IET Image Processing

Review Article

Performance evaluation of single and cross-dimensional feature detection and description

ISSN 1751-9659
Received on 14th November 2019
Revised 20th March 2020
Accepted on 7th April 2020
E-First on 6th July 2020
doi: 10.1049/iet-ipr.2019.1523
www.ietdl.org

Odysseas Kechagias-Stamatis[1,2] ✉, *Nabil Aouf*[1], *Mark A. Richardson*[2]
[1]*Department of Electrical and Electronic Engineering, City University of London, EC1V 0HB, London, UK*
[2]*Centre for Electronic Warfare Information and Cyber, Cranfield University Defence and Security, Shrivenham, UK*
✉ *E-mail: o.kechagiasstamatis@cranfield.ac.uk*

Abstract: Three-dimensional (3D) local feature detection and description techniques are widely used for object registration and recognition applications. Although several evaluations of 3D local feature detection and description methods have already been published, these are constrained in a single dimensional scheme, i.e. either 3D or 2D methods that are applied onto multiple projections of the 3D data. However, cross-dimensional (mixed 2D and 3D) feature detection and description are yet to be investigated. Here, the authors evaluated the performance of both single and cross-dimensional feature detection and description methods on several 3D data sets and demonstrated the superiority of cross-dimensional over single-dimensional schemes.

1 Introduction

Local features in three-dimensional (3D) data have been widely investigated to improve the distinctiveness and robustness of local feature (keypoint) detection and description methods. Given the importance of these methods for 3D data registration and classification applications, it is necessary to evaluate keypoint detectors and feature descriptors. Most such evaluations have been presented in the context of reports comparing current methods to newly proposed techniques, although some studies dedicated to the evaluation of 3D keypoint detectors or feature descriptors have also been published [1]. However, such evaluations have been limited to a single domain, with 3D methods applied directly to 3D data [2–5], or 2D methods applied to multiple 2D projections of 3D data [6, 7].

Examples of the direct 3D approach include the evaluation of several 3D keypoint detectors by comparing the robustness of each technique to rotation, scaling and translation [8], an evaluation focusing on the optimum combination of 3D keypoint detection and feature description [9], and a complete and thorough evaluation of 3D keypoint detectors, with further limited evaluation carried out of selected 3D descriptors [1]. The most comprehensive studies of 3D feature descriptors reported thus far also included work involving a selection of 3D keypoint detectors [2, 8, 10]. These latter reports represent the most comprehensive evaluations of 3D keypoint detection [1] and description methods [2, 10] published thus far.

In an example of the indirect approach (2D schemes applied to 3D data, where the 3D data are presented in a 2D range image form), state-of-the-art 2D descriptors have been evaluated via several transformations of the initial 2D range image, including maximum curvature, mean curvature and shape index (ShI) [11]. The authors found that scale-invariant feature transform (SIFT) [12] achieved the best performance in terms of facial recognition, whereas fast retina keypoint (FREAK) [13] achieved the best trade-off between performance and speed. The evaluation of 2D methods on projections of 3D data in point cloud form has also been attempted, but only in the context of comparing current methods to newly proposed techniques [7].

Currently, the performance of 2D and 3D schemes has been evaluated independently without cross-dimensional keypoint detection and feature description or the direct comparison of pure 3D and 2D schemes. Cross-dimensional evaluation has not been attempted yet and refers to challenging both 2D and 3D local keypoint detection and feature description methods against 3D data using a cross-dimensional approach. Currently, the comparison of 3D and 2D schemes has been superficially addressed in the context of comparing a proposed technique against 3D methods [14]. Hence, driven by the absence of such comparisons, we therefore evaluated both single and cross-dimensional keypoint detection and feature description, i.e. 2D–2D, 3D–3D, 2D–3D and 3D–2D keypoint detection and feature description data domain combinations, on several 3D point cloud data sets varying in content and complexity. The aim of the study was to identify potential cross-modality combinations that exploit the advantages of both 2D and 3D methods in terms of robustness and computational efficiency, with performance and computational requirements given equal priority. It should be noted that despite single-modality keypoint detection and feature description on pure 3D and 2D data (not originating from projections) has already been presented in [2, 15–17], to make a direct comparison between single and cross-modality comparison feasible, it is necessary to re-evaluate these keypoint detection and feature description methods on the same data set used in this paper.

The contributions of this study, which is a thorough expansion of our pre-print [18], can be summarised to

(i) We extend the evaluation of current keypoint detection and feature description methods on 3D point cloud data by exceeding the typical single data modality constraint (either 2D or 3D methods) and adopt a novel cross-dimensional (mixed 2D and 3D) scheme. This cross-modality evaluation has not yet been presented in the current literature.

(ii) Our novel cross-dimensional evaluation demonstrates that 2D keypoint detectors present higher repeatability rates, are more robust to nuisances such as resolution variation and noise and are faster to compute compared to their 3D counterparts.

(iii) Regarding the optimum keypoint detector and feature descriptor combination, our cross-dimensional evaluation scheme revealed that overall a multi-dimensional solution combining the 3D keypoint detector intrinsic shape signatures (ISSs) or uniform subsampling, with the 2D feature descriptor named speeded-up robust features (SURFs), are the optimum combinations incorporating the advantages of each individual data modality and affording high quality correspondences at a low computational cost. Additionally, our trials demonstrated that overall a cross-dimensional 3D keypoint detection and 2D feature description combination is more appealing than a typical single dimensional

3D solution affording twice the performance and a 54× computational speedup.

The remainder of the article is organised as follows: Section 2 presents the 2D and 3D keypoint detectors and feature descriptors that were evaluated. Section 3 presents the experimental setup and Section 4 evaluates the 2D and 3D techniques in single and cross-dimensional schemes. Our conclusions are presented in Section 5.

2 2D/3D keypoint detection and feature description methods

2.1 Keypoint detectors

Keypoint detectors analyse the structure around a vertex or a pixel depending on the data domain (3D or 2D, respectively) and classify as keypoints the vertices/pixels that fulfil some specific criteria that are dependent on the detector itself. Ideally, keypoints are prominent among their surroundings, have unique features, and can be redetected even if the object to which they belong is distorted or corrupted.

2.1.1 2D detectors: *Harris*: Harris is a fixed scale corner detector [19], which relies on an autocorrelation function that captures the intensity variations of an image I in a neighbourhood window Q centred at pixel $p(x, y)$

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

where (x, y) are the pixel coordinates in I and $w(u, v)$ is the window patch at position (u, v). Using Taylor's approximation, Harris rearranges (1) as follows:

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

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

where I_u, I_v represent the spatial gradients of the image.

The shape of Q is classified based on the eigenvalues λ_1 and λ_2 of M. Specifically, if both values are small, E also has a small value and Q has an approximately constant intensity. If both are large, E has a sharp peak indicating that Q includes a corner, and if $\lambda_1 > \lambda_2$ then Q includes an edge. To measure the corner or edge quality, Harris introduced metric R

$$R(x, y) = \det(M) - k \cdot \mathrm{tr}(M) = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2) \tag{4}$$

where $k \in [0.04, \ldots, 0.15]$.

Good features to track (GFTT): Shi and Tomasi [20] extended the robustness of the Harris corner detector by proposing that Q encloses a corner if $\min(\lambda_1, \lambda_2) > \lambda$, where λ is a predefined threshold. GFTT, like Harris, is a fixed scale detector.

Difference of Gaussians (DoG): Lowe [12] proposed the SIFT keypoint detection and description scheme. For the keypoint detection part, Lowe detected local extrema in image I utilising a DoG scheme aiming to reduce the overall processing burden during keypoint detection.

For the DoG scheme, a pyramid of images is created to achieve scale invariance by convolving I with Gaussian kernels at various scales. The output of two sequential convolutions is subtracted creating a new set of images, i.e. DoG images, in which pixels are classified as candidate keypoints. Then the pixel value of each candidate keypoint is compared with its eight neighbours in the same scale, the nine pixels one scale above and the nine pixels one scale below. If the pixel value of a candidate keypoint has the highest value within its neighbourhood then it is labelled as a keypoint. The latter comparison is the popular *non-maxima suppression* process. Finally, the keypoint detection stage ends with a refinement process to discard keypoints that have a low contrast and that lie on edges. The former are discarded by applying a texture threshold, whereas the latter are discarded by identifying Harris keypoints [19]. DoG is an adaptive scale keypoint detector.

Fast Hessian (FH): a processing-efficient alternative to DoG is the FH detector used as the keypoint detection part of the popular SURFs algorithm [21]. In order to avoid convolution with second-order derivatives, this technique approximates the Gaussian kernels with their discretised version (i.e. box filters) that are computed with a constant time cost by utilising the integral image concept [22]. Like the DoG detector, candidate features are obtained after a 3 × 3 × 3 neighbourhood non-maximum suppression process. Finally, candidate keypoints with a response R exceeding a pre-defined threshold are preserved while the rest are discarded

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

where $D_{xx}(\sigma)$, $D_{yy}(\sigma)$ and $D_{xy}(\sigma)$ are the outputs after convolving the corresponding box filters of standard deviation σ with image I. FH is an adaptive scale keypoint detector.

Features from accelerated segment test (FAST): FAST [23] detects keypoints in an image I by placing around the pixel of interest p a circle that has a circumference of 16 pixels. If I_p is the pixel intensity at pixel p and thresh is the pre-defined threshold, then p is labelled as a keypoint if *N–contiguous* pixels in the circle are brighter than I_p + thresh or darker than I_p−thresh. FAST is a fixed scale keypoint detector.

Binary robust invariant scalable keypoints (BRISK): the BRISK technique [24] involves both a keypoint detection and a description scheme. For the former, it uses the FAST [23] keypoint detector in 9–16 mask configuration, i.e. placing around the pixel of interest p a circle that has a circumference of 16 pixels, and considering the intensity of nine contiguous pixels within that circle. In BRISK, the FAST technique is combined with maxima suppression applied in a scale-space fashion using the FAST score as a measure of saliency. However, in contrast to DoG and SURF, keypoints are sought within a continuous scale-space by involving not only the true octaves but also virtual intra-octave levels.

KAZE: KAZE [25] is similar to SURF in that it relies on the response of a scale-normalised determinant of the Hessian at multiple scale levels, but it involves a non-linear scale-space rather than the linear scale-space used in SURF. KAZE is an adaptive scale keypoint detector.

2.1.2 3D detectors: *Intrinsic shape signatures*: ISS [26] measures the saliency of a point $p(x, y, z)$ based on the eigenvalue decomposition of the scatter matrix $\Sigma(p)$ of the N vertices within the support region (neighbourhood) $\boldsymbol{V}$ of p

$$\Sigma(p) = \frac{1}{N}\sum_{q \in V}\left(q - \mu_p\right)\left(q - \mu_p\right)^T \tag{6}$$

$$\mu_p = \frac{1}{N}\sum_{q \in V} q \tag{7}$$

ISS suggests that vertices fulfiling (8) are labelled as candidate keypoints

$$\frac{\lambda_2(p)}{\lambda_1(p)} < \mathrm{threshold}_1 \wedge \frac{\lambda_3(p)}{\lambda_2(p)} < \mathrm{threshold}_2 \tag{8}$$

where the λ_1, λ_2, λ_3 are the eigenvalues of $\Sigma(p)$ in order of decreasing magnitude. Finally, candidate keypoints with the smallest eigenvalues and large variation along each principal direction are labelled as ISS keypoints. ISS is a fixed scale keypoint detector.

KeyPoint quality (KPQ): KPQ is a keypoint detector that ranks candidate keypoints based on a quality metric [27]. Specifically, $\boldsymbol{V}$ is aligned to the canonical reference frame given by the principal directions, and then non-distinctive vertices are discarded by thresholding the ratio between the maximum lengths along the first two principal axes. The remaining vertices p are labelled as candidate keypoints, which are then evaluated for their saliency $\rho(p)$ with respect to a local sampling surface S that utilises a uniform sampling grid and is fit to the remaining vertices within $\boldsymbol{V}$

$$\begin{aligned}\rho(p) \doteq \frac{1000}{N^2} \sum_{q \in V} |K(p)| + \max_{q \in V}(100K(p)) \\ + \min_{q \in V}(100K(p)) + \max_{q \in V}(10k_1) + \min_{q \in V}(10k_2)\end{aligned} \tag{9}$$

where K is the Gaussian curvature and k_1, k_2 are the principal curvatures. Given that positive and negative curvature values are equally descriptive, (9) considers absolute curvature values so that positive and negative curvatures do not cancel each other. The constant multiplicative terms are empirically chosen aiming at giving the appropriate weight to each term [27]. Sensitivity to noise and sampling is reduced by estimating k_1 and k_2 after the sampling surface S is fitted to the vertices within $\boldsymbol{V}$.

Vertices with a $\rho(p)$ value exceeding a threshold and fulfiling certain constraints are labelled as KPQ keypoints. These constraints are (i) the minimum Euclidean distance between two KPQ keypoints is greater than a certain threshold and (ii) that within a support radius only one KPQ can exist. KPQ is a fixed scale keypoint detector.

Harris 3D: although 2D and 3D Harris [19] are conceptually similar, the modification required for extension to a 3D keypoint detector involves substituting the image gradients in the covariance matrix of (3) with the normal vector of the support region $\boldsymbol{V}$ centred on vertex $p(x, y, z)$ of the point cloud. Harris 3D is a fixed scale keypoint detector.

Local surface patches (LSP): LSP [28] uses the ShI metric to measure the saliency of vertex $p(x, y, z)$. Vertices p that fulfil the following constraint are considered as candidate LSP keypoints

$$\mathrm{ShI}(p) \geq (1 + a)\mu_{\mathrm{ShI}(p)} \vee \mathrm{ShI}(p) \leq (1 - \beta)\mu_{\mathrm{ShI}(p)} \tag{10}$$

where $\mu_{\mathrm{ShI}(p)}$ is the average ShI of the support region $\boldsymbol{V}$ and α, β are user-defined thresholds. Candidate LSP keypoints then undergo a non-maxima suppression process and the remaining vertices are classified as LSP keypoints. LSP is a fixed scale keypoint detector.

Heat kernel signature (HKS): HKS [29] is a saliency metric based on the restriction of the heat kernel to the temporal domain that is computed on the mesh $\boldsymbol{M}$ of the point cloud. Vertex $p(x, y, z)$ is defined as an HKS keypoint if its saliency $k_{t'}$ at time interval t' fulfils the following constraint:

$$k_{t'}(p, p) > k_{t'}(q, q) \tag{11}$$

where q is a vertex belonging to a two-ring neighbourhood of p and $k_{t'}(p, q)$ is a function that represents the amount of heat transferred from vertex p to q in time t' given a unit heat source at vertex p. Thus, $k_{t'}(p, q)$ is governed by the heat equation:

$$\Delta_M u(x, t) = -\frac{\partial u(x, t)}{\partial t} \tag{12}$$

where $\Delta_M u(x, t)$ is the Laplace-Beltrami operator defined on manifold $\boldsymbol{M}$. HKS is a fixed scale keypoint detector.

Laplace–Beltrami scale space (LBSS): Unnikrishnan and Hebert [30] classify a vertex $p(x, y, z)$ as a keypoint if its scale-space saliency $\rho(p, t)$ exceeds a certain threshold

$$\rho(p, t) = \frac{2\| p - A(p, t) \|}{t} \mathrm{e}^{-2\| p - A(p, t) \|/t} \tag{13}$$

$$A(p, t) = p + \frac{t^2}{2}\Delta_M p \tag{14}$$

where Δ_M is the Laplace-Beltrami operator. In simpler terms, $\rho(p, t)$ can be considered as a displacement of p along its normal that is proportional to the mean curvature. LBSS is an adaptive scale keypoint detector and scale-space is implemented by increasing the size of the support region $\boldsymbol{V}$.

MeshDoG: MeshDoG [31] is a similar solution to LBSS but scale-space is created using the DoG concept [12]. MeshDoG is applied on a transformed representation of the point cloud, where for the context of this paper we use the mean curvature [1]. The scale-space saliency $\rho(p, t)$ of a vertex $p(x, y, z)$ is defined as

$$\rho(p, t) = C_H^{(t)}(p) - C_H^{(t-1)}(p) \tag{15}$$

$$C_H^{(t)} = C_H^{(t-1)} \times G(\sigma) \tag{16}$$

where $C_H^{(t)}$ is the tth convolution of the mean curvature map C_H with the Gaussian kernel of zero mean and σ standard deviation. MeshDoG is an adaptive scale keypoint detector.

Salient points (SPs): SP [32] is similar to MeshDoG [31] but is directly applied to the vertex coordinates rather than a transformed representation of the point cloud. SP is an adaptive scale keypoint detector.

KPQ-AS: this is an extension of the KPQ technique [27] that facilitates adaptive-scale keypoint detection. Scale-space is created by increasing the support region $\boldsymbol{V}$ and scale selection is achieved by performing non-maxima suppression.

2.2 Local feature descriptors

Local feature description techniques describe local patches around a point of interest by encoding the properties of the local patch. Ideally, feature descriptors describe each keypoint in a unique manner and are robust to nuisance factors such as resolution variation and noise.

2.2.1 2D descriptors:

Scale-invariant feature transform: Lowe [12] describes a keypoint detection method but also suggests a feature description technique. The latter initially assigns to each keypoint one or multiple orientations that are based on the local gradient information. The magnitude and direction of the gradient form an orientation histogram with 36 bins based on the neighbourhood of the keypoint. The histogram is then weighted by a Gaussian kernel that is placed around the keypoint and the peak of the histogram corresponds to the orientation of the keypoint. In the event this histogram has peaks of at least 80% of the main peak, then additional descriptions of the same keypoint are created that share the same scale but have different orientations.

The scale and orientation linked to each keypoint form a local coordinate frame. Specifically, the descriptor is computed using the gradient magnitude and orientations in a 16 × 16 window around the keypoint (rotated according to orientation). These are stacked in 8-bin histograms formed in 4 × 4 sub-regions and are weighted by a Gaussian window.

Speeded-up robust feature: SURF [21] initially performs an orientation assignment by computing Gaussian-weighted Haar wavelet responses over a circular region with a radius six times the scale where the keypoint is detected. Once an orientation is assigned, the description process involves a square region (20 × scale) centred on the keypoint and oriented accordingly. This region is further divided into 4 × 4 sub-regions and then vertical and horizontal Haar-wavelet responses are computed, which are weighted with a Gaussian kernel. This process is performed at fixed sample points and is summed up in each sub-region. Finally, the polarity of intensity changes is also calculated by summing the absolute values of the horizontal and vertical responses. SURF features of opposing polarity are not matched.

Binary robust invariant scalable keypoint: The BRISK method [24] encodes keypoints using a handcrafted sampling pattern comprising concentric circular patches centred at a keypoint. Aliasing effects during sampling are avoided by applying local Gaussian smoothing on the patch to be described, with a standard deviation proportional to the distance between the circle centre and the keypoint.

There are two types of sampling pairs (short and long pairs) that depend on the distance between them. The long pairs have a distance greater than threshold $d_{\min}$, and are used to compute the local gradient (of the patch) that defines the orientation of the feature. The short pairs with a distance less than threshold $d_{\max}$ are then rotated accordingly to achieve rotation invariance and are used to compute the binary BRISK descriptor via intensity tests.

Fast retina keypoint: FREAK [13] is a biologically inspired binary keypoint descriptor that applies a series of intensity tests on a patch that encloses the keypoint. FREAK and BRISK share the same sampling pattern and use the same mechanism to estimate the keypoint orientation. However, FREAK is influenced by the human retinal system and exploits a circular sampling grid with sampling points that are denser near the centre and become exponentially less dense further away from the centre. The advantage of this concept is that the test pairs naturally form a coarse-to-fine approach. Feature matching is accelerated by comparing the coarse part of the descriptor and if these exceed a threshold then the fine part is tested.

KAZE: The keypoint description part of KAZE [25] is similar to SURF but is properly adapted to facilitate a non-linear scale-space framework.

For a recent review on 2D keypoint detectors and descriptors the reader is referred to [16, 17].

2.2.2 3D descriptors: The 3D local feature description techniques comprise a support volume $\boldsymbol{V}$ that in centred on a keypoint $p(x, y, z)$ by encoding the geometric properties and the underlying structure of $\boldsymbol{V}$ [33]. Their major advantages include robust feature description for partially visible objects [34] and lower susceptibility to illumination variation and pose changes [35]. The 3D descriptors evaluated herein are described below. However, because we attributed equal importance to performance and processing efficiency, we did not evaluate 3D shape context [36] and its extension the unique shape context [37] due to their high computational burden.

Histogram of distances (HoDs)/HoD-short (HoD-S): HoD [38] is a robust and processing-efficient 3D descriptor that calculates the probability mass density of the normalised point-pair L_2-norm distance distributions within $\boldsymbol{V}$. L_2-norm distances are encoded in a coarse and a fine manner by using different bin sizes during distance quantisation. Finally, the two types of encodings are concatenated in a single descriptor. This dual encoding scheme enhances feature-matching performance in the presence of noise and subsampling perturbations. HoD does not require a local reference frame (LRF) or axis (LRA) and adapts the description radius on the target point cloud resolution rather than the template, which is the norm for a 3D descriptor. HoD-S [39, 40] is a compact version of HoD that exploits only on the coarse part of HoD.

Signatures of histograms of orientations (SHOT): SHOT [41] divides the support volume $\boldsymbol{V}$ into a number of sub-volumes along the azimuth, the elevation and the radius. For each sub-volume, a 1D histogram is computed based on the normal variation between the keypoint $p(x, y, z)$ (including its surrounding vertices) and the vertices that lie in each sub-volume.

Fast point feature histograms (FPFHs): FPFH [42] establishes on $\boldsymbol{V}$ a *Darboux* LRF. Then for each point belonging to $\boldsymbol{V}$, FPFH encodes the angular relationship between the keypoint $p(x, y, z)$ and its neighbours as provided by the LRF. Finally, this angular relationship is transformed into a histogram.

Rotational projection statistics (RoPS): RoPS [43] establishes on $\boldsymbol{V}$ a LRF, then $\boldsymbol{V}$ is rotated around every axis of the LRF and is projected on each of the coordinate planes. Finally, each projection undergoes a statistical analysis based on low-order moments and entropy, which are converted into a 1D histogram.

Tri-spin images (TriSI): TriSI [44] is an extension of the popular 3D descriptor spin images (SIs) [45]. For the latter, given a support volume $\boldsymbol{V}$ centred at point $p(x, y, z)$, a LRA is aligned with the normal vector of the vertices within $\boldsymbol{V}$, a 2D array accumulator with user-defined dimensions is placed on the LRA, and the SI descriptor is generated by accumulating the neighbouring points into each bin of the 2D array as the array *spins* around the LRA. TriSI uses the same technique as SI but substitutes the LRA with an LRF and calculates a SI value for each axis of the LRF. Finally, the three SI values are concatenated to from a TriSI descriptor.

Recently, Zhao *et al.* [46] proposed the statistic of deviation angles on sub-divided space (SDASS), which encodes the geometrical and spatial information within $\boldsymbol{V}$. For the geometrical encoding part, SDASS calculates the angular deviation between a typical sized LRA against an extended radius LRA. Additionally, SDASS encodes the spatial information by dividing $\boldsymbol{V}$ along the LRA axis, project the vertices within $\boldsymbol{V}$ against the radial direction and calculate in each subspace the angular deviation between the typical and the oversised LRAs. Zhou *et al.* [47] encode the salient feature information within $\boldsymbol{V}$ by proposing the Histograms of gaussian normal distribution (HGND) descriptor. Specifically, the vertices in $\boldsymbol{V}$ are projected on the planes of a LRF centred at $\boldsymbol{V}$ and each projection is divided in four equally sized quadrants. Then, HGND calculates the Gaussian point distribution and the normal distribution within each quadrant and finally forms a 1D histogram to represent the feature descriptor. A similar approach to RoPS is the multi-view depth (MVD) descriptor [48]. Both share an LRF estimation process based on an eigen-analysis of the weighted point scatter matrix within $\boldsymbol{V}$ and on a feature calculation process by projecting $\boldsymbol{V}$ on the planes of the LRF. However, the two main differences between these descriptors are, (i) RoPS requires mesh information for the LRF estimation, while MVD can be directly applied on the point cloud and (ii), during the feature calculation process RoPS creates a quantitative distribution matrix for each projection of $\boldsymbol{V}$, while MVD creates a local depth distribution matrix. Lim and Lee [49] extend the 2D SIFT descriptor to be applicable on a 3D mesh and exploit the gradients of the scalar functions defined on $\boldsymbol{V}$ by convolving the point cloud with Gaussian kernels. Then adjacent Gaussian functions are subtracted to produce the DoG functions and this procedure repeats with down-sampled Gaussian functions in the next octave. Lin *et al.* [50] suggest a binary variant of the SHOT descriptor by utilising a Gray-code encrypting scheme, while [51] proposes a binary variant of the HoD descriptor. In [52], the authors propose a deep-learning based solution that directly processes unstructured 3D point clouds and learns a permutation invariant representation of the 3D vertices, while in [53] the authors utilise a deep network to directly match 2D with 3D features. For a systematic review on current feature descriptors the reader is referred to [54]. For completeness it is worth mentioning that a thorough evaluation of current LRF techniques is presented in [55].

It should be noted that despite literature offers recent keypoint detection and feature description techniques (especially for the 3D descriptors), in this work we focus our evaluation on state-of-the-art techniques with an open source code, ensuring the high quality performance of method as presented in the corresponding literature.

3 Experimental setup

3.1 Data sets

We challenged the effectiveness and the robustness of each keypoint detector and feature descriptor by evaluating their performance on the Oakland data set [56], the Laser Scanner data set [57], the Kinect data set [41] and the SpaceTime data set [41].

3.1.1 Oakland data set: The Oakland data set comprises 18 point cloud scenes of the Oakland University campus captured using a LIDAR device. For our point cloud registration scenario, we exploit two consecutive scenes that have some overlap. Then one of the two scenes was randomly rotated (pitch, roll and yaw) by up to 180° and simultaneously translated in the X, Y and Z directions by up to 10 m.

3.1.2 Laser scanner data set: The Laser Scanner data set is the most cited data set in the 3D computer vision literature. It comprises five model point clouds and 50 scene point clouds of high quality. Each model comprises a full 3D point cloud, whereas the scenes are 2.5D point clouds (i.e. viewing-dependent point clouds based on a specific vantage point). Scenes also contain clutter objects and the target is occluded.

3.1.3 Kinect data set: The Kinect data set comprises six models and 16 scenes acquired by a Microsoft Kinect sensor. Given the sensing device, the point cloud quality is low, and the models within scenes are occluded and mixed with clutter objects. In contrast to the Laser Scanner data set, the models and scenes in the Kinect data set share the same dimensionality (2.5D).

3.1.4 SpaceTime: The SpaceTime data set [41] was created by using the SpaceTime Stereo technique and comprises eight models and 15 scenes. Given the use of this technique, the point clouds are of medium quality. Each scene encloses the target object which is cluttered and occluded.

3.2 Evaluation

Given a model and a target point cloud, the first part of our evaluation involved challenging the 3D keypoint detection methods against the 2D methods. For the former, we applied the 3D keypoint detectors presented in Section 2 to both the Model $\boldsymbol{M}$ and the Scene $\boldsymbol{S}$. As previously reported [1], we avoided the influence of border vertices on the keypoint detection process by discarding border keypoints. Then, given the known homography between $\boldsymbol{M}$ and $\boldsymbol{S}$, we calculated a number of performance metrics for each keypoint detector (Section 3.3).

The 3D techniques were applied directly to the point cloud data, whereas for the 2D techniques we initially projected each point cloud onto the main planes of the *XYZ* global reference frame that was fitted to the point cloud during acquisition. Then on each projection, we applied the 2D keypoint detectors presented in Section 2. Finally, we back projected the detected 2D keypoints of each projection to the initial point cloud and calculated the performance metrics used for the 3D keypoint detectors.

During the 3D to multi-2D remapping process, we properly quantised the coordinates of each vertex to remap the 3D floating-point vertex coordinates $p(x, y, z)$ into integer coordinates $p_Q(x, y, z)$, which are then projected on the XYZ reference planes

$$p_Q(x, y, z) = \lfloor q_{\mathrm{f}} \cdot p(x, y, z) \rfloor \tag{17}$$

where q_{f} is the quantisation factor and $\lfloor \cdot \rfloor$ the bottom-round process. Selecting q_{f} is not trivial as it highly affects the amount of details of the multi-2D projections and thus the performance of the keypoint detection and description methods. In fact, large q_{f} values create large 2D projections increasing the total computational time and the memory requirements of the processing platform in such an extent that the fast to compute (in general) 2D keypoint detection and description methods may impose an overall larger processing burden compared to their 3D counterparts. On the contrary, small q_{f} values discard point cloud topology information during the 3D to 2D remapping process by subsampling the projected data. Once the 2D keypoints are detected, their 2D pixel coordinates are remapped into the XYZ global reference frame where the entire point cloud is placed, creating the $p_{3\mathrm{D}}(x, y, z)$ keypoint coordinates. Due to the multi-2D to 3D back-projection process, which includes the inverse process of (17), a complete match between $p_{3\mathrm{D}}$ and a vertex p of the point cloud is not trivial, and therefore we associate $p_{3\mathrm{D}}$ with the closest point cloud vertex

$$p_{3\mathrm{D}} \sim \underset{p}{\arg\min} \left(\| p_{3\mathrm{D}} - p \|_2 \right) \tag{18}$$

The overall keypoint evaluation pipeline is presented in Fig. 1*a*.

The second part of our evaluation compared single and cross-dimensional keypoint detection and feature description. Specifically, we evaluated the performance of 3D keypoint detection and description (3D–3D), 2D keypoint detection and description (2D–2D), 3D keypoint detection with 2D description (3D–2D), and 2D keypoint detection with 3D description (2D–3D). We assessed only the top-performing 2D and 3D keypoint detectors based on the results of the first part of our evaluation. As described for the first stage, we applied each 2D descriptor to the projected planes of the *XYZ* global reference frame that was fitted to the point cloud during acquisition. During 2D feature matching, we cross-matched all features from every plane of the model and the target point clouds and created a list of corresponding pixels, which were back-projected into the original 3D domain. The performance metrics for each keypoint detector and feature descriptor combination (Section 3.3) were used for both single and cross-dimensional keypoint detection. Fig. 1*b* shows the architecture for all single (2D–2D, 3D–3D) and cross-dimensional (2D–3D, 3D–2D) keypoint detection and description methods, which is applied on both $\boldsymbol{M}$ and $\boldsymbol{S}$ point clouds in order to perform single and cross-dimensional feature matching depending on the pipeline evaluated. Fig. 1*c* presents the complete cross/single dimensional evaluation architecture including the feature matching and the model – scene correspondence estimation stages. It is worth noting that we intentionally did not use any correspondence grouping scheme to discard false correspondences as we purely focus on the capabilities of each cross and single dimensional keypoint detection and feature description technique. However, for completeness on current corresponding grouping techniques, the reader is referred to [58].

3.3 Comparison metrics

3.3.1 Absolute/relative repeatability (AR/RR): Repeatability is the most important metric for a keypoint detector because it defines its ability to find the same keypoints on different instances of a given 3D point cloud or 2D image. For the 2D and 3D detectors evaluated in this study, we extracted a keypoint k_M from the model $\boldsymbol{M}$ (either a 3D point cloud or a 2D image projection depending on the evaluation) and transformed it into k_{MS} according to the homography, i.e. rotation R and translation T, between the model $\boldsymbol{M}$ and the scene $\boldsymbol{S}$. A keypoint k_M is repeatable if the Euclidean distance of k_{MS} from its nearest keypoint k_S that is extracted from the scene $\boldsymbol{S}$ is less than a threshold ε

$$\| Rk_M + T - k_S \| = \| k_{MS} - k_S \| < \varepsilon \tag{19}$$

The AR and the RR [2, 59] are defined as

$$\mathrm{AR} = C^{+} \tag{20}$$

$$\mathrm{RR} = \frac{C^{+}}{C} \tag{21}$$

where C^{+} is the number of keypoints that fulfil (19) and C is the number of detected keypoints in the model scene. Model keypoints k_M were only considered if they were present in the scene. We therefore checked whether a real vertex existed within a small neighbourhood of the fictitious k_{MT}created by the known model-scene homography. If this was true, the real vertex closest to the fictitious k_{MT} was linked with k_M. This neighbourhood is defined by a sphere centred at k_{MT} with a radius of $10 \times \mathrm{Tr}$, with Tr the average scene point cloud resolution.

In contrast to previous studies [1, 2, 45], we set a larger radius in order to achieve a common neighbourhood size for all tests and also to compensate for transformation errors that occur when the 3D point cloud vertices are projected onto multiple 2D images and are then back-projected to a 3D point cloud.

3.3.2 Area under curve (AUC): The AUC metric is a single value that indicates the overall performance of the descriptor. Here, we calculated the AUC based on the 1-precision–recall (PR) curve [2]. Given a scene feature f_{s} that encodes the keypoint k_S, a list of model features and the model-to-scene ground truth homography, f_{s} is matched against all model features to find the closest. If the Euclidean distance of the keypoints that have matched features is less than a threshold μ then the match is considered as a true positive (TP) otherwise as a false positive (FP). Features that are

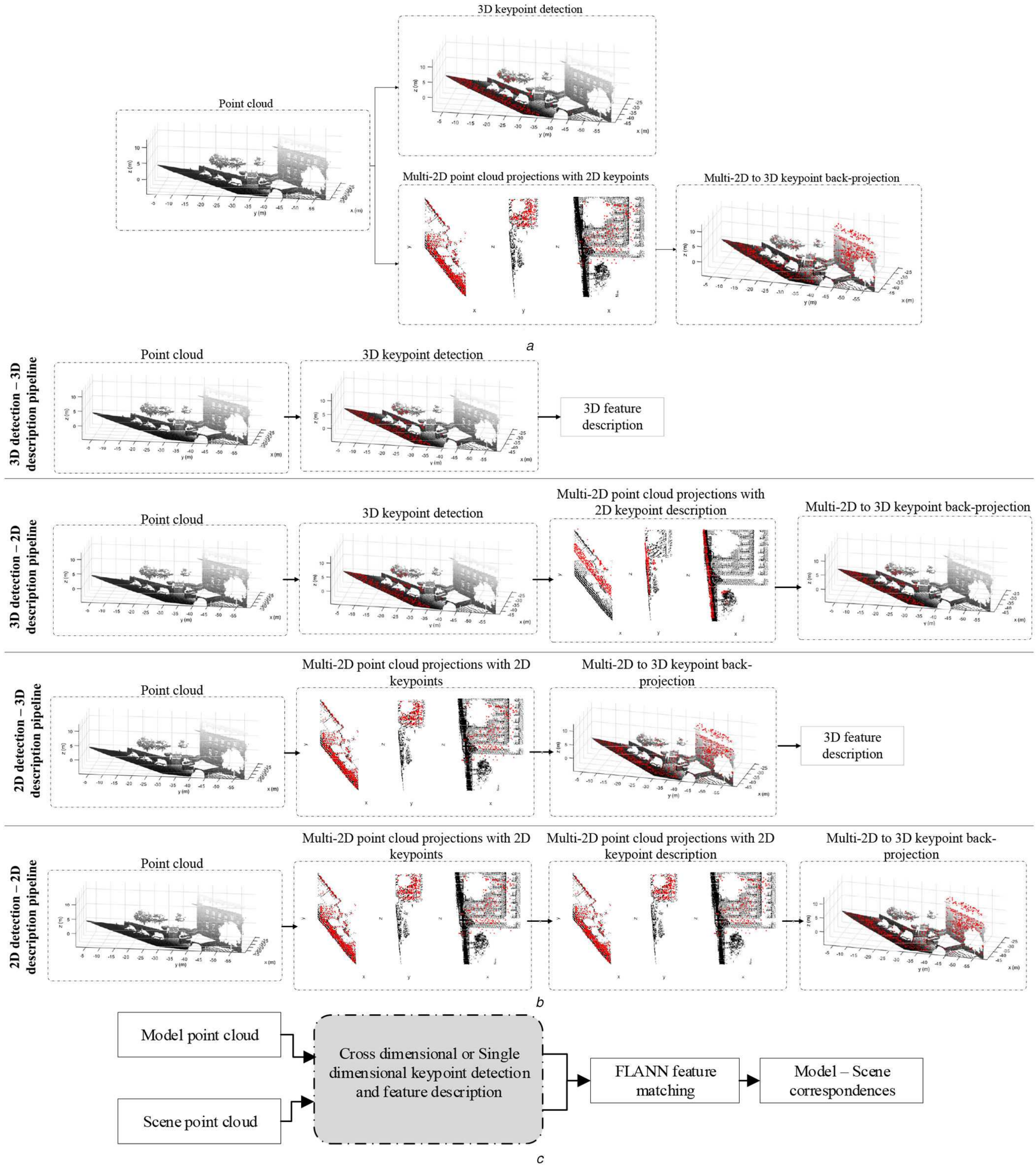

3D keypoint detection
Point cloud
Multi-2D point cloud projections with 2D keypoints
Multi-2D to 3D keypoint back-projection
a
3D detection – 3D description pipeline
Point cloud
3D keypoint detection
3D feature description
3D detection – 2D description pipeline
Point cloud
3D keypoint detection
Multi-2D point cloud projections with 2D keypoint description
Multi-2D to 3D keypoint back-projection
2D detection – 3D description pipeline
Point cloud
Multi-2D point cloud projections with 2D keypoints
Multi-2D to 3D keypoint back-projection
3D feature description
2D detection – 2D description pipeline
Point cloud
Multi-2D point cloud projections with 2D keypoints
Multi-2D point cloud projections with 2D keypoint description
Multi-2D to 3D keypoint back-projection
b
Model point cloud
Scene point cloud
Cross dimensional or Single dimensional keypoint detection and feature description
FLANN feature matching
Model – Scene correspondences
c
z (m)
y (m)
x (m)

Fig. 1 *Single and cross dimensional feature detection and description architecture*
(a) 2D/3D keypoint detection pipeline, ***(b)*** 2D/3D keypoint detection and description pipeline, ***(c)*** Cross/single dimensional evaluation pipeline

incorrectly not matched are labelled as a false negative (FN). Hence, *1-Precision* and *Recall* are defined as

$$1 - \text{Precision} = 1 - \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{22}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{23}$$

The PR curve is obtained by varying threshold $\mu \in [0, 1]$ and matching exploits the fast library for approximate nearest neighbours [60].

3.3.3 Compactness: This metric relates the descriptive power to the cardinality of a description vector. This is important because the length of the feature vector has a great impact on the memory footprint and computational requirements during the feature matching stage. As previously reported [2], we define *compactness* as

$$\text{compactness} = \frac{\text{Average AUC}}{\text{Descriptor cardinality}} \tag{24}$$

3.4 Implementation

All trials were performed in on an Intel i7 with 16 GB of RAM. Keypoint detectors and descriptors were implemented in C++/PCL. The tuned parameters of each detector (Table 1) and descriptor (Table 2) were used to maximise performance. The untuned parameters were fixed either to those proposed by the original authors or to their PCL implementation [9, 38]. For the tuning process, we used the *Oakland* data set and confirmed that SHOT has a stable description performance regardless of the description radius, whereas TriSI, FPFH and RoPS gain peak performance and then drop [2]. For the scenarios we evaluated, this peak performance was identified at a radius of 20 × Mr, with Mr representing the average Model point cloud resolution. For HoD and HoD-S, optimal performance was achieved at 20 × Tr.

Table 1 Keypoint detectors evaluated

Dimension	Descriptor	Tuned parameters
3D	ISS	—
3D	SP	—
3D	Harris 3D	—
3D	KPQ	—
3D	uniform	grid size of 5× point cloud resolution
2D	GFTT	Min. corner quality 10^{-3}/Gaussian filter size 3 × 3
2D	FAST16-Adaptive scale	Min. corner quality 10^{-3}/Min. intensity contrast 10^{-3}/octaves 4
2D	FAST6-Fixed scale	Min. corner quality 10^{-3}/Min. intensity contrast 10^{-3}
2D	DoG	eight scale levels
2D	FAST-9	intensity threshold 9
2D	Harris 2D	Min. corner quality 10^{-3}/Gaussian filter size 3 × 3
2D	KAZE	six scale levels/six octaves
2D	FH-9	six scale levels/blob threshold 10^{-5}

Table 2 Feature descriptors evaluated

Dimension	Descriptor	Descriptor length	Tuned parameters
3D	SHOT	352	—
3D	HoD	240	—
3D	HoD-S	40	—
3D	FPFH	33	—
3D	RoPS	135	—
3D	TriSI	675	—
2D	FREAK	64	—
2D	SURF	64	—
2D	BRISK	64	—
2D	KAZE	64	—
2D	SIFT	128	eight scale levels

4 Experimental results and discussion

4.1 Evaluation of keypoint detectors

4.1.1 Oakland data set: One important factor affecting the performance of the 2D keypoint detection methods is the quantisation factor q_f used during the 3D to multi-2D remapping process applied to the point cloud. As shown in Fig. 2*a*, the RR increased with q_f for all methods with the exception of DoG, which showed a stable but poor performance. This is because q_f defines the amount of detail preserved on the 2D image projections after the remapping process, with higher q_f values corresponding to a higher resolution. Fig. 2*b* shows the corresponding AR achieved by each method, revealing that RR and AR have a similar relationship with q_f. The low RR performance of DoG reflects the extremely low AR. Due to the log scale of the AR plot, zero AR is omitted and thus AR plots can be interrupted.

The selection of q_f has also a direct impact on the number of detected keypoints, the physical size of the 2D projections and ultimately on the overall processing time required to apply the 2D keypoint detection methods.

Given that we regarded performance and computational efficiency as equally important, we set $q_f = 10$ for the remaining trials. The processing burden for $q_f = 10$ is 57 times lower than $q_f = 100$, but most of the keypoint detection methods still perform well (Fig. 2*a*). Table 3 shows the processing time needed for various q_f values and the process acceleration relative to $q_f = 10$. Fig. 3 shows the processing time needed by each 2D keypoint detector (for $q_f = 10$) and the corresponding time for the 3D keypoint detection methods evaluated herein ($q_f$is not applicable in the 3D methods). Fig. 3 shows that although the computational time of the 2D methods includes the 3D to multi-2D remapping, keypoint detection process for all three planes, and keypoint back-projection to the original 3D domain, the computational burden is much lower than that of almost all 3D descriptors. In terms of processing time Fast9 achieved the lowest, followed by GFTT, Harris and FH. The highest computational burden was associated with KPQ and KPQ-AS.

Subsequent trials on the Oakland data set evaluated the robustness of the 2D and 3D keypoint detection methods challenged by variable resolution, Gaussian noise and SHOT noise. These nuisance factors are added to one of the two scene segments and simultaneously the same segment was also randomly rotated up to 180° in pitch, roll and yaw and translated up to 10 m in the *X*, *Y* and *Z* directions, creating a highly complex and challenging scenario that exceeds the typical difficulty of current computer vision scenarios. This complexity was introduced to investigate the limits of the keypoint detection methods and the single and cross-dimensional 2D/3D keypoint detection and description methods described in Section 4.2.

In the nuisance-free setting, the 2D keypoint detection methods achieved an average RR of 38% compared to 22% for the 3D methods, indicating that the 2D methods are more robust to resolution variation (Fig. 4). This is mainly due to the coordinate remapping process in (17), which transforms the floating-point

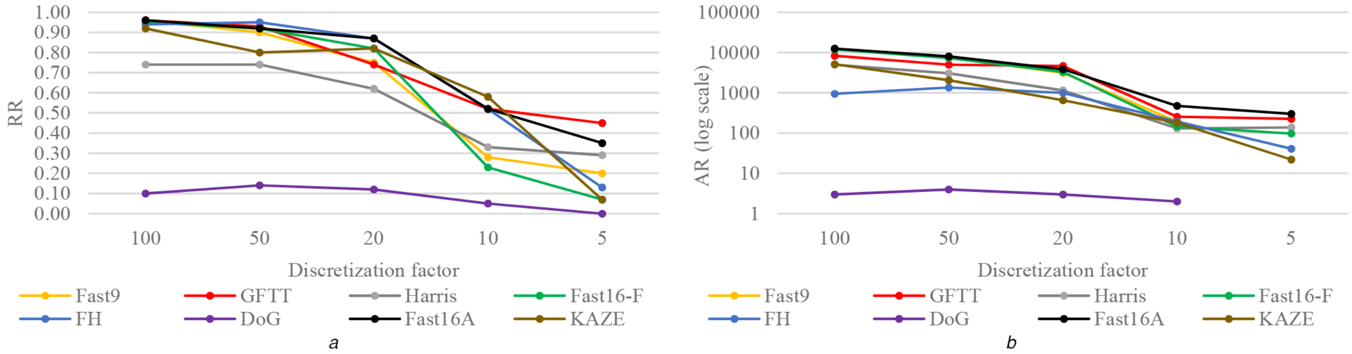


Fig. 2 *Impact of discretisation factor on 2D keypoint detection on the Oakland data set*
(a) RR, ***(b)*** AR

Table 3 Overall processing time for various q_f values

q_f	100	50	20	10	5
avg. time, s	19.12	6.14	1.26	0.33	0.31
gain factor	57	18	4	1	1

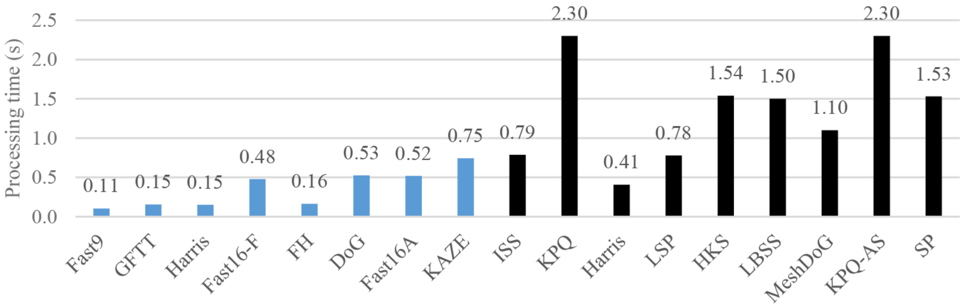


Fig. 3 *Processing time of 2D (blue) and 3D (black) keypoint detectors (q_f = 10 for the 2D methods)*

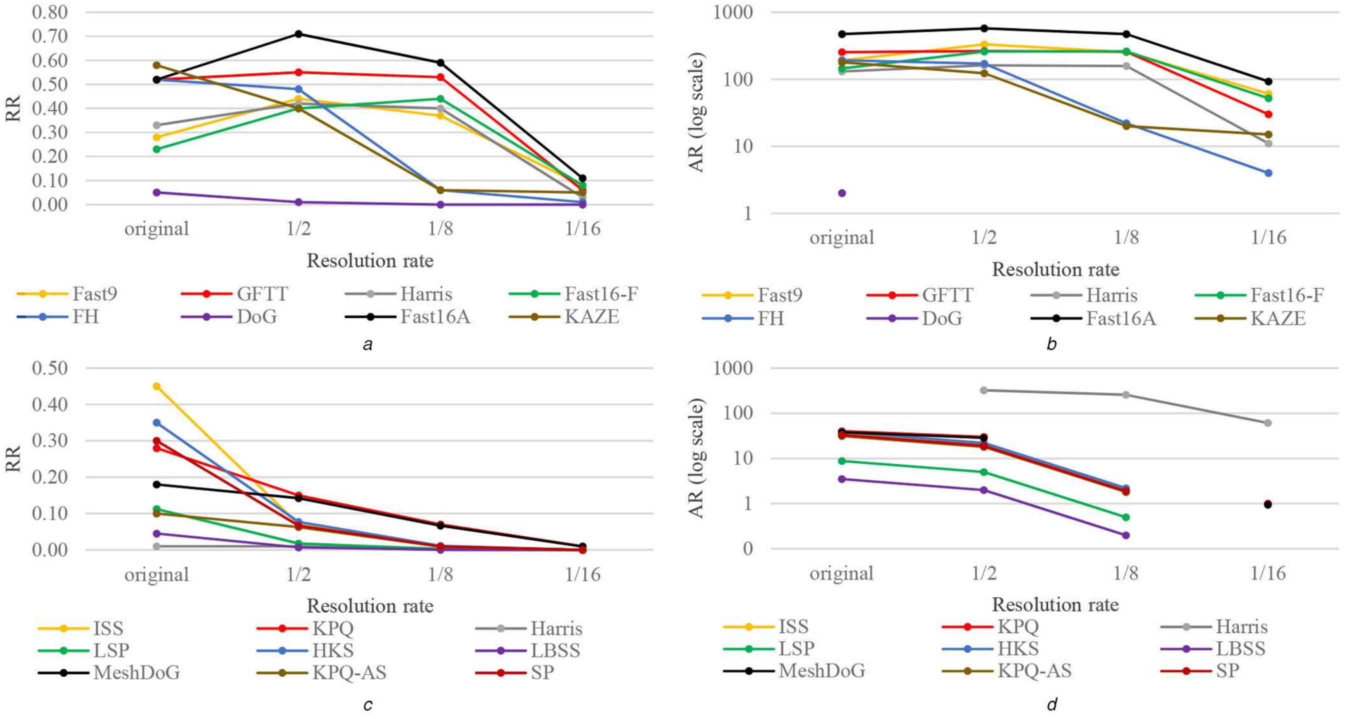


Fig. 4 *Evaluating detector performance on the Oakland data set under various resolution levels*
(a) 2D techniques RR, ***(b)*** 2D techniques AR, ***(c)*** 3D techniques RR, ***(d)*** 3D techniques AR (original resolution refers to resolution during data set acquisition)

vertex coordinates into pixel coordinates. Even when the resolution was reduced to one eighth of the original value, most of the 2D methods, namely Fast9, GFTT, Harris, Fast16-F (fixed scale) and Fast16-A (adaptive scale), were still able to achieve appealing RR and AR scores. Interestingly, DoG performed less well than anticipated, but this was due to the extremely small number of keypoints it provided. In contrast, the 3D methods were much more vulnerable to resolution variation even when the resolution was reduced to only half its original value.

Next we investigated the robustness of each method to various Gaussian noise levels with zero mean and standard deviation $\sigma = \{0.1\text{Mr}, 0.3\text{Mr}, 0.5\text{Mr}\}$ [9, 38]. Figs. 5*a* and *b* clearly shows that the 2D keypoint detectors were only marginally affected regardless of the noise level, with KAZE, GFTT, Fast16-A and FH demonstrating a highly appealing and stable performance. This is because the low quantisation value $q_f = 10$ during the coordinate remapping process of (17) quantises the noisy vertex coordinates in the same pixel coordinates as seen in the noise-free case. Unlike the 2D methods, the 3D methods were strongly affected even by low Gaussian noise levels (Figs. 5*c* and *d*).

Finally, we evaluated the robustness of each method to various SHOT noise levels modelled with a Poisson process where $\lambda = \{0.1\text{Mr}, 0.3\text{Mr}, 0.5\text{Mr}\}$. Fig. 6 shows that the 2D descriptors were only marginally affected, retaining their high RR and AR values. Their appealing performance is yet again due to the quantisation process of (17) and the small q_f value. In contrast, the 3D descriptors were strongly influenced by even low levels of SHOT noise.

Regarding the overall performance of the 2D and 3D keypoint detectors on the Oakland data set, it is evident that the majority of the 2D techniques outperform the 3D ones both in terms of processing efficiency and robustness to resolution and nose variation.

For the sake of a reasonable paper length, in the remaining challenges using alternative data sets, the 2D and 3D keypoint detection methods were tested against the standard data set alone, without resolution or noise variation.

4.1.2 Laser scanner data set:

When tested against the Laser Scanner data set, the best RR performance was achieved by the 3D keypoint detector ISS with the 2D detector GFTT following closely behind (Fig. 7*a*). The 2D and 3D methods achieved average RR values of 21 and 20%, respectively, and in both cases the AR

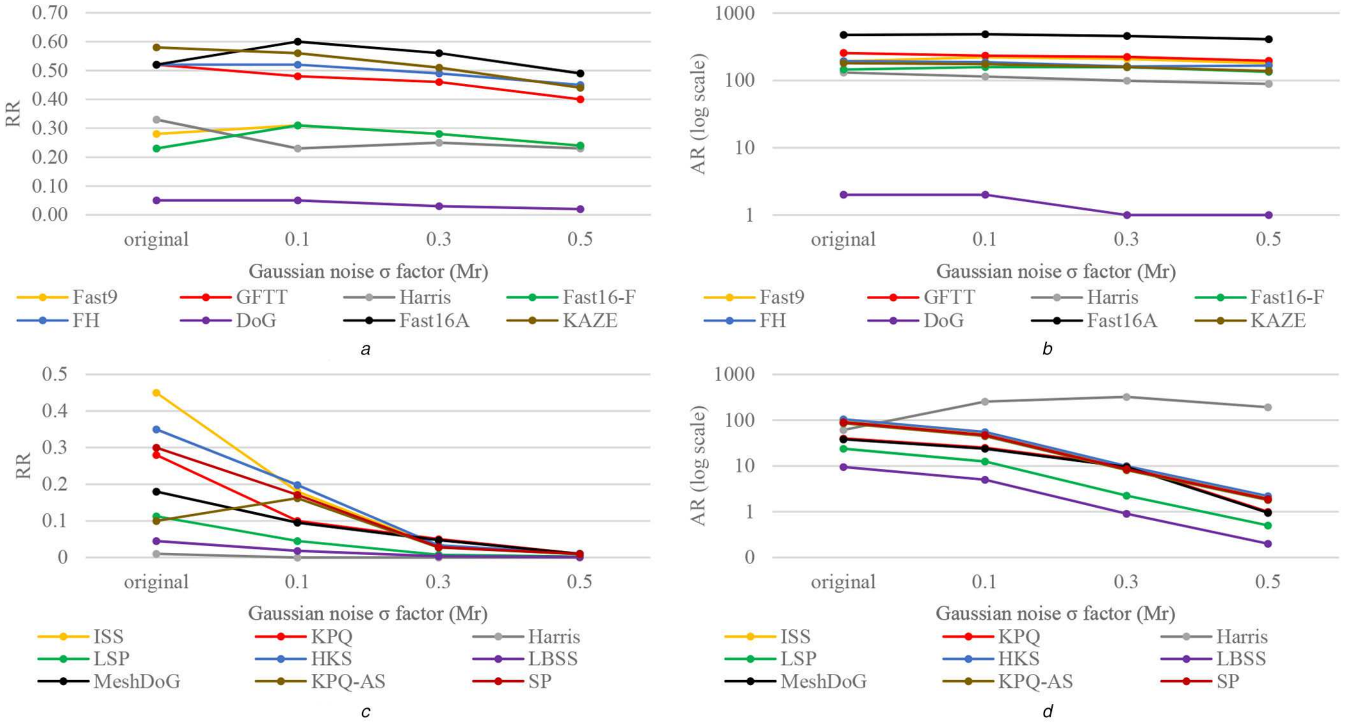


Fig. 5 *Evaluating detector performance on the Oakland data set under various Gaussian noise levels*
(a) 2D techniques RR, ***(b)*** 2D techniques AR, ***(c)*** 3D techniques RR, ***(d)*** 3D techniques AR (original noise level refers to noise during data set acquisition)

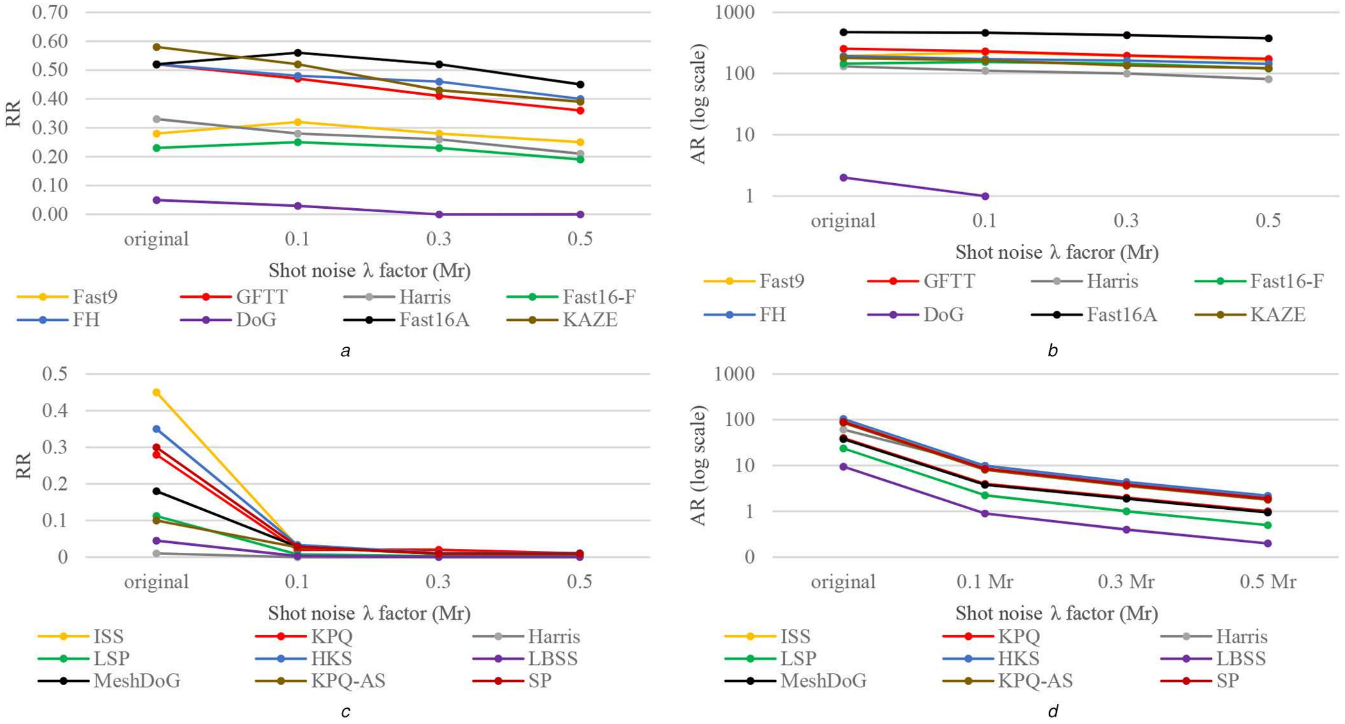


Fig. 6 *Evaluating detector performance on the Oakland data set under various SHOT noise levels*
(a) 2D techniques RR, ***(b)*** 2D techniques AR, ***(c)*** 3D techniques RR, ***(d)*** 3D techniques AR (original noise level refers to noise during data set acquisition)

values provided on average a similar number of keypoints (Fig. 7*b*).

4.1.3 Kinect data set: When tested against the Kinect data set, most of the 2D keypoint detectors (GFTT, FH, KAZE, Harris and Fast9) achieved a better RR performance than the corresponding 3D methods, with GFTT and FH exceeding 75% RR (Fig. 8). The average RR of the 2D methods (45%) was far superior to the average RR of the 3D methods (22%).

4.1.4 SpaceTime data set: The 2D methods achieved higher RR values than the 3D methods when tested against the SpaceTime data set, with Fast16-A performing best (Fig. 9). The average RR of the 2D methods was 47%, compared to 28% for the 3D methods.

4.1.5 Discussion: From the keypoint detection trials it is evident that the 2D methods are overall more appealing than the 3D methods for several reasons:

(i) In the nuisance-free *Oakland* and *Kinect* data sets, the RR values of the 2D methods were double those of the 3D methods. For the *Laser Scanner* data set, both the 2D and 3D methods achieved a similar performance.

(ii) The 2D methods were superior in terms of robustness to nuisances (resolution variation, Gaussian and SHOT noise). This was mainly due to the low quantisation value $q_f = 10$ during the coordinate remapping process of (17). In contrast, due to the challenging complexity of the *Oakland* scenario, the 3D keypoint detection techniques achieved very low RR values even at the lowest nuisance levels.
(iii) The 2D methods were four times faster to execute than the 3D methods, despite the former requiring a multi-staged process that includes 3D to multi-2D remapping, keypoint detection on all three planes and keypoint back-projection to the original 3D domain.

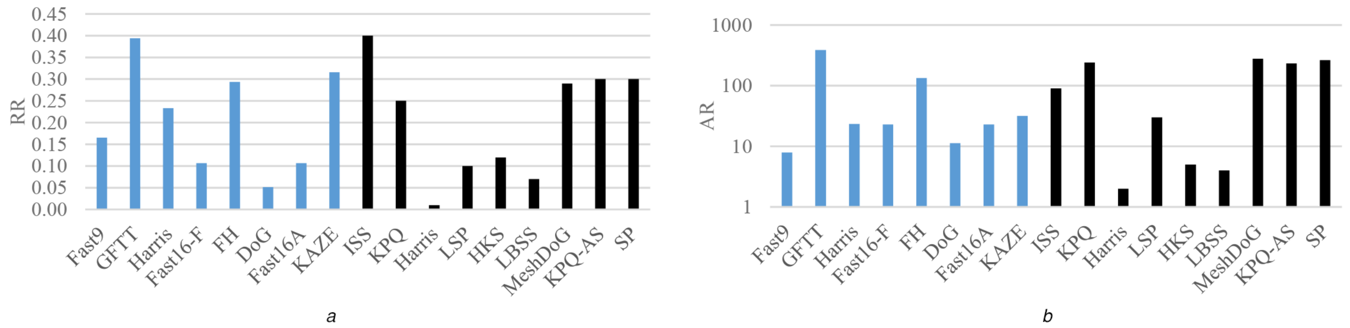
4.2 Evaluation of feature descriptors

Next, we conducted single and cross-dimensional evaluations of the 2D and 3D keypoint detection and description methods on the data sets described in Section 4.1. The trials comprised 2D–2D, 2D–3D, 3D–2D and 3D–3D schemes, where the first and second numbers indicate the dimensionality of the keypoint detector and feature descriptor, respectively. To improve clarity, only the GFTT, Fast16-A and FH keypoint detection methods were used for the 2D scenarios, and only ISS for the 3D scenario. The selection was based on both the RR metric and the computational efficiency demonstrated in Section 4.1. For the 3D keypoint detection methods, we also investigated the performance by applying a uniform subsampling scheme and scoring based on the AUC metric.

4.2.1 Oakland data set: In the first trial, we evaluated the 2D–2D scheme and tested robustness to resolution variation, Gaussian noise and SHOT noise using the same parameters described for the evaluation of keypoint detection (Section 4.1.1).

SURF and KAZE were the most robust feature descriptors in response to resolution variation regardless of the keypoint detection method (Fig. 10*a*). Among the three 2D keypoint detectors we challenged, Fast16-A achieved the highest AUC value and the most robust combination was Fast16-A with the SURF descriptor. In contrast, SIFT, FREAK and BRISK achieved low AUC values at all resolutions regardless of the associated 2D keypoint detector. Interestingly, SIFT, FREAK and BRISK achieved low AUC values even when applied to the original nuisance-free scene.

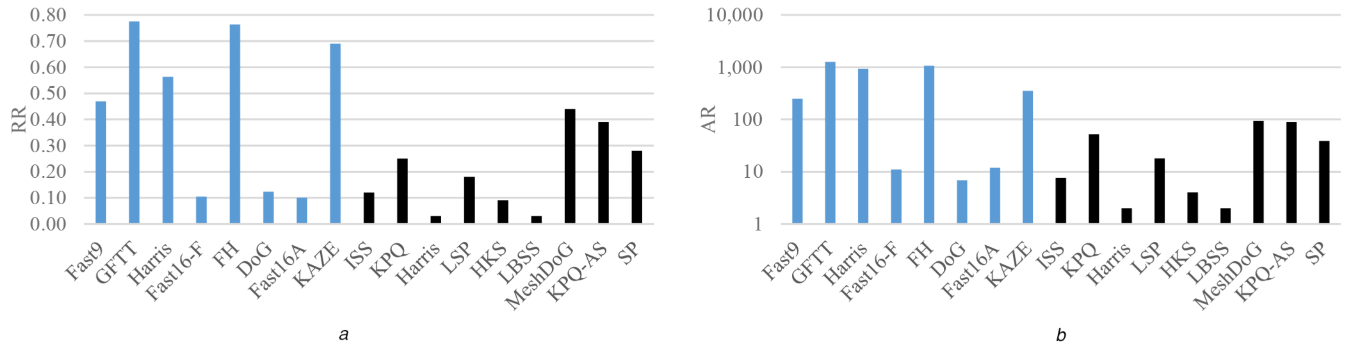
We also evaluated the robustness of the 2D–2D scheme with various levels of Gaussian noise. SURF and KAZE were again the most robust (Fig. 10*b*). Similarly to the initial trial, FREAK, BRISK and SIFT achieved low AUC scores regardless of the Gaussian noise level. The same trend was observed in the SHOT noise trial (Fig. 10*c*). In both noise trials, the AUC achieved by each method was quite stable regardless of the noise level,



a

b

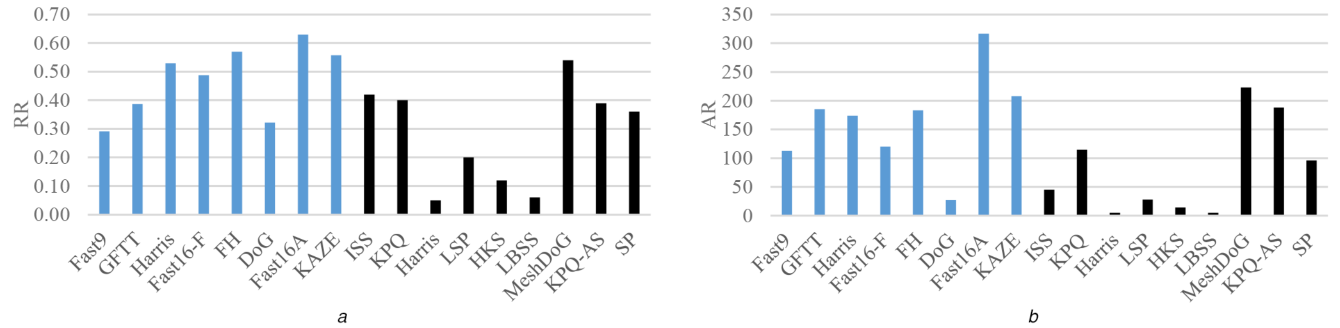
Fig. 7 *Evaluating 2D (blue) and 3D (black) keypoint detectors on the Laser Scanner data set*
(a) RR, ***(b)*** AR



a

b

Fig. 8 *Evaluating 2D (blue) and 3D (black) keypoint detectors on the Kinect data set*
(a) RR, ***(b)*** AR



a

b

Fig. 9 *Evaluating 2D (blue) and 3D (black) keypoint detectors on the SpaceTime data set*
(a) RR, ***(b)*** AR

highlighting the robustness of the 2D methods to noise and also the important contribution of the quantisation process of (17).

In the second trial, we considered the 2D–3D scheme. All of the 3D descriptors we tested were sensitive to resolution variation (Fig. 11*a*), and given the robustness already shown for the 2D keypoint detection methods (Fig. 4*a*), the low AUC values in the 2D-3D trial were attributed to the 3D descriptors. In contrast, the 2D–3D scheme was more robust against noise nuisances but still inferior to the 2D–2D scheme. For Gaussian and SHOT noise (Figs. 11*b* and *c*), the performance of each 3D descriptor depended strongly on the 2D keypoint detector. Hence, for the GFFT keypoint detector, the best performance was achieved by RoPS, closely followed by HoD-S, FPFH and HoD. A similar trend was apparent for the Fast16-A keypoint detector. However, the FH keypoint detector resulted in higher AUC values for most of the 3D descriptors, with HoD-S and HoD again achieving highest AUC values. Interestingly, TriSI and SHOT achieved a low AUC value regardless of the nuisance applied. Overall, the performance of the 2D–3D scheme was inferior to that of the 2D–2D scheme.

The third trial was the cross-dimensional 3D–2D scheme. ISS was more robust to resolution variation compared to uniform subsampling (Fig. 12*a*). Interestingly, the hierarchy between ISS and the uniform subsampling strategy was the same for the 2D–2D and 3D–2D schemes, suggesting that the AUC is mostly affected by the 2D feature descriptors rather than the dimensionality of the keypoint detection method. Even so, the cross-dimensional 3D–2D scheme based on ISS and SURF was more robust to resolution variation, achieving a relatively stable AUC at all resolution levels. The robustness of the 3D-2D scheme to Gaussian noise variation is shown in Fig. 12*b*. The performance of both 3D keypoint detection methods was similar, with ISS gaining a slight advantage. Again, the hierarchy of the 2D–2D and 3D–2D schemes was the same. Finally, we challenged the 3D–2D scheme with various levels of SHOT noise (Fig. 12*c*). SURF achieved the highest performance, although both SURF and KAZE generated appealing AUC scores.

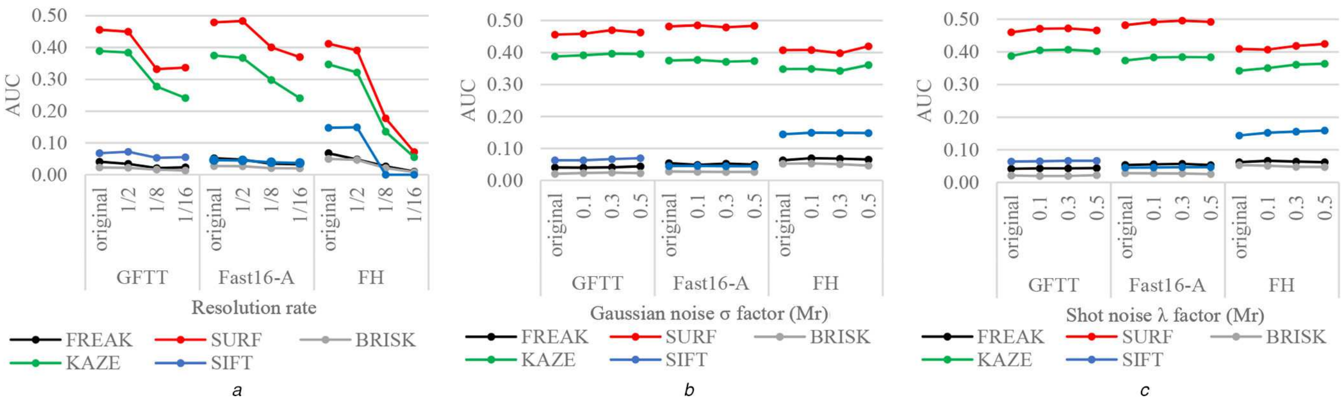


Fig. 10 *Evaluating 2D keypoint detectors and 2D feature descriptors on the Oakland data set*
(a) Resolution variation, ***(b)*** Gaussian noise, ***(c)*** SHOT noise (original noise and resolution refer to noise and resolution, respectively, during data set acquisition)

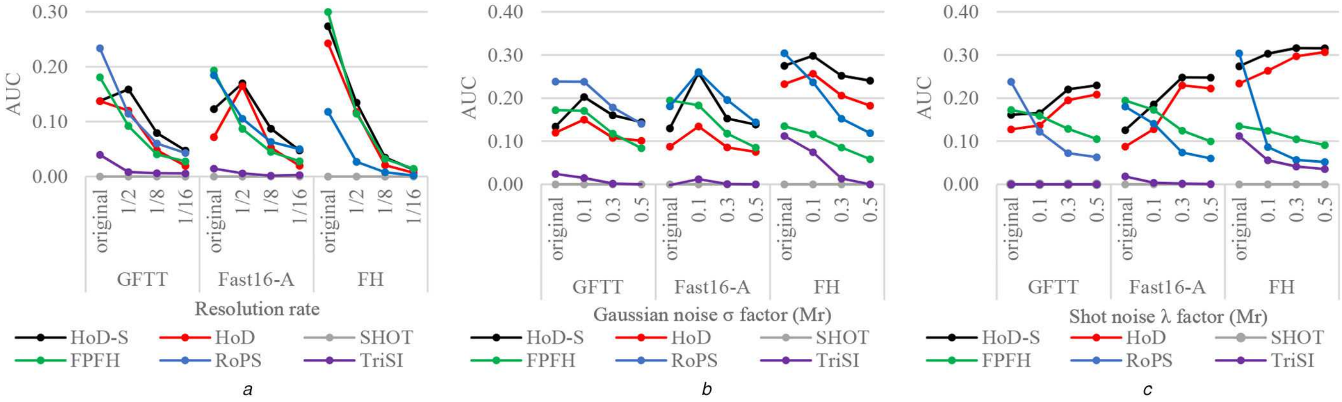


Fig. 11 *Evaluating 2D keypoint detectors and 3D feature descriptors on the Oakland data set*
(a) Resolution variation, ***(b)*** Gaussian noise, ***(c)*** SHOT noise (original noise and resolution refer to noise and resolution during data set acquisition)

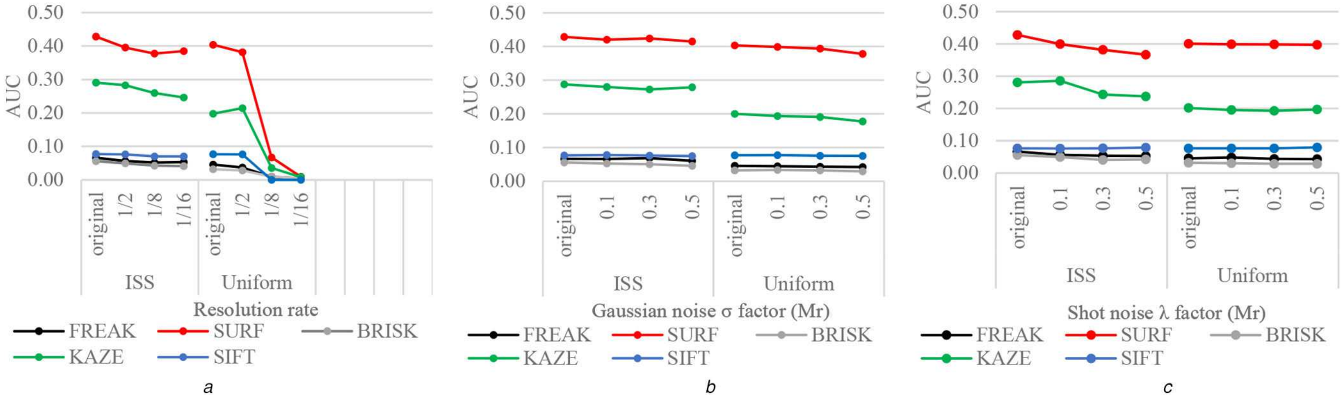


Fig. 12 *Evaluating 3D keypoint detectors and 2D feature descriptors on the Oakland data set*
(a) Resolution variation, ***(b)*** Gaussian noise, ***(c)*** SHOT noise (original noise and resolution refer to noise and resolution during data set acquisition)

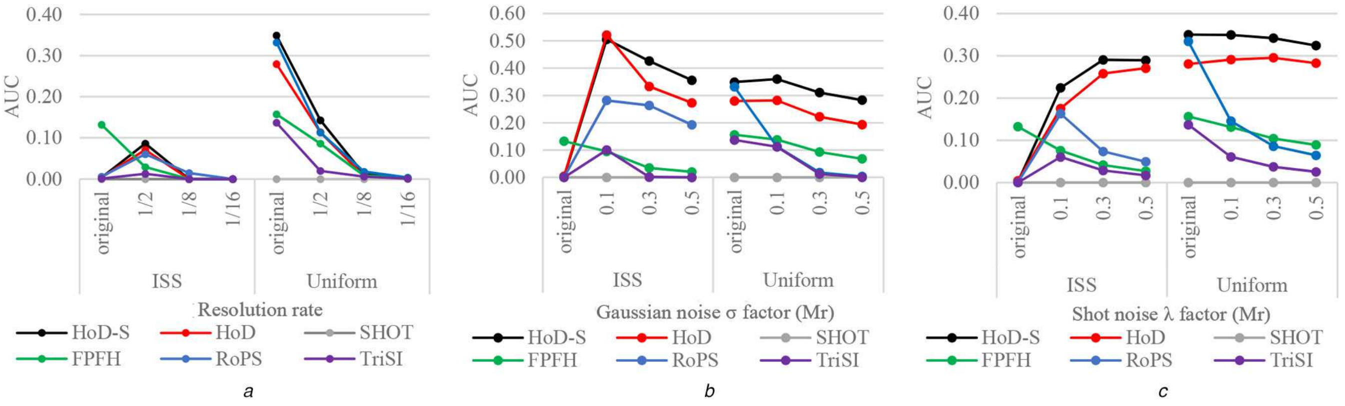


Fig. 13 *Evaluating 3D keypoint detectors and 3D feature descriptors on the Oakland data set*
(a) Resolution variation, ***(b)*** Gaussian noise, ***(c)*** SHOT noise (original noise and resolution refer to noise and resolution during data set acquisition)

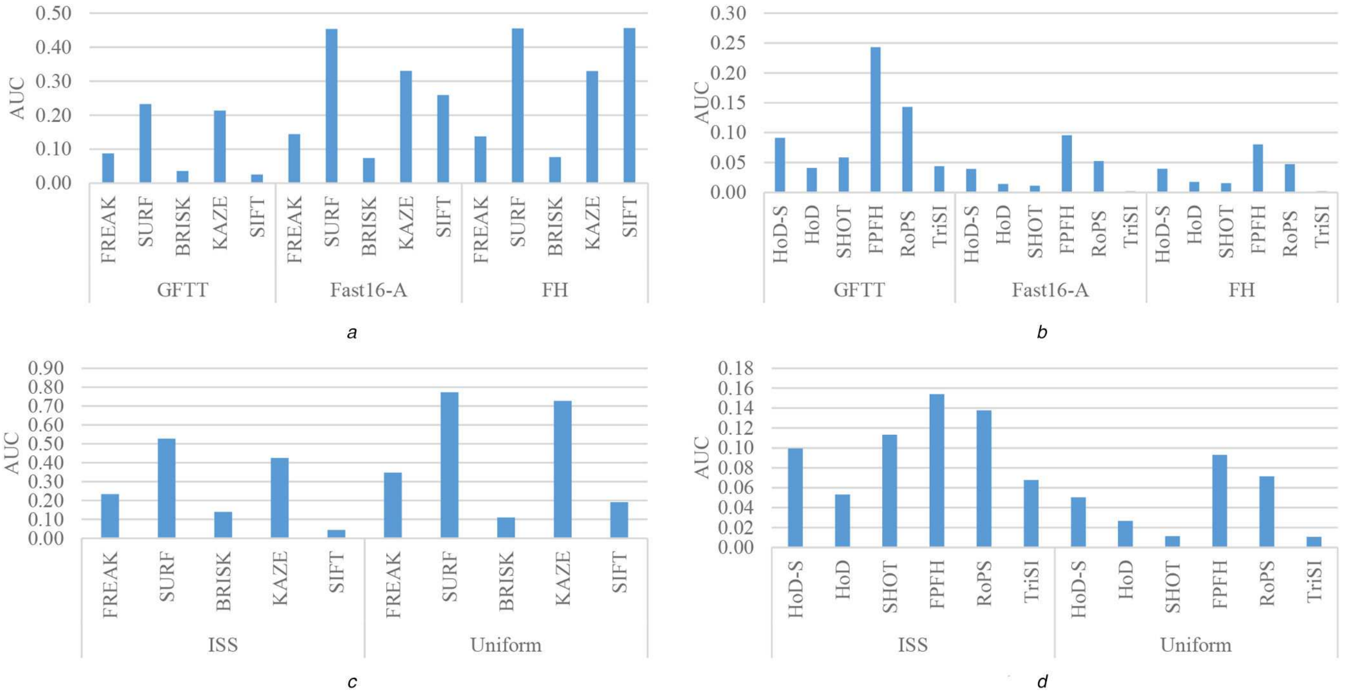


Fig. 14 *Evaluating keypoint detector and feature descriptor combinations on the Laser Scanner data set*
(a) 2D–2D, ***(b)*** 2D–3D, ***(c)*** 3D–2D, ***(d)*** 3D–3D

Again, the hierarchy of the 2D–2D and 3D–2D schemes was preserved.

The final trial considered the 3D–3D scheme. The performance of this scheme in all three nuisance trials was similar to the 2D–3D scheme, with the 3D–3D scheme showing marginally better AUC values.

Given that these combinations involved 2D methods, the processing time not only includes the keypoint detection and feature description methods but also the 3D to multi-2D projection and 2D to 3D back-projection. For the overall performance of the keypoint detection methods considering all description methods, the fastest 2D technique was Fast16-A, and of the 3D methods, uniform subsampling was faster than ISS. For the feature description methods and their overall performance considering all keypoint detection methods, we conclude that most efficient 2D descriptor is SURF, and the most efficient 3D descriptor is HoD-S.

The evaluations on the Oakland data set lead to the following conclusions:

(i) The 2D–2D combination achieves the highest overall performance in terms of AUC and processing efficiency.
(ii) The 2D feature descriptors preserve their hierarchy and their performance regardless of the keypoint detection dimensionality and method.

The 2D feature descriptors are more robust to nuisances than their 3D counterparts. The performance degradation of the 3D descriptors in response to increasing nuisance levels is also described elsewhere, although in the context of different data sets [2]. Therefore, 3D descriptors appear to generally suffer from low robustness to resolution variation, Gaussian noise and SHOT noise.

4.2.2 Laser scanner data set: As stated above, we only considered the nuisance-free versions of the Laser Scanner, Kinect and SpaceTime data sets. The performance of the various keypoint detection and feature description methods against the Laser Scanner data set is shown in Fig. 13. For the 2D–2D scheme (Fig. 14*a*), the highest AUC was achieved by combining SURF with FH, or Fast16-A and SIFT with FH. For the 2D–3D scheme (Fig. 14*b*), GFTT combined with FPFH performed best, whereas Fast16-A or FH combined with any 3D feature descriptor resulted in the poorest performance. The 3D–2D scheme was the best of all four of the dimensional combinations (Fig. 14*c*). Specifically, 3D uniform subsampling combined with SURF and KAZE achieved AUC values of 0.77 and 0.73, respectively. This is almost twice the highest value generated by the 2D–2D scheme, three times that of the 2D–3D scheme and five times that of the 3D–3D scheme. Interestingly, the 3D–3D combination, which is the standard approach for 3D data in the form of point clouds, achieved the lowest AUC scores (Fig. 14*d*) in agreement with earlier studies [2].

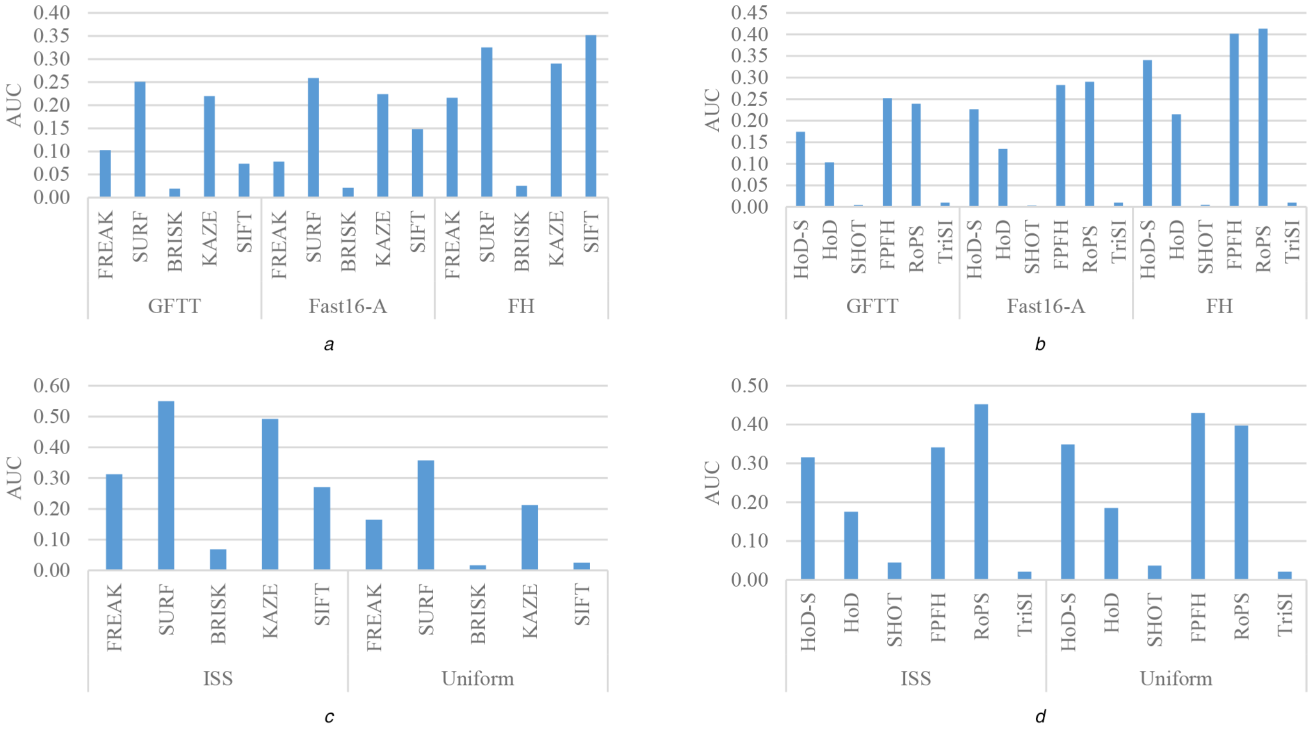


Fig. 15 *Evaluating keypoint detector and feature descriptor combinations on the Kinect data set*
(a) 2D–2D, ***(b)*** 2D–3D, ***(c)*** 3D–2D, ***(d)*** 3D–3D

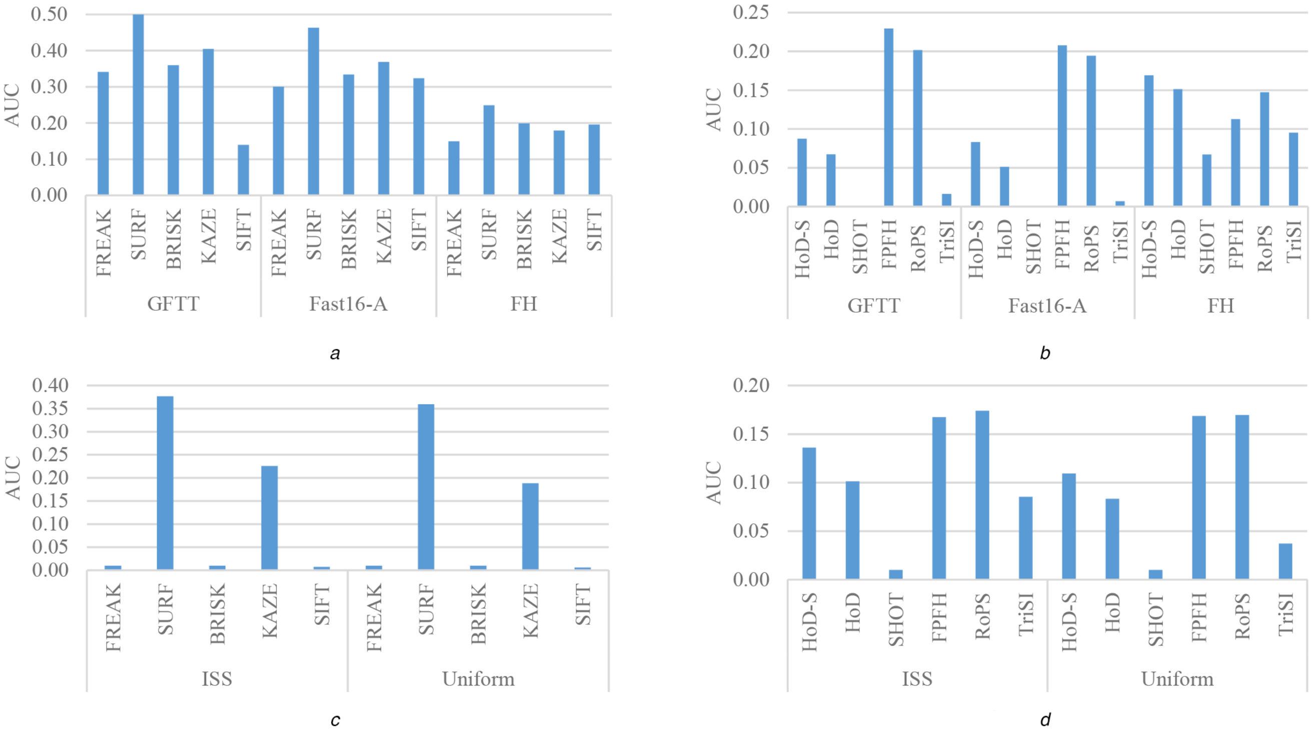


Fig. 16 *Evaluating keypoint detector and feature descriptor combinations on the SpaceTime data set*
(a) 2D–2D, ***(b)*** 2D–3D, ***(c)*** 3D–2D, ***(d)*** 3D–3D

4.2.3 Kinect data set: The AUC values representing each combination of methods tested against the standard Kinect data set are summarised in Fig. 15. The 3D–2D scheme achieved the highest AUC values, specifically ISS combined with SURF.

4.2.4 SpaceTime data set: The AUC values representing each combination of methods tested against the standard SpaceTime data set are summarised in Fig. 16. Here, the 2D–2D scheme achieved the highest AUC scores, followed by the 3D–2D scheme. Interestingly, however, only SURF and KAZE provided meaningful AUC values.

4.2.5 Robustness overall performance: To enhance our comparison of the single and cross-dimensional keypoint detection and feature description combinations, the AUC scores achieved by each method averaged over all data sets are presented in Table 4, along with the average performance per keypoint detection and feature description technique on an independent basis.

This analysis shows that the 3D keypoint detectors contribute to higher AUC values, with the 2D FH method following closely behind. Regarding the feature descriptors, SURF clearly achieves the highest AUC values regardless of the keypoint detector, followed by KAZE. RoPS achieves the highest performance among the 3D techniques, but still lower than SURF and KAZE. Overall,

Table 4 Average AUC performance on all data sets

			2D descriptors					3D descriptors						
			FREAK	SURF	BRISK	KAZE	SIFT	HoD-S	HoD	SHOT	FPFH	RoPS	TriSI	Average
keypoint detectors	2D	GFTT	0.14	0.36	0.11	0.31	0.08	0.12	0.09	0.02	0.23	0.20	0.03	0.15
		Fast16-A	0.14	0.41	0.11	0.32	0.20	0.12	0.09	0.01	0.20	0.18	0.01	0.16
		FH	0.15	0.36	0.09	0.29	0.29	0.21	0.16	0.03	0.18	0.23	0.06	0.18
	3D	ISS	0.16	0.47	0.07	0.36	0.10	0.14	0.08	0.04	0.20	0.26	0.07	0.18
		Uniform	0.14	0.47	0.04	0.33	0.08	0.22	0.14	0.02	0.21	0.24	0.05	0.18
		average	0.15	0.42	0.09	0.32	0.15	0.16	0.11	0.02	0.20	0.22	0.04	

Table 5 Total processing time

			2D descriptors					3D descriptors						
			FREAK	SURF	BRISK	KAZE	SIFT	HoD-S	HoD	SHOT	FPFH	RoPS	TriSI	Average
keypoint detectors	2D	GFTT (F)	0.23	0.11	0.92	0.14	0.26	2.64	4.05	8.20	54.78	26.30	44.92	12.96
		Fast16 (A)	0.16	0.05	0.80	0.07	0.26	0.83	1.17	5.19	26.51	7.69	29.61	6.58
		FH (A)	0.61	0.05	0.80	0.07	0.05	0.65	0.93	4.82	34.34	5.95	29.61	7.08
	3D	ISS	0.29	0.20	0.94	0.29	0.26	1.84	2.74	6.54	59.46	21.39	46.35	12.75
		uniform	0.20	0.11	0.84	0.15	0.27	0.98	1.40	5.19	53.23	9.43	35.15	9.72
		average	0.30	0.10	0.86	0.14	0.22	1.39	2.06	5.99	45.66	14.15	37.13	

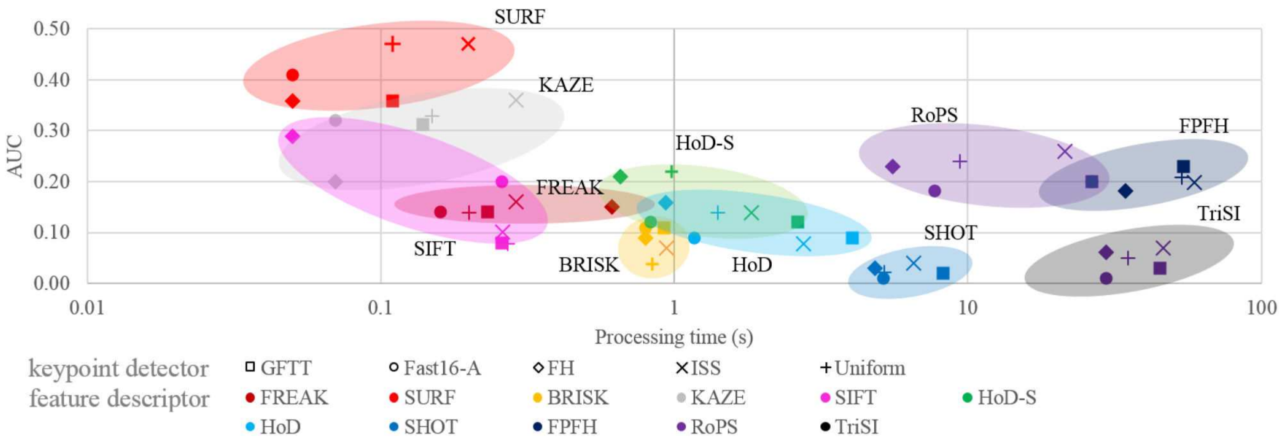


Fig. 17 *Overall performance visualising AUC versus processing time (best seen in colour)*

the 3D–2D scheme is the most appealing combination, achieving the highest AUC values while imposing among the lowest computational requirements. A detailed analysis of the computational requirements is presented in Table 5 summarising the computational time required by each 2D/3D keypoint detection and feature description combination, and the average time required by each method regardless of the combination. The most efficient methods were Fast16-A combined with SURF and FH combined with SURF, each requiring only 0.05 s per point cloud.

For completeness, in Fig. 17 we visualise the AUC versus processing time relationship between every single and cross-modal combination. From Fig. 17 it is obvious that the feature descriptor is the main computational contributor greatly defining the overall performance and the processing time of each combination. As expected, the 2D descriptors are faster to compute with HoD-S and HoD being at the computational margin of the 2D and the 3D techniques. Interestingly, despite 3D descriptors being designed to manipulate 3D data, the 2D techniques afford a higher AUC. This is because during the 3D to multi-2D data remapping process and vice versa, quantising the vertex coordinates to pixel coordinates contributes to reducing the minor model-scene nuisances and thus enforcing the descriptor's robustness. Additionally, it is worth noting that Fig. 17 reveals that the keypoint detection hierarchy within each coloured feature descriptor-based cluster places the 3D keypoint detection methods as the top performing ones. Exceptions are SIFT, BRISK, HoD and FPFH.

Additionally, in Figs. 18 and 19 we present examples of single and cross dimensional keypoint detection/feature description correspondences utilising scenes from the Oakland data set under 1/8 subsampling rate difference. However, to maintain a reasonable paper length we only present the top AUC keypoint detection / feature description combinations as of Table 4. Figs. 18 and 19 highlight that overall the 2D descriptors are superior compared to the 3D ones. This is due to the 2D to 3D remapping and vice versa that smoothes the different point cloud densities.

We also evaluated the performance of each method based on the compactness metric. This reveals the description capability of each feature description technique, but also uses the AUC value so assesses the joint performance of the keypoint detector and feature descriptor. Table 6 presents the average compactness values for all data sets, revealing that ISS and uniform subsampling combined with SURF are the most descriptive combinations.

Overall, the contributions of our work can be summarised to

(i) In contrast to current literature, we evaluated current keypoint detection and feature description methods on a broader basis by adopting a novel cross-dimensional (mixed 2D and 3D) scheme. It is worth noting that such a broad multi-dimensional evaluation has not yet been reported in the literature.

(ii) Our trials demonstrated that 2D keypoint detectors attain higher repeatability rates, are less prone to nuisances such as resolution variation and noise, and are faster to compute compared to their 3D counterparts. The reason for their advantages was the 2D–3D remapping, which flattened minor nuisances. However, this smoothing process negatively affected the 2D feature description process reducing their descriptiveness. In terms of processing efficiency, manipulating 2D data with a low quantisation factor for the 3D–2D remapping process was computationally more efficient than directly exploiting 3D data despite the remapping process. In fact, the average processing time per modality combination

(Table 5) was 0.31, 0.36, 16.01 and 20.31 s for the 2D–2D, 3D–2D, 2D–3D and 3D–3D for the keypoint detection – feature description combinations, respectively.

(iii) The proposed cross-dimensional evaluation scheme revealed that a cross-dimensional solution combining the 3D keypoint detector ISS or uniform subsampling with the 2D feature descriptor SURF are the most appealing combinations. This is because it incorporates the advantages of each individual data modality and technique, affording high AUC scores with only a minor computational burden compared to the faster 2D single-dimensional keypoint detection and feature description techniques. Despite ISS/SURF and uniform/SURF attain only 5% higher AUC compared to the 2D keypoint detection methods combined with SURF, this performance gain is still important for registration and low-drift LIDAR-based odometry applications. Examples of the feature correspondences between the model and the scene point clouds are presented in Figs. 18 and 19.

(iv) Overall, from Table 4 the average AUC per modality combination was 0.22, 0.22, 0.12 and 0.14 for the 2D–2D, 3D–2D, 2D–3D and 3D–3D modality combinations highlighting the contribution of the 2D feature descriptors. In terms of compactness, based on Table 6, the average values per modality combination are 3.20, 3.34, 1.98, 2.19, for the corresponding 2D–2D, 3D–2D, 2D–3D and 3D–3D combinations, revealing that the small feature length of the 2D descriptors affords a higher description capability per feature element. The latter is important as while the feature descriptiveness is high, the memory storage requirements are maintained low.

5 Conclusions

Local feature detection and description techniques are commonly used for 3D object registration and recognition applications. Therefore, current literature offers several evaluations of 3D local feature detection and description methods. However, literature is

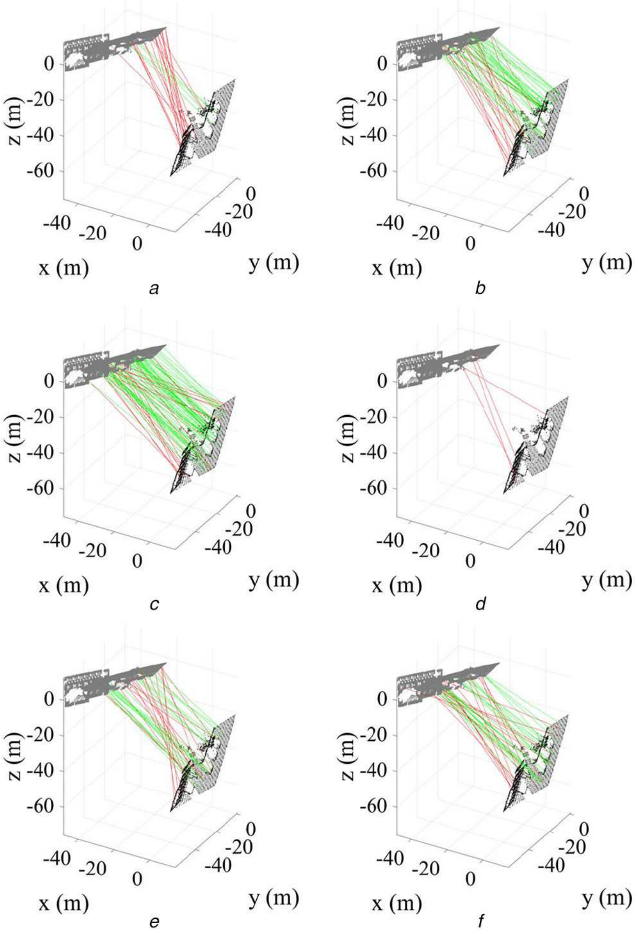


Fig. 18 *Example presenting the overall top AUC performing cross/single dimensional feature correspondences on two scenes of the Oakland data set under 1/8 subsampling rate difference*

(a) ISS/FREAK, ***(b)*** ISS/SURF, ***(c)*** Uniform/SURF, ***(d)*** GFTT/BRISK, ***(e)*** ISS/KAZE, ***(f)*** FH/SIFT (green and red lines present the TP and FP correspondences, figure is best seen in colour)

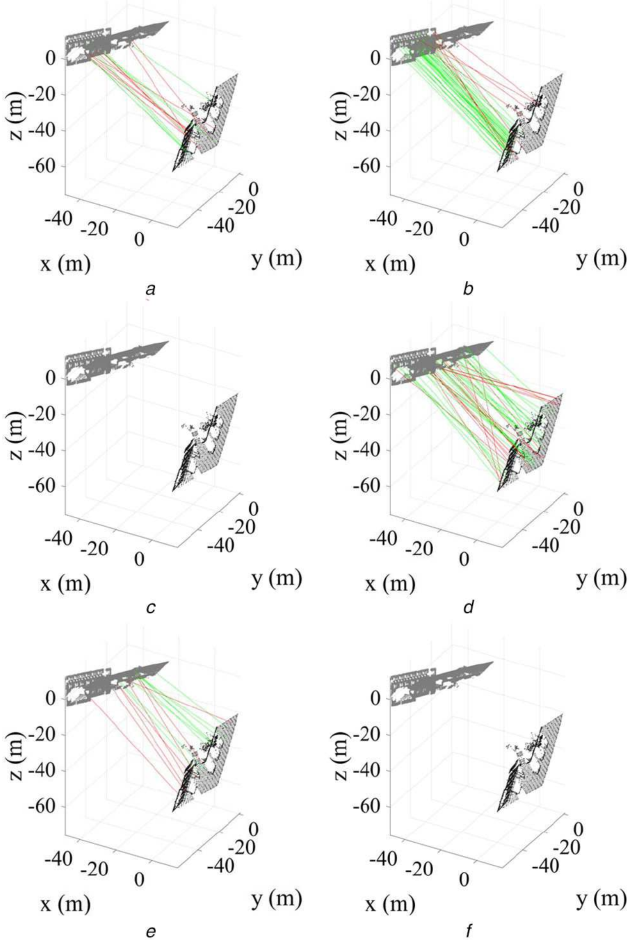


Fig. 19 *Example presenting the overall top AUC performing cross/single dimensional feature correspondences on two scenes of the Oakland data set under 1/8 subsampling rate difference*

(a) Uniform /HoD-S, ***(b)*** FH/ HoD, ***(c)*** ISS/SHOT, ***(d)*** GFTT/FPFH, ***(e)*** ISS/ RoPS, ***(f)*** ISS/TriSI (green and red lines present the TP and FP correspondences, figure is best seen in colour)

Table 6 Average compactness on all data sets

			2D descriptors						3D descriptors				
			FREAK	SURF	BRISK	KAZE	SIFT	HoD-S	HoD	SHOT	FPFH	RoPS	TriSI	Average
keypoint detectors	2D	GFTT	2.23	5.63	1.72	4.77	0.61	3.06	0.36	0.05	6.82	1.50	0.04	2.43
		Fast16-A	2.23	6.45	1.76	5.04	1.52	2.94	0.38	0.01	5.91	1.31	0.01	2.51
		FH	2.27	5.66	1.41	4.49	2.27	5.13	0.65	0.07	5.53	1.69	0.09	2.66
	3D	ISS	2.42	7.38	1.09	5.63	0.78	3.50	0.34	0.11	5.98	1.94	0.10	2.66
		uniform	2.23	7.38	0.66	5.20	0.61	5.38	0.59	0.04	6.44	1.80	0.08	2.76
		average	2.27	6.50	1.33	5.02	1.16	4.00	0.46	0.06	6.14	1.65	0.07	

constrained to evaluating methods of a single data dimension, i.e. either 3D or 2D methods that are applied onto multiple projections of the 3D data, while cross-dimensional (mixed 2D and 3D) feature detection and description has not been investigated yet. Spurred by that evaluation gap and aiming at exploiting the advantages of both the 2D and the 3D methods, we evaluated all possible multi-dimensional combinations of keypoint detection and feature description methods and compared their performance against the single-dimensional methods. Our evaluation included four data sets differing in quality and complexity and under various levels of three nuisance factors (resolution variation, Gaussian and SHOT noise).

Our trials indicated that the optimum combination is multi-dimensional, contrasting with the typical approach for keypoint detection and feature description of 3D data based on methods explicitly designed for the 3D data domain. Specifically, we found that the most appealing keypoint detection/feature description combination is a cross-dimensional scheme blending the 3D ISS/uniform subsampling with the 2D FH. Our findings demonstrated that a pure 3D scheme poses an inferior solution due to the mediocre AUC performance, the poorer performance under resolution and noise nuisances, and the higher computational burden. In fact, we demonstrated that a cross-dimensional 3D keypoint detection and 2D feature description combination is more appealing than a typical single dimensional 3D solution affording twice the performance and a 54× computational speedup. These findings are especially important for time critical applications that involve accurate correspondence estimation of 3D point cloud data.

Additionally, our evaluation revealed that the major contributor to the processing burden of both the single and the multi-dimensional keypoint detection and feature description schemes is the description part. Hence, all schemes involving a 3D feature descriptor are on average one order of magnitude slower. Considering the pure 2D solutions, these are appealing but attain lower AUC values compared to the 3D–2D multi-dimensional combination.

Future work shall focus on implementing our findings on time-critical applications such as LIDAR-based odometry for vehicle, unmanned air vehicles, sea and space odometry applications. We believe that the low processing time of our cross-dimensional scheme along with its high-quality correspondence estimation, will afford a low drift odometry solution.

6 Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments.

7 References

[1] Tombari, F., Salti, S., Di Stefano, L.: 'Performance evaluation of 3D keypoint detectors', *Int. J. Comput. Vis.*, 2013, **102**, (1–3), pp. 198–220

[2] Guo, Y., Bennamoun, M., Sohel, F., *et al.*: 'A comprehensive performance evaluation of 3D local feature descriptors', *Int. J. Comput. Vis.*, 2016, **116**, (1), pp. 66–89

[3] Zhao, B., Chen, X., Le, X., *et al.*: 'A quantitative evaluation of comprehensive 3D local descriptors generated with spatial and geometrical features', *Comput. Vis. Image Underst.*, 2020, **190**, p. 102842

[4] Spezialetti, R., Salti, S., Di Stefano, L.: 'Performance evaluation of learned 3D features'. Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2019 (LNCS, **11751**), pp. 519–531

[5] Kechagias-Stamatis, O., Aouf, N., Dubanchet, V.: 'Evaluating 3D local descriptors and recursive filtering schemes for LIDAR-based uncooperative relative space navigation', *J. F. Robot.*, 2019, p. rob.21904

[6] Kechagias-Stamatis, O., Aouf, N., Richardson, M.A.: 'High-speed multi-dimensional relative navigation for uncooperative space objects', *Acta Astronaut..*, 2019, **160**, pp. 388–400

[7] Kechagias-Stamatis, O., Aouf, N., Richardson, M.A.: '3D automatic target recognition for future LIDAR missiles', *IEEE Trans. Aerosp. Electron. Syst.*, 2016, **52**, (6), pp. 2662–2675

[8] Filipe, S., Alexandre, L.A.: 'A comparative evaluation of 3D keypoint detectors in a RGB-D object dataset'. Proc. Ninth Int. Conf. on Computer Vision Theory and Applications, Lisbon, Portugal, 2014, pp. 476–483

[9] Hänsch, R., Weber, T., Hellwich, O.: 'Comparison of 3D interest point detectors and descriptors for point cloud fusion', *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, 2014, **II–3**, pp. 57–64

[10] Yang, J., Quan, S., Wang, P., *et al.*: 'Evaluating local geometric feature representations for 3D rigid data matching', *IEEE Trans. Image Process.*, 2019, **PP**, (8), p. 1

[11] Krizaj, J., Struc, V., Mihelic, F.: 'A feasibility study on the use of binary keypoint descriptors for 3D face recognition'. Mexican Conf. on Pattern Recognition, Cancun, Mexico, 2014, pp. 142–151

[12] Lowe, D.G.: 'Distinctive image features from scale invariant keypoints', *Int. J. Comput. Vis.*, 2004, **60**, (2), pp. 91–110

[13] Alahi, A., Ortiz, R., Vandergheynst, P.: 'FREAK: fast retina keypoint'. 2012 IEEE Conf. on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 510–517

[14] Bayramoglu, N., Alatan, A.A.: 'Shape index SIFT: range image recognition using local features'. 2010 20th Int. Conf. on Pattern Recognition, Istanbul, Turkey, 2010, pp. 352–355

[15] Tombari, F.: 'Keypoints and features'. Available at http://www.pointclouds.org/assets/uploads/cglibs13_features.pdf

[16] Wu, S., Oerlemans, A., Bakker, E.M., *et al.*: 'A comprehensive evaluation of local detectors and descriptors', *Signal Process. Image Commun.*, 2017, **59**, pp. 150–167

[17] Zhang, Y., Yu, F., Wang, Y., *et al.*: 'Performance evaluation of feature detection methods for visual measurements', *Eng. Lett.*, 2019, **27**, (2), pp. 320–327

[18] Kechagias-Stamatis, O., Aouf, N., Richardson, M.A.: 'Single and cross-dimensional feature detection and description: an evaluation', 2019, arxiv.org/abs/1910.08515

[19] Harris, C., Stephens, M.: 'A combined corner and edge detector'. Proc. Alvey Vision Conf. 1988, Manchester, UK, 1988, pp. 23.1–23.6

[20] Shi, J., Tomasi, : 'Good features to track'. Proc. of IEEE Conf. on Computer Vision and Pattern Recognition CVPR-94, Seattle, WA, USA, 1994, pp. 593–600

[21] Bay, H., Ess, A., Tuytelaars, T., *et al.*: 'Speeded-up robust features (SURF)', *Comput. Vis. Image Underst.*, 2008, **110**, (3), pp. 346–359

[22] Viola, P., Jones, M.: 'Robust real-time face detection', *Int. J. Comput. Vis.*, 2004, **57**, (2), pp. 137–154

[23] Rosten, E., Drummond, T.: 'Machine learning for high-speed corner detection'. European Conf. on Computer Vision, Berlin, Germany, 2006, pp. 430–443

[24] Leutenegger, S., Chli, M., Siegwart, R.Y.: 'BRISK: binary robust invariant scalable keypoints'. 2011 Int. Conf. Computer Vision, Barcelona, Spain, 2011, pp. 2548–2555

[25] Alcantarilla, P.F., Bartoli, A., Davison, A.J.: 'KAZE features'. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2012, pp. 214–227

[26] Zhong, Y.: 'Intrinsic shape signatures: A shape descriptor for 3D object recognition'. 2009 IEEE 12th Int. Conf. on Computer Vision Workshops, ICCV Workshops, Kyoto, Japan, 2009, pp. 689–696

[27] Mian, A.S., Bennamoun, M., Owens, R.: 'On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes', *Int. J. Comput. Vis.*, 2009, **89**, (2–3), pp. 348–361

[28] Chen, H., Bhanu, B.: '3D free-form object recognition in range images using local surface patches', *Pattern Recognit. Lett.*, 2007, **28**, (10), pp. 1252–1262

[29] Sun, J., Ovsjanikov, M., Guibas, L.: 'A concise and provably informative multi-scale signature based on heat diffusion', *Comput. Graph. Forum*, 2009, **28**, (5), pp. 1383–1392

[30] Unnikrishnan, R., Hebert, M.: 'Multi-scale interest regions from unorganized point clouds'. 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (IEEE, 2008), Anchorage, AK, USA, 2008, pp. 1–8

[31] Zaharescu, A., Boyer, E., Varanasi, K., *et al.*: 'Surface feature detection and description with applications to mesh matching'. 2009 IEEE Computer Society Conf. Computer Vision Pattern Recognition Work. CVPR Work. 2009, Miami, FL, USA, 2009, pp. 373–380

[32] Castellani, U., Cristani, M., Fantoni, S., *et al.*: 'Sparse points matching by combining 3D mesh saliency with statistical descriptors', *Comput. Graph. Forum*, 2008, **27**, (2), pp. 643–652

[33] Kechagias-Stamatis, O., Aouf, N., Gray, G., *et al.*: 'Local feature based automatic target recognition for future 3D active homing seeker missiles', *Aerosp. Sci. Technol.*, 2018, **73**, pp. 309–317

[34] Kechagias-Stamatis, O., Aouf, N., Nam, D.: '3D automatic target recognition for UAV platforms'. 2017 Sensor Signal Processing for Defence Conf. (SSPD), London, UK, 2017, pp. 1–5

[35] Yunqi, L., Haibin, L., Xutuan, J.: '3D face recognition by SURF operator based on depth image'. 2010 third Int. Conf. on Computer Science and Information Technology, Chengdu, People's Republic of China, 2010, pp. 240–244

[36] Frome, A., Huber, D., Kolluri, R., *et al.*: 'Recognizing objects in range data using regional point descriptors'. European Conf. on Computer Vision, Prague, Czech Republic, 2004, pp. 224–237

[37] Tombari, F., Salti, S., Di Stefano, L.: 'Unique shape context for 3d data description'. Proc. ACM workshop on 3D object retrieval – 3DOR '10', New York, NY, USA, 2010, p. 57

[38] Kechagias-Stamatis, O., Aouf, N.: 'Histogram of distances for local surface description'. 2016 IEEE Int. Conf. on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 2487–2493

[39] Kechagias-Stamatis, O., Aouf, N.: 'A new passive 3-D automatic target recognition architecture for aerial platforms', *IEEE Trans. Geosci. Remote Sens.*, 2019, **57**, (1), pp. 406–415

[40] Kechagias-Stamatis, O., Aouf, N.: 'Evaluating 3D local descriptors for future LIDAR missiles with automatic target recognition capabilities', *Imaging Sci. J.*, 2017, **65**, (7), pp. 428–437

[41] Salti, S., Tombari, F., Di Stefano, L.: ‘SHOT: unique signatures of histograms for surface and texture description’, *Comput. Vis. Image Underst.*, 2014, **125**, pp. 251–264

[42] Rusu, R.B., Blodow, N., Beetz, M.: ‘Fast point feature histograms (FPFH) for 3D registration’. 2009 IEEE Int. Conf. on Robotics and Automation, Kobe, Japan, 2009, pp. 3212–3217

[43] Guo, Y., Sohel, F., Bennamoun, M., *et al.*: ‘Rotational projection statistics for 3D local surface description and object recognition’, *Int. J. Comput. Vis.*, 2013, **105**, (1), pp. 63–86

[44] Guo, Y., Sohel, F., Bennamoun, M., *et al.*: ‘A novel local surface feature for 3D object recognition under clutter and occlusion’, *Inf. Sci. (Ny).*, 2015, **293**, pp. 196–213

[45] Johnson, A.E., Hebert, M.: ‘Using spin images for efficient object recognition in cluttered 3D scenes’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 1999, **21**, (5), pp. 433–449

[46] Zhao, B., Le, X., Xi, J.: ‘A novel SDASS descriptor for fully encoding the information of a 3D local surface’, *Inf. Sci. (Ny).*, 2019, **483**, pp. 363–382

[47] Zhou, W., Ma, C., Yao, T., *et al.*: ‘Histograms of Gaussian normal distribution for 3D feature matching in cluttered scenes’, *Vis. Comput.*, 2019, **35**, (4), pp. 489–505

[48] Guo, W., Hu, W., Liu, C., *et al.*: ‘3D object recognition from cluttered and occluded scenes with a compact local feature’, *Mach. Vis. Appl.*, 2019, **30**, (4), pp. 763–783

[49] Lim, J., Lee, K.: ‘3D object recognition using scale-invariant features’, *Vis. Comput.*, 2019, **35**, (1), pp. 71–84

[50] Lin, Y., Sun, Y., Min, H.: ‘A general gray code quantized method of binary feature descriptors for fast and efficient keypoint matching’. Proc. 2019 Second Int. Conf. Intellegent Autonomous Systems. ICoIAS 2019, Singapore, Singapore, 2019, (2), pp. 1–7

[51] Kechagias-Stamatis, O., Aouf, N., Chermak, L.: ‘B-HoD: A lightweight and fast binary descriptor for 3D object recognition and registration’. 2017 IEEE 14th Int. Conf. on Networking, Sensing and Control (ICNSC), Calabria, Italy, 2017, pp. 37–42

[52] Srivastava, S., Lall, B.: ‘Deeppoint3d: learning discriminative local descriptors using deep metric learning on 3D point clouds’, *Pattern Recognit. Lett.*, 2019, **127**, pp. 27–36

[53] Feng, M., Hu, S., Ang, M.H., *et al.*: ‘2D3D-matchnet: learning to match keypoints across 2D image and 3D point cloud’. 2019 Int. Conf. on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 4790–4796

[54] Carvalho, L.E., von Wangenheim, A.: ‘*3D object recognition and classification: a systematic literature review*’ (Springer, London, 2019)

[55] Yang, J., Xiao, Y., Cao, Z.: ‘Toward the repeatability and robustness of the local reference frame for 3D shape matching: an evaluation’, *IEEE Trans. Image Process.*, 2018, **27**, (8), pp. 3766–3781

[56] Munoz, D., Bagnell, J.A., Vandapel, N., *et al.*: ‘Contextual classification with functional max-margin Markov networks’. 2009 IEEE Conf. on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 975–982

[57] Mian, A.S., Bennamoun, M., Owens, R.: ‘Three-dimensional model-based object recognition and segmentation in cluttered scenes’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **28**, (10), pp. 1584–1601

[58] Yang, J., Xian, K., Wang, P., *et al.*: ‘A performance evaluation of correspondence grouping methods for 3D rigid data matching’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019, **PP**, (8), pp. 1–1

[59] Mouats, T., Aouf, N., Nam, D., *et al.*: ‘Performance evaluation of feature detectors and descriptors beyond the visible’, *J. Intell. Robot. Syst.*, 2018, **92**, pp. 33–63

[60] Muja, M., Lowe, D.G.: ‘Fast approximate nearest neighbors with automatic algorithm configuration’. Int. Conf. on Computer Vision Theory and Applications (VISAPP ‘09), Lisboa, Portugal, 2009, pp. 1–10