem to decline as the rotation angle is increased. Note that the lower numbers of matched features indicate that the previously computed geometric correspondences were not distinctive enough to be matched using their descriptors.

Camera Panning Figures 14–15 show the performance against variations in the camera panning angle. The algorithms seem less affected than rotation variations for the consecutive angle changes. This means that they should be robust for navigation applications where small panning angles are expected from frame to frame. Despite having steady repeatability curves for large variations, the absolute number of matches drops significantly. This can impact applications where large viewpoint variations are

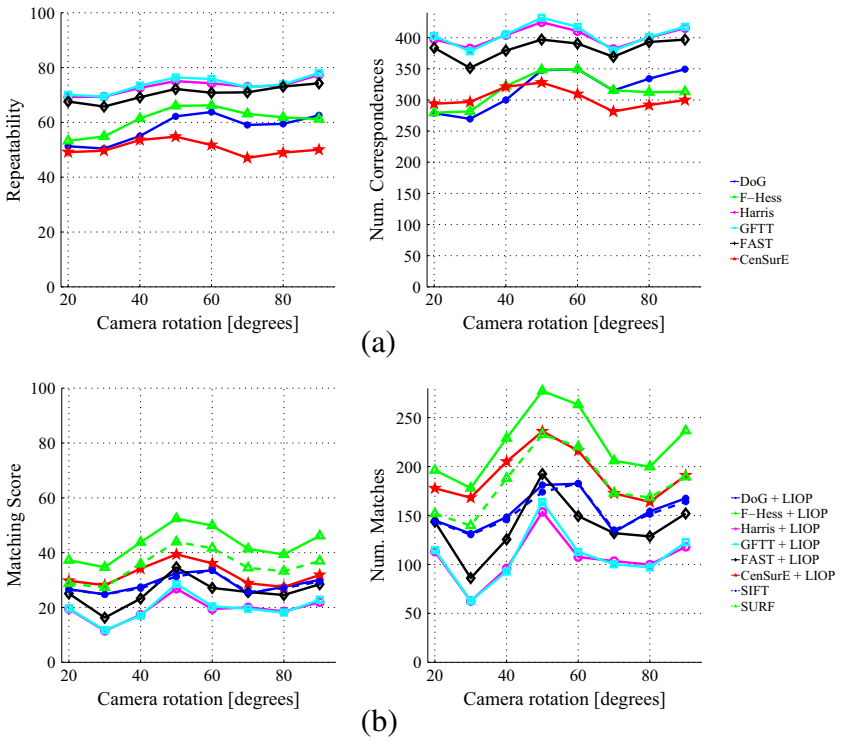


Fig. 12 Effects of in-plane rotation. **a** repeatability scores and number of correspondences **b** matching score and number of matches. Dashed lines show the results for DoG and Fast-Hessian with their original descriptor

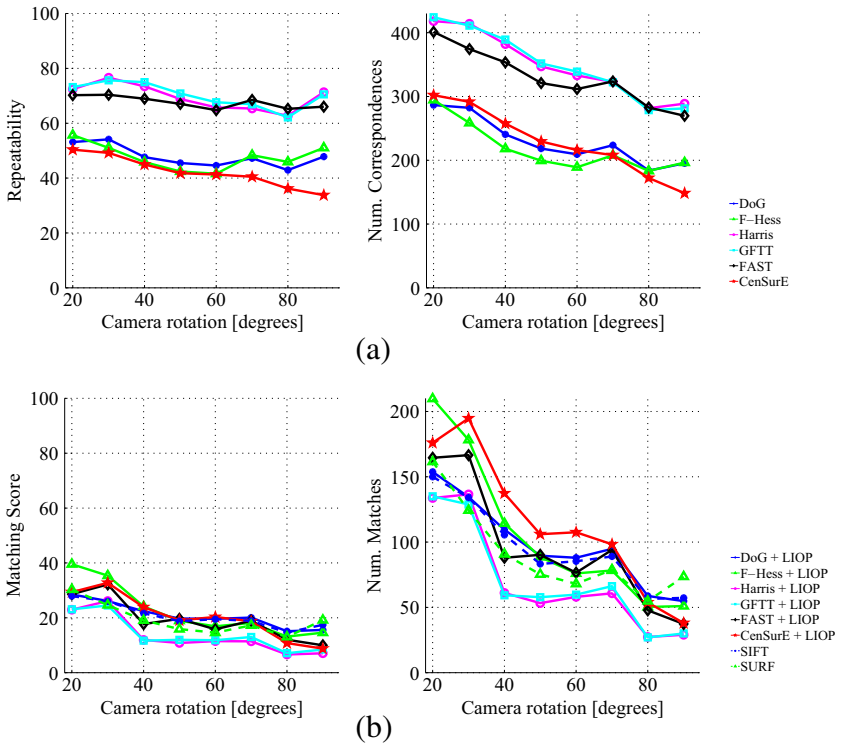


Fig. 13 Effects of large in-plane rotation angles. **a** repeatability scores and number of correspondences **b** matching scores and number of matches. Dashed lines show the results for DoG and Fast-Hessian with their original descriptors

expected e.g. image stitching. For this case, Fast-Hessian and CenSurE outperform the other algorithms in terms of matching scores/number of matches.

Scale Change Figures 16–17 show the the effects of scale variations on the detectors performance. The algorithms are resilient to continuous scale changes. This is generally the

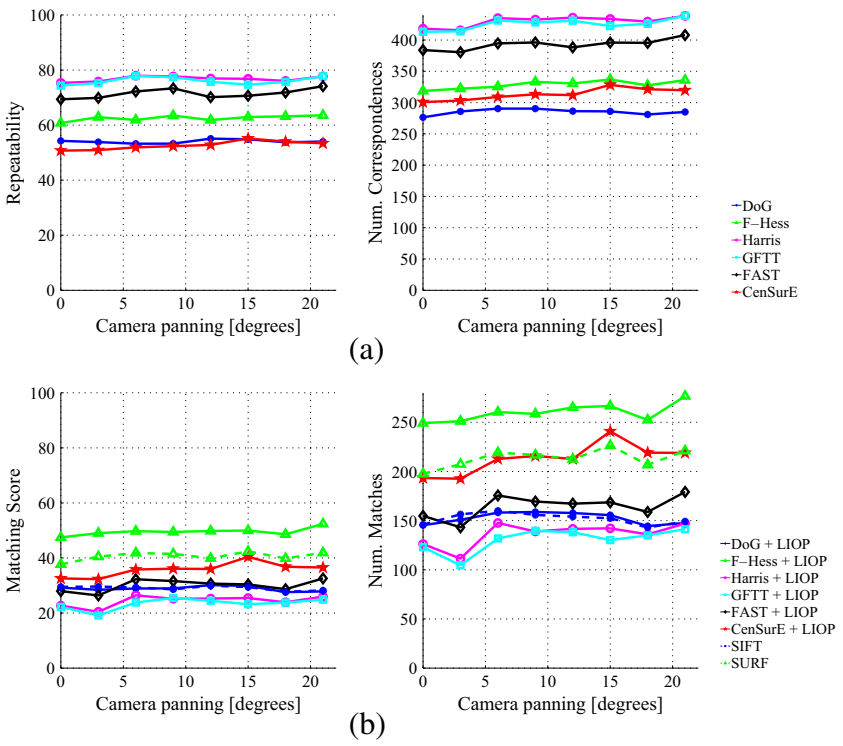


Fig. 14 Effects of camera panning. **a** repeatability scores and number of correspondences **b** matching score and number of matches. Dashed lines show the results for DoG and Fast-Hessian with their original descriptor

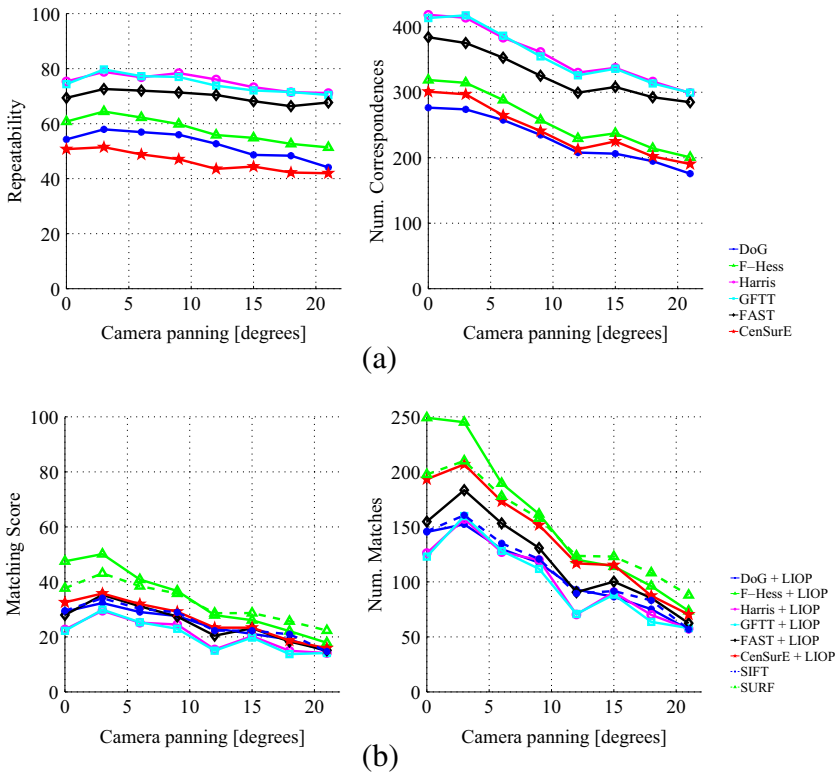


Fig. 15 Effects of large camera panning angles. **a** repeatability scores and number of correspondences **b** matching scores and number of matches. Dashed lines show the results for DoG and Fast-Hessian with their original descriptors

case in navigation applications e.g. visual odometry where the inter-frame scale variation is relatively small. Harris and GFTT outperform the others in terms of repeatability but rank last with respect to the matching scores. This indicates that most of the correspondences were due to accidental overlap.

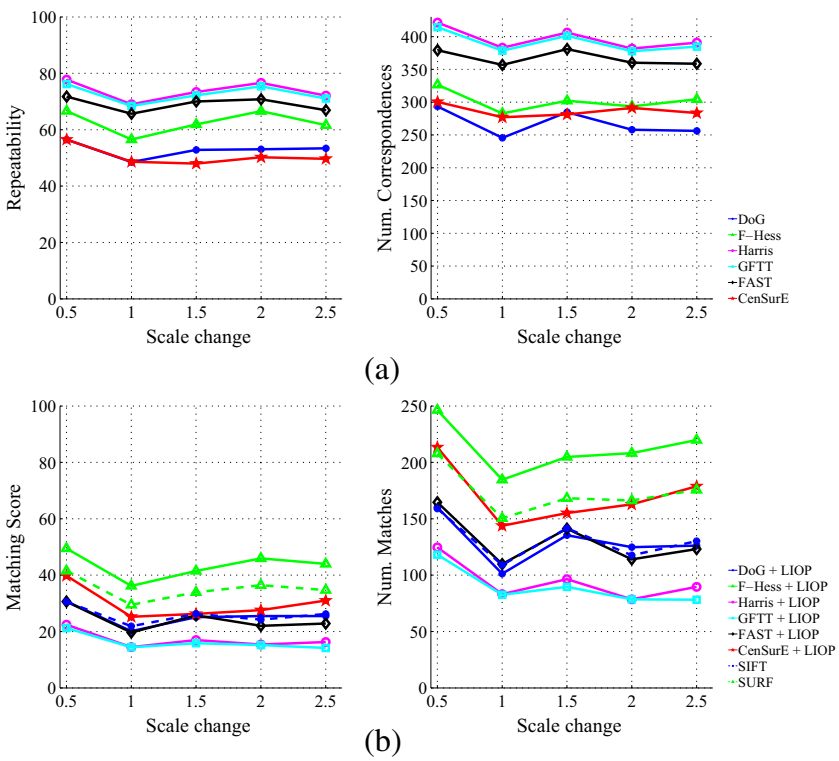


Fig. 16 Effects of scale change. **a** repeatability scores and number of correspondences **b** matching score and number of matches. Dashed lines show the results for DoG and Fast-Hessian with their original descriptor

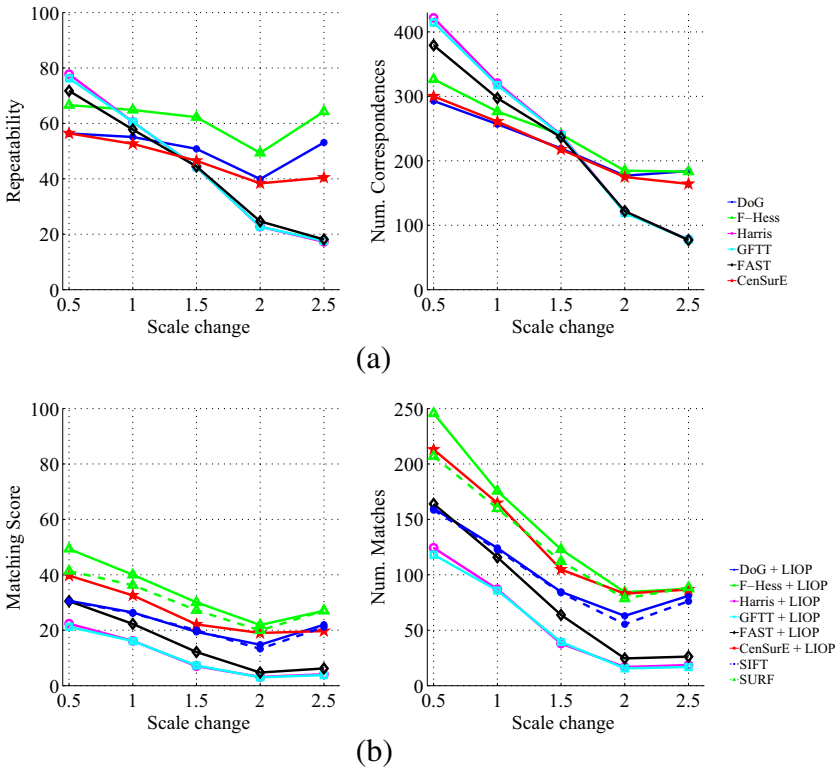


Fig. 17 Effects of large scale change. **a** repeatability scores and number of correspondences **b** matching scores and number of matches. Dashed lines show the results for DoG and Fast-Hessian with their original descriptors

When the scale change is relatively large (Fig. 17), we can observe that the scale independent feature detectors perform very poorly. Notably, both metrics for Harris, GFTT and FAST drop rapidly. On the other hand, algorithms which take scale information into account when extracting features are less affected (DoG, Fast-Hessian and CenSurE).

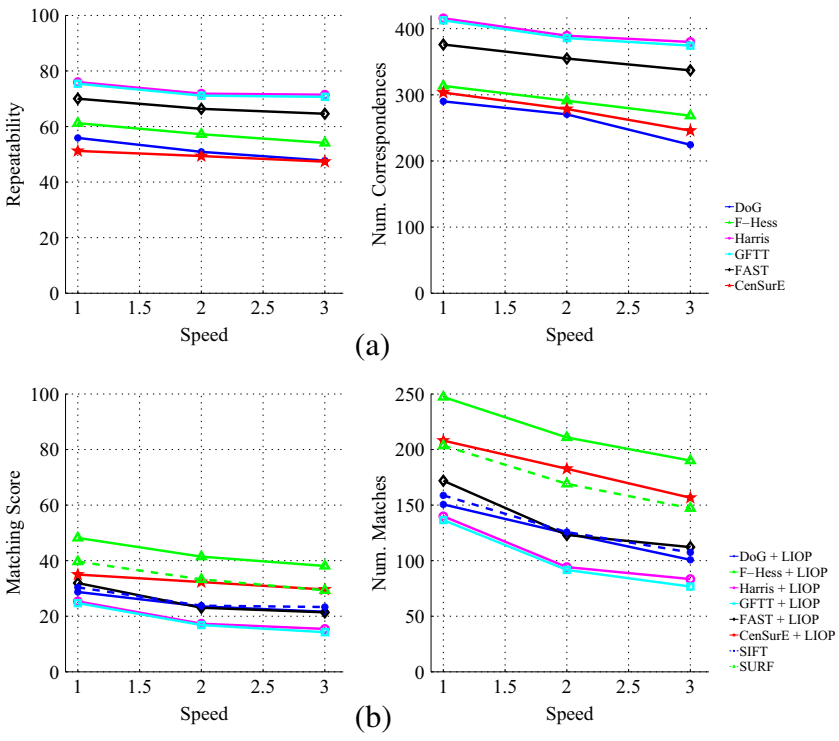


Fig. 18 Effects of motion blur. **a** repeatability scores and number of correspondences **b** matching score and number of matches. Dashed lines show the results for DoG and Fast-Hessian with their original descriptor

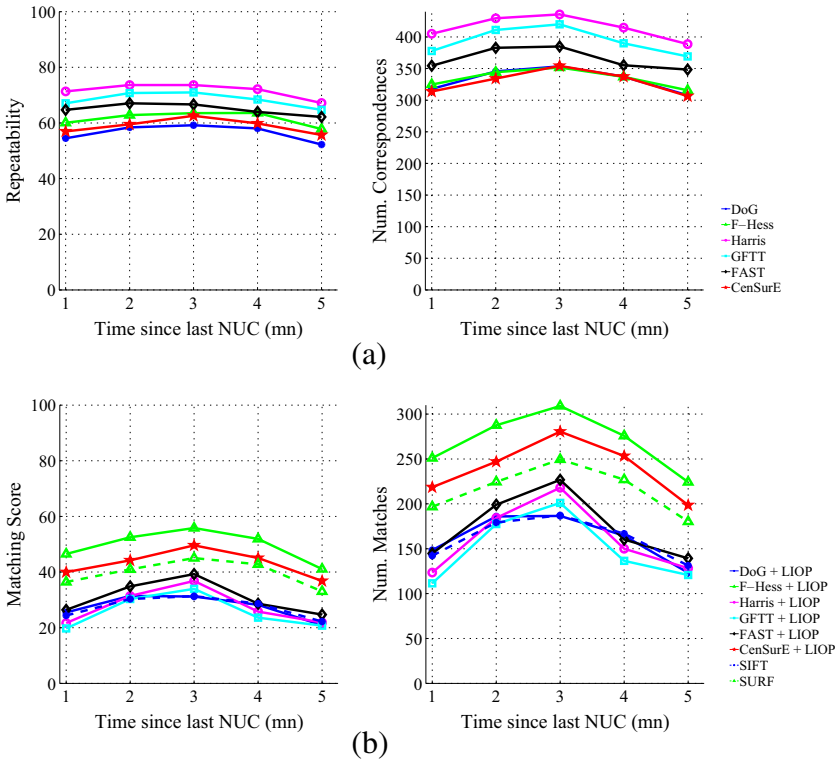


Fig. 19 Effects of non-uniformity noise. **a** repeatability scores and number of correspondences **b** matching score and number of matches. Dashed lines show the results for DoG and Fast-Hessian with their original descriptor

Motion Blur Figure 18 shows the results for the effects of varying motion blur. Here, we can note that the algorithms metrics (repeatability & matching scores) drop as the motion speed is increased. This is emphasised in the absolute numbers of matches. Here again, Fast-Hessian outperforms the other algorithms.

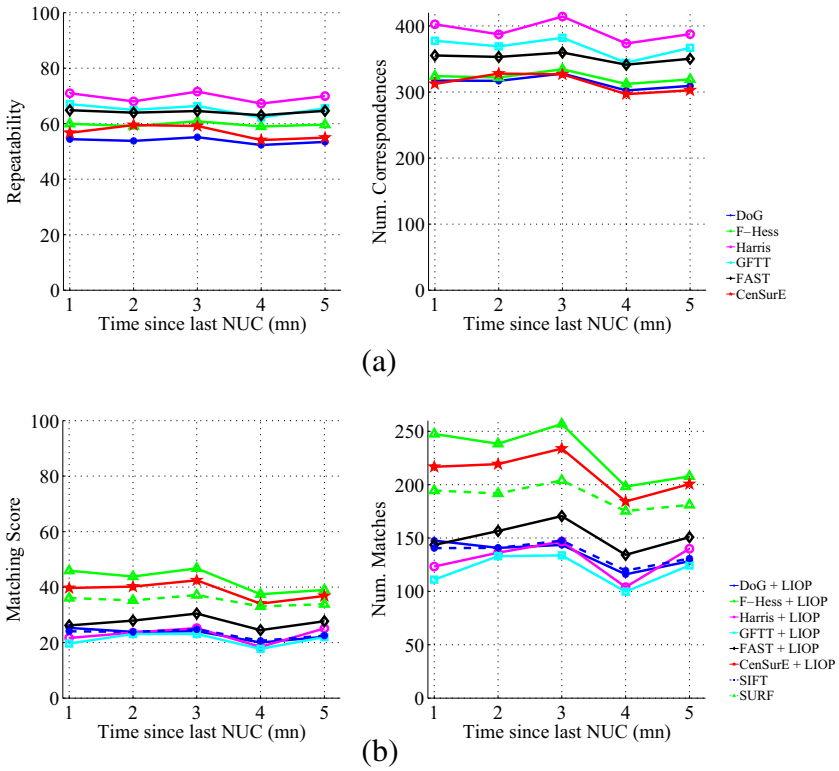


Fig. 20 Effects of large non-uniformity noise. **a** repeatability scores and number of correspondences **b** matching scores and number of matches. Dashed lines show the results for DoG and Fast-Hessian with their original descriptors

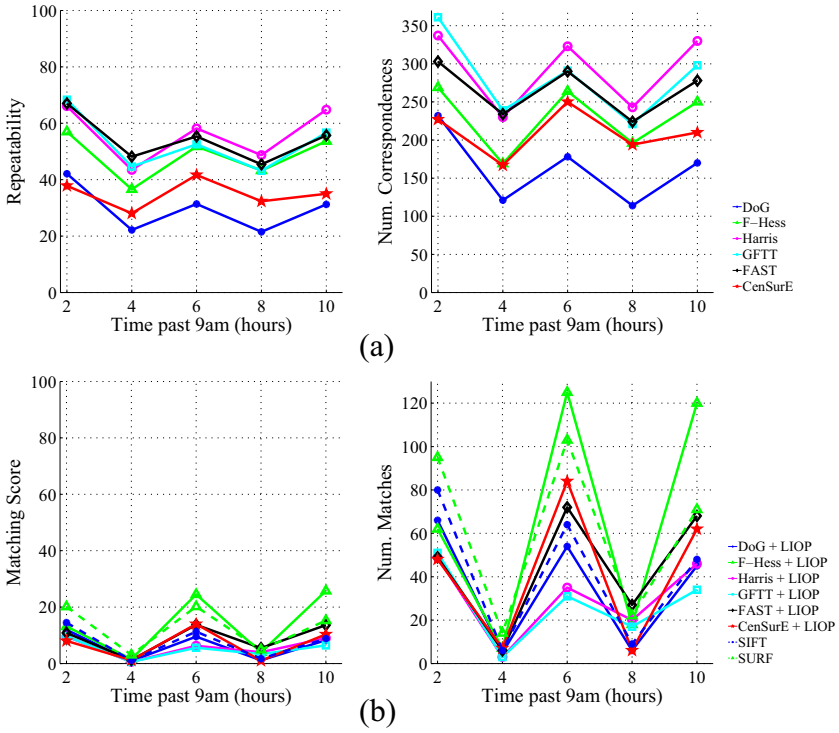


Fig. 21 Effects of time-of-day. **a** repeatability scores and number of correspondences **b** matching score and number of matches. Dashed lines show the results for DoG and Fast-Hessian with their original descriptor

Non-uniformity Noise Figures 19–20 show the effect of non-uniformity noise. Although decay can be observed (Fig. 20), non-uniformity noise has a small impact on the performance of the detectors. The largest effect was noticed in the indoor environments where the low SNR accentuates the effects of non-uniformity noise. Harris outperformed the other algorithms with respect to repeatability in both consecutive and large variations. However, Fast-Hessian provided the best matching score. It can be clearly seen that using the descriptors in computing the matches instead of the overlap only (geometry) leads to a significant decrease in the matching score with respect to the repeatability score.

Time of day Figure 21 shows the performance of the detectors with respect to the variation of the time-of-day. We illustrate the effect of a relatively long inter-frame time period (2 hours), although such variation is not expected in navigation applications where the time frame is of the order of milliseconds. However, the insights gained in this test may be useful for other applications such as place recognition.

All the algorithms exhibit a similar behaviour for this image transform where their metrics fluctuate with time. In particular, the number of matches drops to very low levels between time instants corresponding to large image variations. The two drops correspond to time periods with the largest variation in daylight.

5.2.1 DoG Results

As it can be seen in the results presented above, the DoG detector did not provide the best performance for most image transforms. We suspected an implementation issue as a possible cause. For this reason, we tested the original DoG implementation[8] against two publicly available codes from OpenCV (used in our evaluation) and Andrea Vedaldi. A series of tests were conducted. The repeatability scores and the number of correspondences are shown in Fig. 22 for two image transforms corresponding to in-plane rotation and non-uniformity noise (averaged over all the sequences). We can clearly observe that the OpenCV implementation provides relatively similar outcomes to the original code whereas Vedaldi's software behaves slightly differently. Indeed, even lower repeatability scores are obtained by the latter in comparison to the others. Note that the algorithms were used with default parameters since the original implementation does not allow to tune the settings. The

[8] http://www.cs.ubc.ca/~lowe/keypoints/.

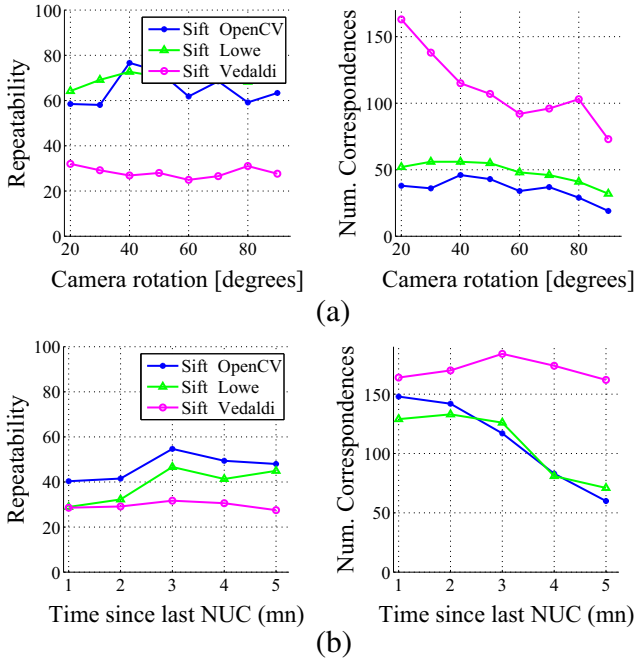


Fig. 22 Repeatability scores and numbers of correspondences for SIFT-OpenCV, SIFT-Lowe and SIFT-Vedaldi. **a** in-plane rotation **b** non-uniformity noise

implication is that we obtain a lower number of extracted features which is reflected in the very low correspondences. The main reason for that lies in the fact that the detector was originally tuned for visible-band images where contrast variations are expected more than for thermal imagery. From these tests, we can conclude that the behaviour of DoG in the evaluation is not due to an implementation issue.

5.2.2 General Observations

In general, and looking at Figs. 12–21, one can note that the repeatability and matching scores plots are quite similar. However, the latter are rather lower than the former. In addition, there are also differences with respect to the relative ranking of the detectors per image transform type.

In broad terms, similar curves shapes are obtained when using either the native descriptors (SIFT and SURF) or the combination of the detectors with LIOP descriptor for most image transforms. In general, using DoG with its native descriptor provides similar performance to its combination with LIOP. This suggests that both descriptors respond in a comparable manner to the different image transforms. Comparing Fast-Hessian with its native descriptor scores to its combination with LIOP shows that its performance is maintained in cases and degraded in others. This indicates that sometimes better performance can be achieved when combining different algorithms rather than using them in their default configuration.

The other comment is that using a common descriptor (in this case LIOP) to evaluate the performance of the detectors is less likely to be biased by the shortcomings/strengths of the native detector/descriptor combinations.

The results obtained so far provide a good insight into two aspects of feature detectors; namely the repeatability and distinctiveness. Looking at Figs. 12–21, we can draw conclusions on the performance of the studied algorithms with respect to the investigated image transforms. Although demonstrated in [18] to perform well on visible-band images, DoG did not perform well in most cases. In contrast, FAST and GFTT provided high repeatability scores. However, when it came to distinctiveness, they both obtained lower scores. This was more noticeable for scale changes. Fast-Hessian seems to perform consistently in terms of repeatability and distinctiveness in all experiments. We could observe that even when its repeatability scores were average for some tests, its matching scores were the best. This suggests that features extracted using Fast-Hessian are very distinctive.

On a more general level, and in contrast to what was suggested in [8], blob-like detectors (DoG, Fast-Hessian and CenSurE) provide lower repeatability scores than the corner extractors (Harris, GFTT and FAST). However, the trend for matching scores is quite the opposite i.e. the blob-like detectors achieve better performance. This disparity can be explained in two ways: the normalisation of features' size and the location of the extracted interest points. Indeed, looking at Figs. 9 and 10 we can note that, for instance, the detected Harris corners are close to each other. This yields high repeatability scores due in part to accidental overlaps. In contrast, we can observe that Fast-Hessian features are more distributed in the image space.

5.3 Evaluation of Feature Descriptors

In this test, we used the same feature detector for all the descriptors. The main driver for this experiment is to test the performance of the descriptors regardless of their detector's behaviour (if applicable). Similar settings to the repeatability evaluation were also used here to compute the *recall*/1-*precision* curves. Two sets of results are shown for each image transform. The first set represents the performance of the descriptors using Fast-Hessian detector for small and large image transform variations. The second set corresponds to the same experiment using DoG detector. We investigated the performance of different feature detection/description combinations. In what follows,

we show the results for Fast-Hessian and DoG combined with the studied descriptors.

Observation while performing the detector/descriptor investigation, we noted that combining scale independent features e.g. Harris with the scale aware descriptors e.g. SIFT and SURF causes significant drops in the performance of the latter. This can be explained by the inability of the algorithms to compute distinctive descriptors as the detected region of interest is considerably small.

A workaround for this issue is to normalise the detected features to provide a sufficiently large neighbourhood which includes the relevant signal changes around the keypoint.

5.3.1 Matching Strategy

Declaring two features as a match depends on the considered matching strategy. We investigated two schemes. The first one consists of the nearest neighbour matching where two features (A and B) are deemed a match if the descriptor D_B is the nearest (distance wise) to D_A and the distance between them is below a threshold i.e. $\|D_A - D_B\| < t$. The second matching strategy that was implemented corresponds to the nearest neighbour distance ratio (NNDR), which states that two feature candidates match if the distance ratio between the first and second best matches (D_B & D_C) is below a user defined threshold i.e. $\frac{\|D_A - D_B\|}{\|D_A - D_C\|} < t$.

One can note from Fig. 23, illustrating performance plots using Fast-Hessian detector and averaged for in-plane rotation sequences, that there are differences in terms of the descriptors performance with respect to the matching strategy. However, the relative ranking of the descriptors remains similar for both matching approaches. The most significant difference is in terms of precision. Indeed, implementing the NNDR matching leads to the penalisation of features which may have similar matches i.e. the distance to the nearest candidate descriptor is comparable to the distances to other descriptors. This in turn yields higher precision for the NNDR matching as it can be seen in Fig. 23b. In the following set of results, we choose to show the curves corresponding to the nearest neighbour matching. Note that we do not set a fixed search window around the descriptor of interest when computing the matches.

5.3.2 Comparison of Descriptors

Rotation Variation The performance of the descriptors was evaluated against in-plane rotation using images with a small rotation angle (15°) and a relatively larger angle of approximately 50°. Note that the images also contain non-uniformity noise. We can observe from Fig. 24 that the descriptors performance varies with the feature detection algorithm i.e. DoG and Fast-Hessian. However, the general observation is that the performance of all descriptors degrades for larger rotation angles.

Surprisingly, SIFT seems to perform better when computed on Fast-Hessian features. Similarly, most descriptors obtained higher precision scores when computed on Fast-Hessian features. LIOP, followed by SURF and SIFT, outperformed the binary descriptors for small rotation variation on Fast-Hessian features. However, it was superseded by SIFT and SURF for large angle changes (Fast-Hessian features). This indicates that the rotation invariant sampling

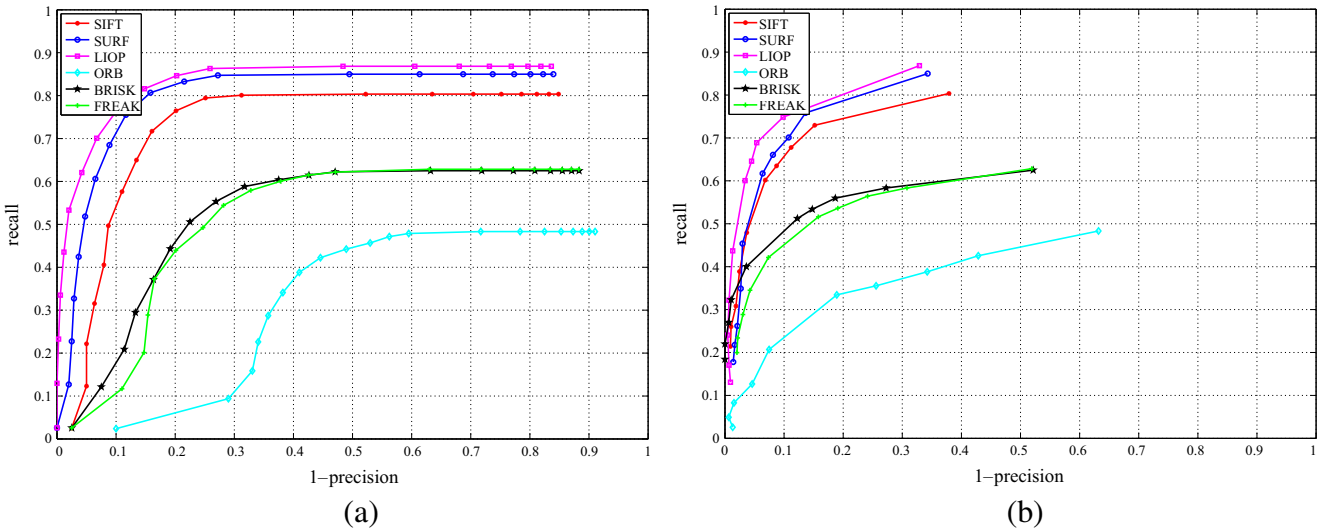


Fig. 23 Comparison of different matching strategies **a** nearest neighbour **b** NNDR

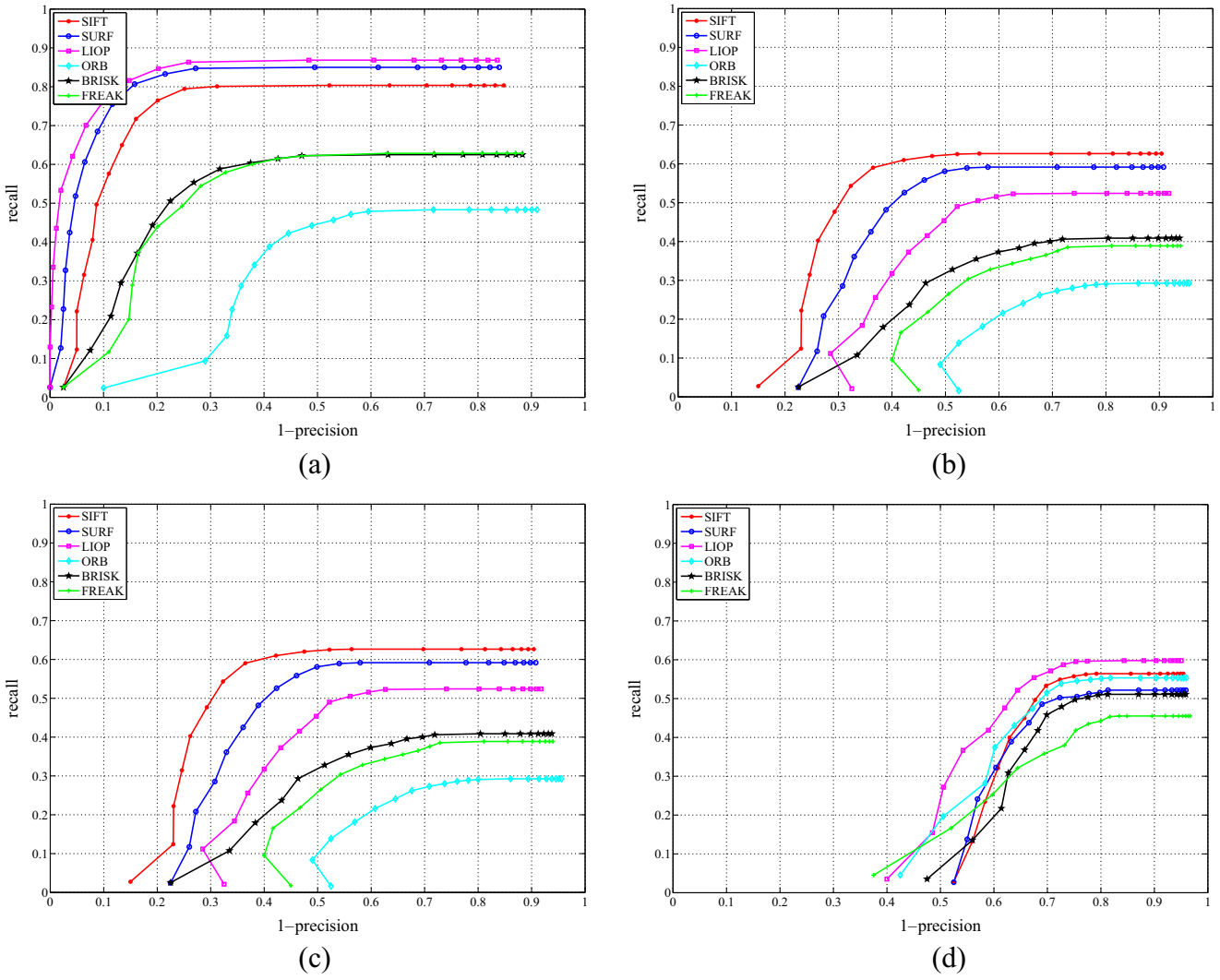


Fig. 24 Evaluation of rotation variation **a**, **b**, Fast-Hessian features with small and large rotation angles **c**, **d** DoG features with small and large rotation angles

mechanism of LIOP may be vulnerable to large angles where the underlying monotonic intensity changes might not hold. This reflects the plots in Fig. 13b where the performance of SURF is better than LIOP (Fast-Hessian + LIOP) for large rotation angles. FREAK and BRISK had similar outcomes while ORB obtained the lowest scores (with Fast-Hessian). Note that ORB achieved relatively high performance with DoG. We also looked briefly at a local shape based descriptor, histogram of oriented gradients [39]. Performance was evaluated on rotated, panned, and zoomed image pairs; where key points were detected used the Harris corner detector. Overall this descriptor gave lower precision and recall values, (2.0%, 36.4%), (1.5%, 5.9%) and (3.5%, 8.2%), for the rotated, panned, and zoomed images, respectively.

The use of descriptors is also prevalent in methods for facial detection and recognition. We investigated the performance of SURF, BRISK and FREAK, on head shots shifted by rotation, panning and zooming. The Harris corner detector was used for feature extraction. Like the outdoor scenes, SURF outperformed the other binary descriptors, where a precision and recall of (8.2%, 27.7%), (17.9%, 29.4%) and (13.2%, 30.4%) was obtained for rotation, panning and zooming respectively. BRISK obtained a precision and recall of (2.4%, 15.4%), (6.0%, 11.6%) and (7.6%, 28.6%) for rotation, panning and zooming respectively. For the same order of scenarios FREAK obtained a precision and recall of (2.4%, 13.8%), (4.8%, 9.8%) and (9.4%, 50.0%) respectively.

Camera Panning to evaluate the performance of the descriptors against camera panning, we used images with an angle varying between 5° to 20° (note also the presence of non-uniformity noise). Looking at Fig. 25, we can observe that the effect of camera panning has less impact on the performance of the descriptors than in-plane rotation. When Fast-Hessian features are used, we can note that SIFT achieves the best results for large angle variation. However, LIOP performs better in the other cases.

Scale Change to evaluate the performance of the descriptors against scale change, we used pairs of images with a small and large scale factor (0.5-2.5). From Fig. 26, we can observe that all descriptors are affected by large scale change more than in-plane rotation. Despite being computed on its scale-invariant features (DoG), SIFT obtains lower precision scores than LIOP and SURF for large scale variations. On the other hand, LIOP seems more affected when computed on Fast-Hessian features. The binary descriptors hardly cope with large scale change (for both features). In general, SIFT provides the best performance.

Motion Blur Figure 27 shows the descriptors' performance against two motion speeds. We can clearly observe that most detectors are not affected by motion blur. Indeed, matching works well for fast motion (Fig. 27b–d). Overall, SURF achieves the best performance. For DoG features, the binary descriptors obtain comparable results to the distribution

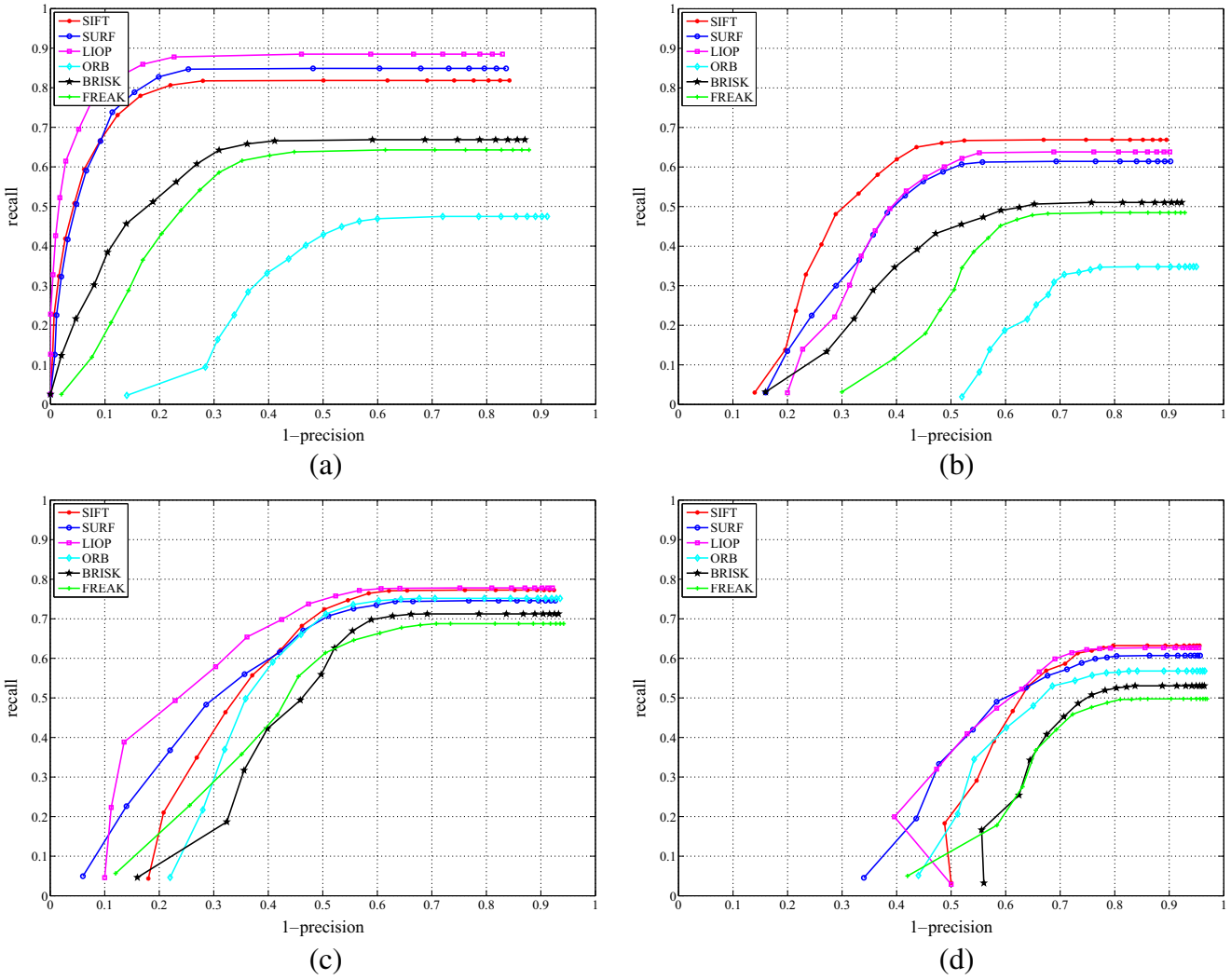


Fig. 25 Evaluation of camera panning **a**, **b** Fast-Hessian features with small and large panning angles **c**, **d** DoG features with small and large panning angles

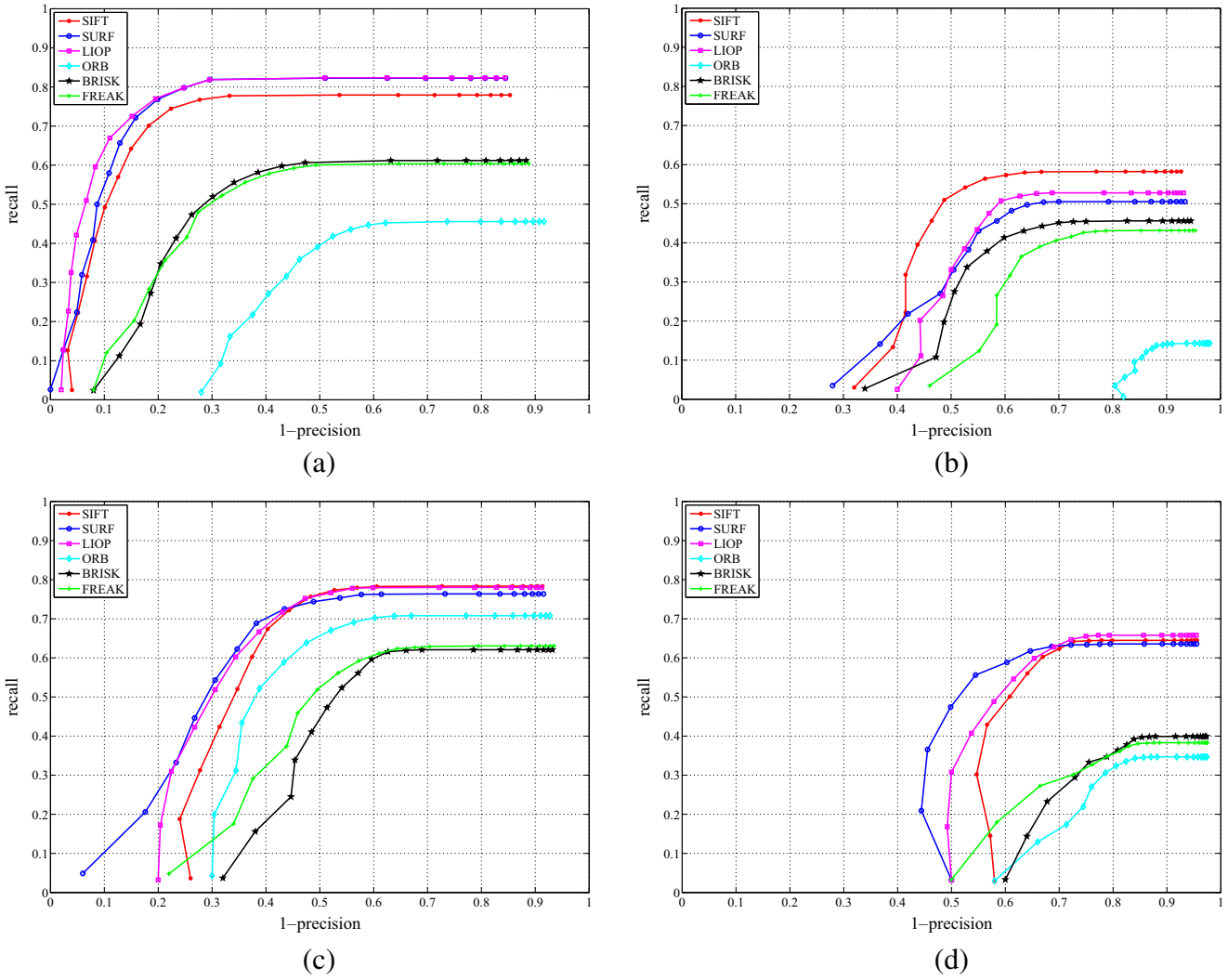


Fig. 26 Evaluation of scale change **a**, **b** Fast-Hessian features with small and large scale variations **c**, **d** DoG features with small and large scale variations

based descriptors. In addition, faster motion seems to have low impact when the descriptors are computed on DoG features. However, higher precision values are obtained with Fast-Hessian interest points. This reflects the findings in Section 5.2 where Fast-Hessian features showed more resilience to motion blur.

Non-uniformity Noise we evaluate the performance of the descriptors against the inherent non-uniformity noise of thermal cameras. We use pairs of images at time intervals of 1mn and 5mn, respectively (Fig. 28). A general observation is that descriptors are less sensitive to this type of noise than expected. Lower precision scores are obtained when computing the descriptors on DoG features (Fig. 28c) than Fast-Hessian (Fig. 28a). The impact of non-uniformity noise is therefore low except for indoor environments (*office*) where severe degradation of the performance was observed. This is mainly due to the low SNRs common to indoor environments as most objects are at similar temperatures (i.e. low contrast variations). This in turn amplifies the non-uniformity noise. Additionally, depending on the targeted application, longer operational times may be required (more than 5mn tested here) where more noise would build up in the image.

Time of Day (*Balcony*) to evaluate the performance against changing time-of-day, we used pairs of images corresponding to different acquisition times (2 hour gap). We can

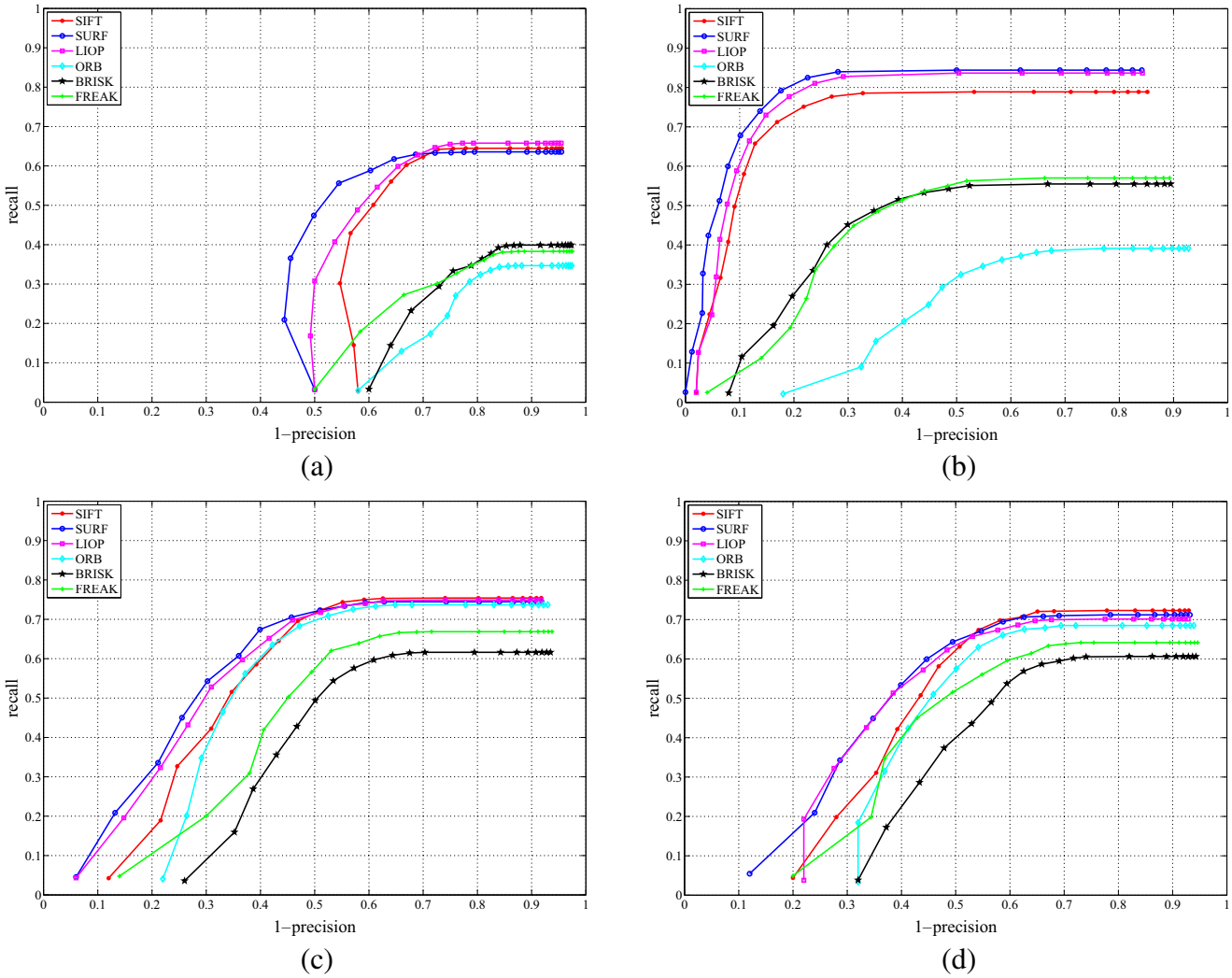


Fig. 27 Evaluation of motion blur **a**, **b** Fast-Hessian features with small and large motion speeds **c**, **d** DoG features with small and large motion speeds

observe from Fig. 29 that there is a large performance disparity when the descriptors are computed on the two types of features. Fast-Hessian features lead to higher precision values for all descriptors (even DoG+SIFT). SIFT obtains the best scores with Fast-Hessian features indicating that it is more robust to this type of variations. The same observation was reported in [8] for lighting variations in the visible-band imagery. This could be due to the descriptor normalisation and thresholding (values above 0.2 are recast to 0.2) suggested by Lowe to reduce the effects of illumination changes [15]. The binary descriptors obtain slightly higher recall scores for DoG features at the expense of very low precision values.

This test indicates that the performance of the descriptors can be affected when acquiring thermal images at different times (note that the time gap here is two hours). This is induced by the changing scene content in a similar fashion to the varying lighting conditions for visible band cameras. The only difference is that in thermal imagery the process might take more time as the changes are caused by variations in temperature (here two hours).

5.3.3 Discussion

The general observation is that the performance of a given descriptor is coupled to the used features. This is the case for gradient based descriptors (SIFT and SURF) as well as the intensity based binary descriptors. As reported in Section 5.3.2, most of the descriptors seem to perform better when computed on Fast-Hessian features for the studied image transformations.

From our experiments, we observed that Fast-Hessian extracts larger blobs in comparison to DoG. This would explain the performance disparity for the descriptors.

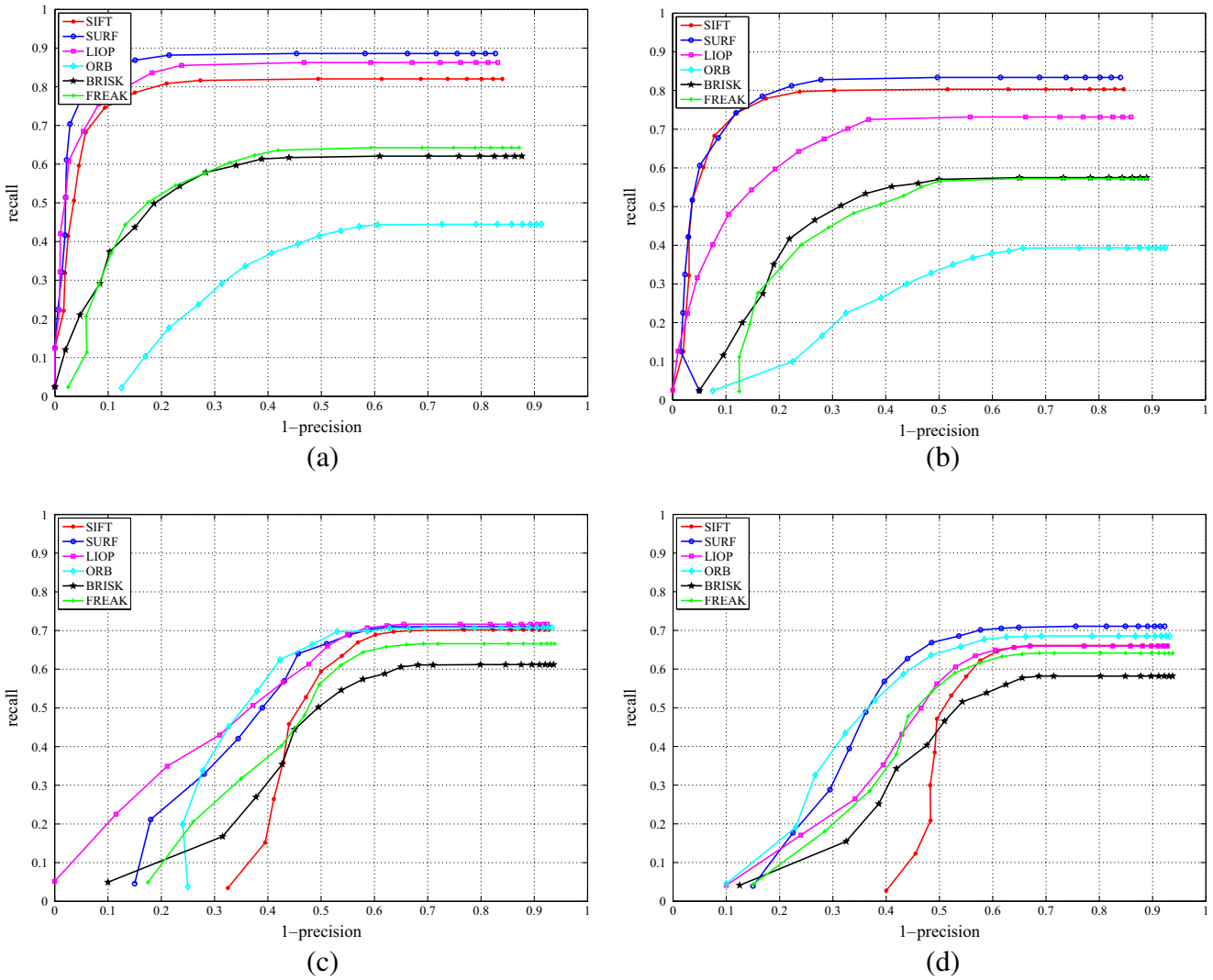


Fig. 28 Evaluation of non-uniformity noise **a**, **b** Fast-Hessian features with small and large noise variations **c**, **d** DoG features with small and large noise variations

Indeed, SURF and SIFT use a window around the keypoint to compute the descriptor where the size (of the window) is dependent on the scale of the feature. Therefore, the larger the size the more discriminative the descriptor is, given that there is enough signal variations in the neighbourhood of the keypoint. This applies also to BRISK where the sampling pattern is scaled according to the detection scale. In contrast, LIOP uses a patch of fixed size (default 41 pixels).

LIOP, which is also an intensity based descriptor, seems to perform better when computed on Fast-Hessian features. It can be ranked amongst the best descriptors for the studied image transforms. However, we must note that its performance degraded more than SIFT for large baseline variations. Overall, SURF also obtained good scores and can be ranked with the best performers. In contrast to SIFT, SURF is inherently less sensitive to noise. This comes from the fact that SURF integrates the gradient information in the neighbourhood of the keypoint whereas SIFT is based on the orientations of the individual gradients. In the binary descriptors category, FREAK provided the best results throughout the experiments with Fast-Hessian features. Note that SIFT obtained slightly better precision figures with the nearest neighbour ratio matching strategy.

5.4 Descriptors Invariance

In this Section, we evaluate the invariance of the descriptors against in-plane image rotation, camera panning, scale

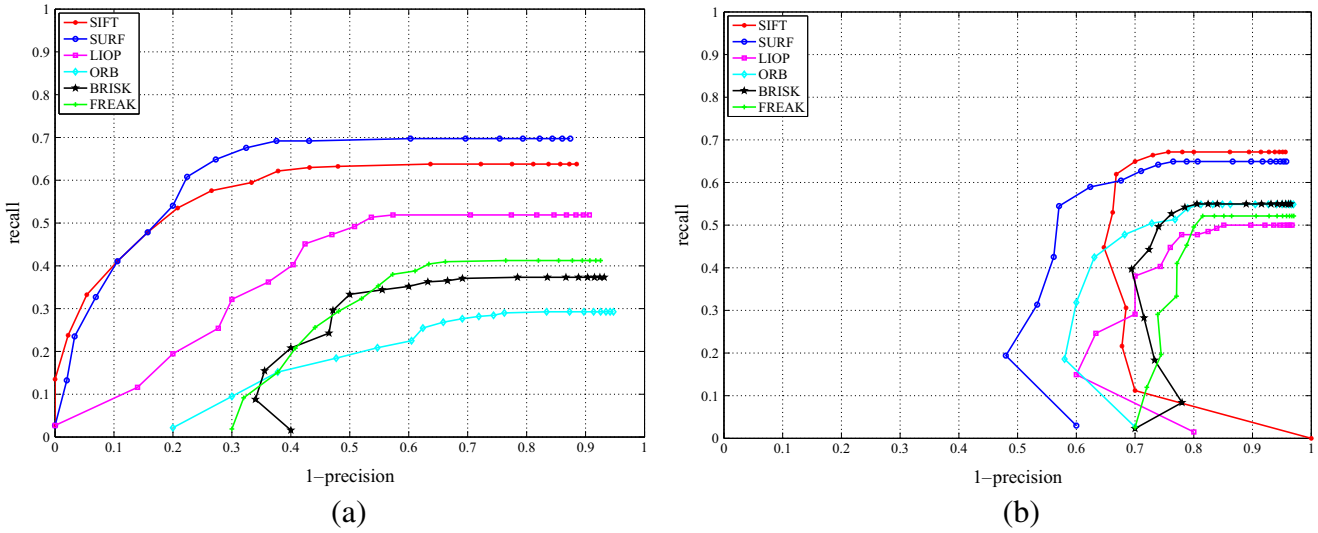


Fig. 29 Evaluation of time-of-day **a**, **b** Fast-Hessian features **c**, **d** DoG features

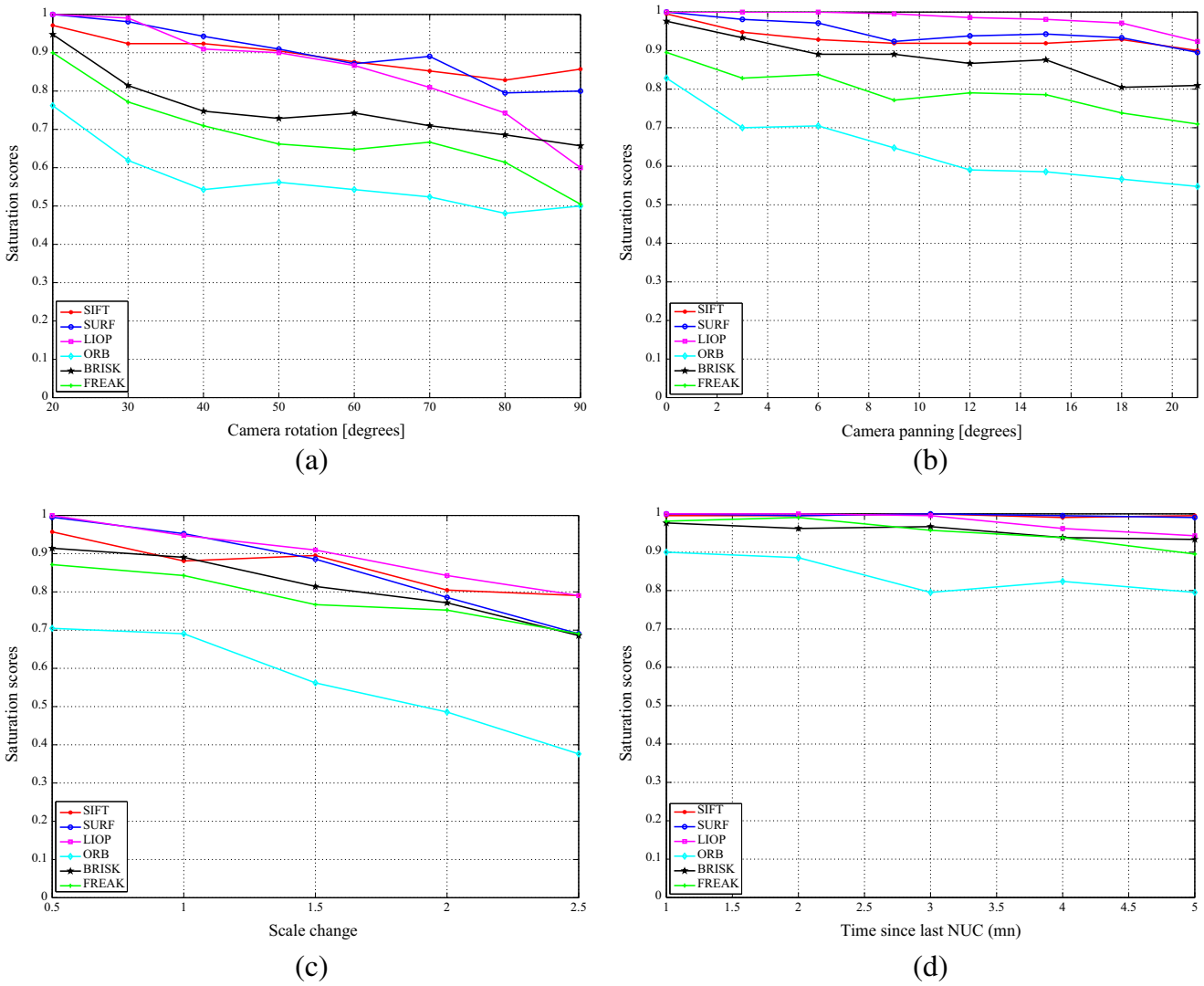


Fig. 30 Saturation scores. **a** invariance to in-plane rotation **b** invariance to camera panning **c** invariance to scale change **d** invariance to non-uniformity noise

Table 2 Feature extraction computation times

#	Detector	Time (ms)	Speed-up
1	DoG	**0.0583**	1
2	Fast-Hessian	0.0164	4
3	Harris	0.0078	7
4	GFTT	0.0064	9
5	FAST	0.0008	73
6	CenSurE	0.0048	12

Slowest and fastest times in bold and underlined, respectively

change and non-uniformity noise. We use the results obtained from the previous section. However, they are presented and interpreted differently. Indeed, we use the saturation point of the *recall*/1-*precision* curves corresponding to the considered image transforms. Also, the plots presented here include all the images of the considered sequences to illustrate the variation (or non-variation) of the descriptors' performance with increasing amounts of image transform. An ideal descriptor would obtain a horizontal line suggesting that it is not affected by the varying amount of transformation applied to the images. However, this is not the case in real-world scenes.

Figure 30 shows the descriptors saturation points for image rotation, camera panning, scale change and non-uniformity noise, respectively. Note that Fast-Hessian features were used for these tests. We can observe from Fig. 30a that most descriptors are affected when increasing the rotation angle. However, different scores were achieved. SIFT, SURF and LIOP provided the best results while FREAK and BRISK obtained similar results. ORB provided the lowest scores. SIFT stands out though as it obtained the highest score for the largest rotation while LIOP degrades as the angle increases. As concluded before, the camera panning has less effect on the performance of the detectors (Fig. 30b). LIOP seems to be particularly invariant to this image transform. SIFT and SURF behave similarly whereas ORB obtains the lowest scores in this category too. The curve of all descriptors decreases with increasing order of scale change (Fig. 30c). LIOP and SURF outperform the others for the largest scale change. From Fig. 30c, we can observe that only the binary descriptors are slightly affected when images are corrupted with non-uniformity noise (especially ORB). FREAK, BRISK and LIOP include a smoothing procedure for the computation of the descriptor. This can explain the disparity of their results compared to ORB which also uses a smoothing routine via an approximation based on integral images which seems to be less effective (than the Gaussian smoothing). SIFT and SURF appear invariant to this type of noise. They also use Gaussian smoothing which may provide this invariance.

5.5 Computation Times

Here, we show some figures relating to feature extraction and descriptor computation times for the studied algorithms. The objective is to have an insight into the computation burden of each algorithm in order to make an informed choice according to the targeted application (e.g. thermal-based visual odometry). The latter sets computational requirements on the chosen detectors/descriptors where some compromise between robustness and time consumption has to be made. Note that we used an Intel(R) Core(TM) i7-3770 CPU at 3.40 GHz. The reported execution times are only for the core task of the algorithm without the necessary operations of the benchmark (e.g. initialisation, memory allocation, etc.). Table 2 highlights the computation times required for feature extraction for the studied algorithms. DoG corresponds to the slowest detection algorithm

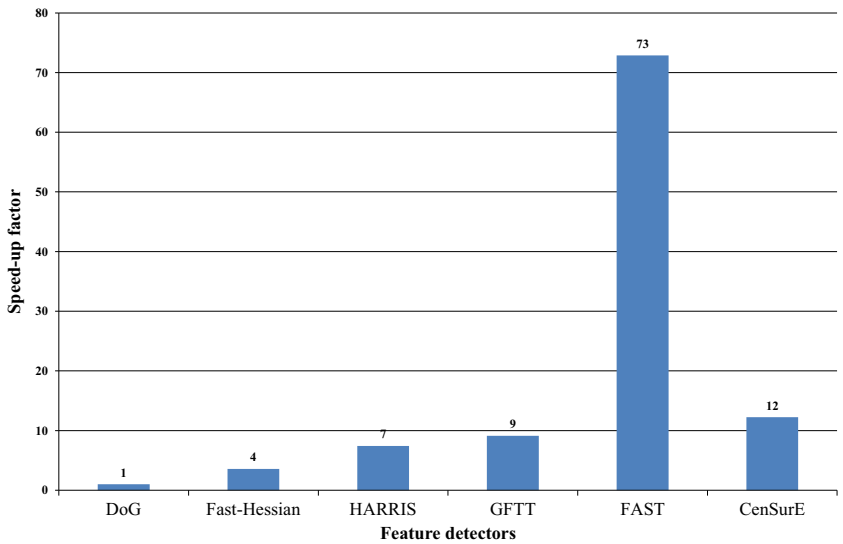


Fig. 31 Feature extraction times as a speed-up factor of DoG

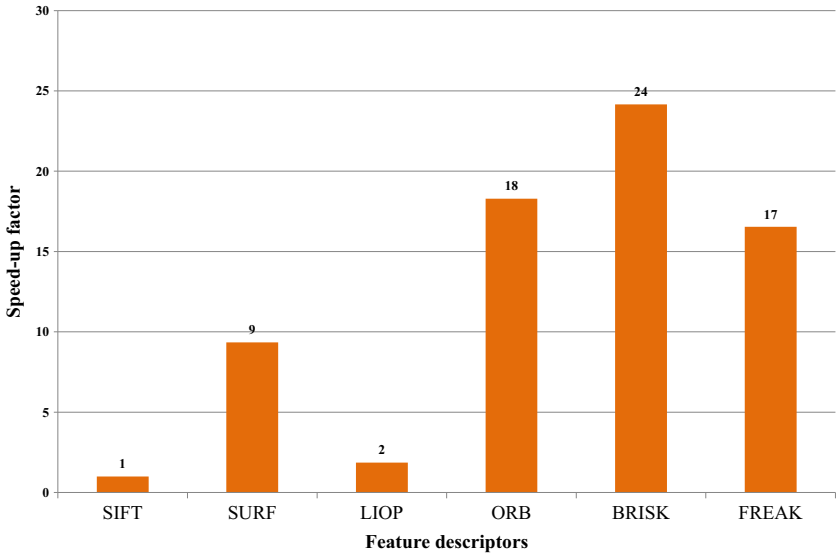


Fig. 32 Descriptors computation times as speed-up factor of SIFT

requiring, on average, 0.0583ms per feature. For more clarity, we show the performance of the other detectors in terms of speed-up factor in comparison to DoG (Fig. 31). We can clearly observe that FAST achieves the fastest extraction times being nearly two orders of magnitude faster than DoG. Naturally, GFTT requires slightly less time than Harris.

Figure 32 shows a similar plot as in Fig. 31 for the descriptors computation times (for 1 descriptor). Here also we provide the metrics in terms of speed-up with SIFT computation time as reference. Therefore, large values in Fig. 32 correspond to fast descriptor computation times. Note that SIFT required on average 0.015 ms to compute one descriptor. The actual times are highlighted in Table 3 along with the size in Bytes for each algorithm. We can observe that the binary descriptors are an order of magnitude faster than SIFT where BRISK represents the fastest description algorithm followed by FREAK and ORB. Despite having the same descriptor dimension (64 bytes), BRISK is faster than FREAK. SURF also provides decent performance in comparison to SIFT and LIOP. The latter can be categorised with SIFT as a relatively slow description algorithm.

Table 3 Descriptors computation times

#	Descriptor	Time (ms)	Speed-up
1	SIFT	**0.015**	1
2	SURF	0.0016	9
3	LIOP	0.008	2
4	ORB	0.0008	18
5	BRISK	<u>0.0006</u>	24
6	FREAK	0.0009	17

Slowest and fastest times in bold and underlined, respectively

6 Conclusion

The experiments presented in this work provided interesting insights into the performance of different feature extraction and description algorithms applied to far-infrared imagery. From repeatability to the computation times, each algorithm showed weaknesses and strengths. In order to opt for one algorithm or a combination (detector/descriptor), one needs to state the specific requirements of the targeted application. The latter help to define the various trade-offs which need to be considered in order to derive the right tools. For instance, some form of compromise between robustness and speed is mandatory to reach one's objectives.

We presented an extensive thermal dataset which encompasses various environments and image transforms (in-plane rotation, scale change, etc.). It consists of video sequences capturing real-world scenes (*office*, *container*, *ground*, etc.). The dataset can be used in different ways: (i) to evaluate the performance of feature detectors/descriptors against continuous variations of image transforms as expected in navigation applications or (ii) to study the algorithms behaviour in the presence of large image transformations (e.g. large rotation angles) which could benefit other applications such as image registration and loop closure in visual odometry (and SLAM).

We have also presented a comprehensive evaluation of feature detection and description algorithms using our dataset. The evaluation framework was divided in three main parts: first we studied the repeatability and matching scores of the detectors. Second, we investigated the descriptors' performance using *recall*/1-*precision* curves where we also provided the *precision* saturation points for each image transform. Third, we compared the execution time of all the algorithms in order to gain an insight into their computational requirements. This modular evaluation

provides insights into the weaknesses and strengths of individual detection and description algorithms. This can help improving existing algorithms to cope better with far-infrared imagery and define future research avenues to integrate the thermal modality in the field.

The evaluation resulted in a large amount of quantitative data and graphs presented in Fig. 11a–32. Although a final decision about the best detector/descriptor or combination is difficult to reach, useful insights can be gained from this evaluation: (i) blob-like detectors (especially Fast-Hessian) provide lower repeatability scores than the corner extractors (Harris, GFTT and FAST). However, the trend for matching scores is quite the opposite i.e. the blob-like detectors achieve better performance. (ii) Different *recall*/1-*precision* curves are obtained when changing the detection algorithm. (iii) Interestingly, SIFT provided higher scores with Fast-Hessian features than with its native DoG detector. (iv) BRISK and FREAK showed better performance than ORB partly thanks to the Gaussian smoothing included in their description routine, which makes them more resilient to noise. (v) Although SIFT and LIOP provided very good scores, examining their performance in terms of computation times suggests that they would not be suitable for time-constrained applications (unless optimised) e.g. visual odometry. Based on the results of the different detectors and descriptors, combining Fast-Hessian with FREAK could provide satisfactory performance for a thermal-based navigation application. This combination, as well as others, constitute a candidate to be tested against far-infrared navigation sequences which will be generated as part of our future work. We also intend to benchmark several navigation algorithms (i.e. for visual odometry) that are well-established in the visible modality against thermal navigation-oriented datasets. Finally, we intend to extend the analysis to computer vision algorithms commonly used in the field of classification/scene recognition in the infrared modality.

References

1. Agrawal, M., Konolige, K., Blas, M.R.: CenSurE: Center surround extremas for realtime feature detection and matching. In: European Conference on Computer Vision, pp. 102–115 (2008)
2. Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: Fast retina keypoint. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 510–517 (2012)
3. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
4. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. In: European Conference on Computer Vision, pp. 778–792 (2010)
5. Carlevaris-Bianco, N., Ushani, A.K., Eustice, R.M.: University of Michigan North Campus long-term vision and lidar dataset. Int. J. Robot. Res. **35**(9), 1023–1035 (2015)
6. Dahl, A.L., Aanæs, H., Pedersen, K.S.: Finding the best feature detector-descriptor combination. In: International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, pp. 318–325 (2011)
7. Evangelidis, G.D., Psarakis, E.Z.: Parametric image alignment using enhanced correlation coefficient maximization. IEEE Trans. Pattern Anal. Mach. Intell. **30**(10), 1858–1865 (2008)
8. Gauglitz, S., Höllerer, T., Turk, M.: Evaluation of interest point detectors and feature descriptors for visual tracking. Int. J. Comput. Vis. **94**(3), 335–360 (2011)
9. Harris, C., Stephens, M.: A combined corner and edge detector. In: 4th Alvey Vision Conference, pp. 147–152 (1988)
10. Krajník, T., Cristóforis, P., Kusumam, K., Neubert, P., Duckett, T.: Image features for visual teach-and-repeat navigation in changing environments. Robot. Auton. Syst. **88**, 127–141 (2017)
11. Kumar, A., Sarkar, S., Agarwal, R.P.: Correcting infrared focal plane array sensor non uniformities based upon adaptive filter. In: International Conference on Image Processing, pp. 1537–1540 (2006)
12. Lee, J.H., Kim, Y.S., Lee, D., Kang, D.G., Ra, J.B.: Robust ccd and ir image registration using gradient-based statistical information. IEEE Signal Process. Lett. **17**(4), 347–350 (2010)
13. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: Binary robust invariant scalable keypoints. In: IEEE International Conference on Computer Vision, pp. 2548–2555 (2011)
14. Lin, S.: Review: Extending visible band computer vision techniques to infrared band images. Technical Report No. MS,-CIS-01-04, pp. 1–23 (2001)
15. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
16. Mair, E., Hager, G.D., Burschka, D., Suppa, M., Hirzinger, G.: Adaptive and generic corner detection based on the accelerated segment test. In: 11th European Conference on Computer Vision, pp. 183–196 (2010)
17. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1615–1630 (2005)
18. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. Int. J. Comput. Vis. **65**(1), 43–72 (2005)
19. Moravec, H.P.: Obstacle avoidance and navigation in the real world by a seeing robot rover. PhD thesis, Stanford University, Stanford CA (1980)
20. Mouats, T., Aouf, N., Sappa, A.D., Aguilera, C., Toledo, R.: Multispectral stereo odometry. IEEE Trans. Intell. Transp. Syst. **16**(3), 1210–1224 (2015)
21. Mousa, M., Zhang, X., Claudel, C.: Flash flood detection in urban cities using ultrasonic and infrared sensors. IEEE Sensors J. **16**(19), 7204–7216 (2016)
22. Owens, K., Matthies, L.: Passive night vision sensor comparison for unmanned ground vehicle stereo vision navigation. In: IEEE International Conference on Robotics and Automation, vol. 1, pp. 122–131 (2000)
23. Ricaurte, P., Chilán, C., Aguilera-Carrasco, C.A., Vintimilla, B.X., Sappa, A.D.: Feature point descriptors: Infrared and visible spectra. Sensors (Basel Switzerland) **14**(2), 3690 (2014)

24. Rosin, P.L.: Measuring corner properties. Comput. Vis. Image Underst. **73**(2), 291–307 (1999)
25. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: European Conference on Computer Vision, pp. 430–443 (2006)
26. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In: International Conference on Computer Vision, pp. 2564–2571 (2011)
27. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. Int. J. Comput. Vis. **37**(2), 151–172 (2000)
28. Schmidt, A., Kraft, M., Kasiński, A.: An evaluation of image feature detectors. In: International Conference on Computer Vision and Graphics, pp. 251–259 (2010)
29. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
30. Sünderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., Milford, M.: On the performance of convnet features for place recognition. In: IEEE International Conference on Intelligent Robots and Systems, pp. 4297–4304 (2015)
31. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer Science & Business Media (2010)
32. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: A survey. Found. Trends Comput. Graph. Vis. **3**(3), 177–280 (2008)
33. Vidas, S., Lakemond, R., Denman, S., Fookes, C., Sridharan, S., Wark, T.: An exploration of feature detector performance in the thermal-infrared modality. In: International Conference on Digital Image Computing: Techniques and Applications, pp. 217–224 (2011)
34. Vidas, S., Lakemond, R., Denman, S., Fookes, C., Sridharan, S., Wark, T.: A mask-based approach for the geometric calibration of thermal-infrared cameras. IEEE Trans. Instrum. Meas. **61**(6), 1625–1635 (2012)
35. Wang, Z., Fan, B., Wu, F.: Local intensity order pattern for feature description. In: IEEE International Conference on Computer Vision, pp. 603–610 (2011)
36. Weinmann, M., Leitloff, J., Hoegner, L., Jutzi, B., Stilla, U., Hinz, S.: Thermal 3D mapping for object detection in dynamic scenes. ISPRS Annals of Photogrammetry Remote Sensing and Spatial Information Sciences II-1, pp 53–60 (2014)
37. Yun, J., Song, M.H.: Detecting direction of movement using pyroelectric infrared sensors. IEEE Sensors J. **14**(5), 1482–1489 (2014)
38. Zeng, D., Benilov, A., Bunin, B., Martini, R.: Long-wavelength ir imaging system to scuba diver detection in three dimensions. IEEE Sensors J. **10**(3), 760–764 (2010)
39. Zhang, L., Wu, B., Nevatia, R.: Pedestrian detection in infrared images based on local shape features. In: International Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)

Tarek Mouats received the M.Sc. degree in defense sensors and data fusion from Cranfield University, Shrivenham Campus, U.K., in 2008, and Ph.D. in 2015 with the Centre for Electronic Warfare, Information and Cyber. His research focuses on image processing, multimodal image processing, intelligent transportation systems, localization techniques, and in particular, visual odometry.

Nabil Aouf is currently a Professor with the Centre of Electronic Warfare, Information and Cyber, Cranfield University, U.K. His research interests are aerospace and defense systems, information fusion and vision systems, guidance and navigation, and tracking and control and autonomy of systems. He has authored over 100 publications of high caliber in his domains of interest. He is also an Associate Editor of the International Journal of Computational Intelligence in Control.

David Nam is a research fellow at Cranfield University. He received his B.Eng. in biomedical engineering from Carleton University, Ottawa, Canada, in 2010. He also received his Ph.D. in electrical engineering from Bristol University, in 2014. His research interests include image processing, infrared image mosaicing and medical imaging.

Stephen Vidas (M'07) received the B.Eng. degree (with first class honors) in electrical engineering as part of the Dean's Scholars program at QUT, Brisbane, Australia in 2009. He received his Ph.D. in 2013 from the same institution. His thesis is entitled: "Automatic 3D Reconstruction Beyond the Visible Spectrum".