# A spatio-temporal multi-scale binary descriptor

Alessio Xompero, Oswald Lanz, Andrea Cavallaro

***Abstract*—Binary descriptors are widely used for multi-view matching and robotic navigation. However, their matching performance decreases considerably under severe scale and viewpoint changes in non-planar scenes. To overcome this problem, we propose to encode the varying appearance of selected 3D scene points tracked by a moving camera with compact spatio-temporal descriptors. To this end, we first track interest points and capture their temporal variations at multiple scales. Then, we validate feature tracks through 3D reconstruction and compress the temporal sequence of descriptors by encoding the most frequent and stable binary values. Finally, we determine multi-scale correspondences across views with a matching strategy that handles severe scale differences. The proposed spatio-temporal multi-scale approach is generic and can be used with a variety of binary descriptors. We show the effectiveness of the joint multi-scale extraction and temporal reduction through comparisons of different temporal reduction strategies and the application to several binary descriptors.**

## I. Introduction

LOCAL image features [1] are important for object recognition [2], retrieval [3][4], Structure-from-Motion [5], 3D reconstruction [6], visual Simultaneous Localisation and Mapping (SLAM) [7], and Collaborative SLAM [8]. To facilitate matching across views, these features describe the neighbourhood (or patch) of an interest point with a distinctive signature, which is designed to be invariant to scale, viewpoint and illumination changes, blur and compression artefacts [9].

Scale Invariant Feature Transform (SIFT) [2] and its variants [10][11][12][13] describe statistics of the patch and cope with challenging geometric variations. However, the extraction and matching of SIFT-like features, which accumulate gradient orientation information, are generally slow for real-time applications or resource-constrained devices. Efficient extraction and matching are instead obtained with binary signatures generated, for example, with a set of predefined comparisons (or binary tests) within the patch [14][15][16][17]. These binary features can be encoded ten times faster than SIFT-like descriptors and their representation is typically stored with only 32 bytes, whereas SIFT uses 128 bytes [13][14][18]. This gain in efficiency comes at the cost of a reduced robustness to geometric changes. For example, the accuracy of the widely used Oriented FAST and Rotated BRIEF (ORB) [15] features on the matching task of the HPatches dataset with viewpoint and illumination changes is 15.32%, whereas SIFT achieves 25.47%, on average [1].

To improve descriptiveness and robustness, local features can benefit from temporally accumulated information [19][20]. Spatio-temporal features can be extracted within a (fixed) temporal window [19] or by tracking local image features [7][20]. Cuboid [21], HOG/HOF [22], HOG3D [23], Extended SURF (eSURF) [24], and 3D-SIFT [25] are extracted in a fixed volume around an interest point localised in scale, space and time [19]. As the local temporal structure depends on the camera view, these features are mainly designed for in-camera tasks and are unsuitable for matching across cameras with considerable view changes [20]. Daisy-3D [20] ORB-SLAM [7], and STB [26] use tracking instead. Daisy-3D [20], which is obtained by tracking with optical flow priors and concatenating dense 2D Daisy features [27], is computationally expensive to track and match. ORB-SLAM [7] tracks ORB features and selects the ORB descriptor from the sequence of descriptors with the least median Hamming distance from all the other ORB descriptors. ORB-SLAM uses Bag of (Binary) Words [4] to match instances of ORB descriptors without exploiting information from the spatio-temporal feature. STB [26] encodes as binary representation the trajectory information as well as the horizontal and vertical components of the temporal gradient of a local spatio-temporal volume. Dense viewpoint- and illumination-invariant descriptors from models obtained with dense SLAM systems can be learned from RGB-D data [28] for indoor or well-structured scenes. However, the underlying SLAM system may fail outdoors due to inaccurate or incomplete depth information. Therefore, an important open problem for cross-camera matching is how to design a binary descriptor that is robust to severe changes in viewpoint and scale, while preserving efficiency.

In this paper, we propose a novel spatio-temporal binary descriptor that captures appearance variations of a 3D scene point as observed by a moving camera. This compact descriptor selectively encodes the temporal information associated with the 3D point to improve robustness to view differences. In particular, we propose a temporal reduction approach to encode the most frequent and stable binary values, so that the descriptor identifies temporally dominant values and the most stable tests over time. Moreover, to handle scale variations, the proposed descriptor relies on a multi-scale feature extraction and representation associated with a cross-scale matching strategy. Unlike [17], we augment the feature point detector with a feature suppression approach that increases scale invariance and leads to a more desirable spatially uniform feature distribution. Moreover, unlike [29], we use a pyramidal local search for feature point tracking [30] to increase the lifespan of feature tracks and to better capture appearance variations of the 3D point. The proposed descriptor is generic and we validate it with a range of binary descriptors. Moreover, we show that, on scenarios with challenging scale and viewpoint changes, the proposed approach outperforms alternative temporal reductions and the cross-camera matching based on Bag of (Binary) Words used in Collaborative SLAM [8].

## II. Background

Local image features may be generated with Convolutional Neural Networks (CNN) operating on patches or on the whole

frame; histogram representations of gradients or intensities of local patches; or binary descriptors. Binary descriptors result from hash or projection functions followed by thresholding, or from tests on pre-defined sampling patterns that are learnt [15][18][31][32], defined deterministically [16] or probabilistically [14].

*Patch-based CNN features* learn to discriminate correct and incorrect matches with supervised training. Examples include DeepDesc [33], DeepCompare [34], TFeat [35], and Multi-resolution CNN (MR-CNN) [36]. DeepDesc [33] and DeepCompare [34] train a Siamese network with pairs of annotated patches to push away incorrect patches and to move corresponding patches closer on a Euclidean, Hamming, or learnt metric. To reduce overfitting, TFeat [35] extends this network to triplets (anchor, positive sample, negative sample) and uses hard negative mining by swapping anchor and positive samples. To improve scale invariance, MR-CNN [36] learns a descriptor using image patches scaled at three resolutions as input to a three-stream Siamese network. However, TFeat outperforms MR-CNN in patch and image matching as well as in efficiency.

*Image-based CNN features* learn to localise and describe interest points on the whole image. Examples include Fully Convolutional Recursive Network - Patch Descriptor Network (FCRN-PDN) [37], Learned Invariant Feature Transform (LIFT) [38], Local Feature Network (LF-Net) [39], and Superpoint [40]. FCRN-PDN [37] learns to detect scale-invariant keypoints using a multi-scale branching mechanism within a fully convolutional recursive network. To assign a descriptor to the extracted patches, a second CNN is used that, similarly to TFeat, is trained with a triplet loss. Each network is trained independently in a self-supervised manner with data collected through SfM with aerial images at different scales. LIFT [38] uses an end-to-end network to learn detector, orientation estimator and descriptor in cascade starting from the descriptor stage. For learning the descriptor, LIFT extends TFeat to quadruplets, including an image patch with non distinctive information. The training is based on SIFT features and therefore LIFT has the limitations of SIFT. Instead of relying on supervised labelled data, LF-Net [39] uses ground truth camera poses and depth images to improve the learning of the end-to-end feature extraction pipeline. Superpoint [40] is a self-supervised approach that estimates interest point locations and associated descriptors directly on raw input images, assuming as model a homography. However, the training on synthetic images or real images with affine transformations does not guarantee its applicability to wide-baseline matching.

*Histogram-based features* include SIFT [2] and its variants [9][10][13][41][42], Speeded Up Robust Features (SURF) [43], Daisy [27], Local Intensity Order Pattern (LIOP) [44], Overall Intensity Order Pattern (OIOP) [45] and Mixed Intensity Order Pattern (MIOP) [45]. SURF [43] approximates the gradient with responses of Haar wavelets. Daisy [27] estimates convolutional oriented maps for each pixel with Gaussian filters and has a similar invariance to SIFT but a better efficiency for dense matching. Local Intensity Order Pattern (LIOP) [44], Overall Intensity Order Pattern (OIOP) [45] and Mixed Intensity Order Pattern (MIOP) [45] rank pixels in a patch according to their intensity value which is assigned to an intensity bin (ordinal cluster). LIOP [44] encodes the local ordinal information of each pixel by mapping the quantised intensities of corresponding neighbouring sampling points to a decimal code via a look-up table. OIOP [45] instead encodes the overall ordinal information by linearly combining the quantised values. The normalised histogram of the LIOP and OIOP codes are then computed for each ordinal cluster and concatenated to form the descriptor. MIOP [45] exploit the complementary information between the two descriptors at a reduced dimensionality (128 vs. 144/256 bytes) by applying PCA to the concatenation of LIOP and OIOP. LIOP, OIOP and MIOP outperform SIFT and Daisy [45].

*Binary features* are generated from comparisons of intensity values of pixels pairs, e.g. Binary Robust Invariant Elementary Features (BRIEF) [14], Oriented FAST and Rotated BRIEF (ORB) [15], Binary Robust Invariant Scale Keypoint (BRISK) [16], or Fast REtinA Keypoint (FREAK) [31]. Distinctiveness can be increased by extending comparisons to statistics of small windows pairs [32] or triplets [18]. ORB and FREAK learn the sampling pattern with a variance-correlation bit selection strategy. Local Difference Binary (LDB) [32] minimises the distance between pre-annotated matching interest points, whereas Learned Arrangements of Three patCH codes (LATCH) [18] maximises the distance of non-matching interest points. However, the efficiency gained with the pixel pair intensity comparisons comes at the cost of a reduced accuracy and robustness to geometric transformations and photometric variations, for example due to the changes in the binary values (instability) [51]. Binary Online Learned Descriptor (BOLD) [51] addresses this problem by selecting the most discriminative tests, e.g. after quantifying their stability to small geometric variations (e.g. scale or affine) in BRIEF [14]. The stability flag for each binary test is encoded as an additional binary vector. Other approaches use hash or projection functions and they threshold the resulting vector to obtain the final binary descriptor [48][49][50][54]. LDA-Hash [54] applies discriminative hash functions to SIFT descriptors followed by thresholding to obtain the binary representation. D-BRIEF [48] projects the patch intensities to a compact binary representation using a linear combination of box or Gaussian filters. Binboost [49] learns a set of hash functions that are the binary response of a boosting strong classifier built as a linear combination of weak classifiers. Receptive Field Descriptor (RFD) [50] learns a binary descriptor by first selecting the set of most discriminative receptive fields, defined as the aggregation of low-level filter responses within a patch, and then binarising the responses with learnt thresholds for each receptive field. These representations outperform histogram-based representations, e.g. SIFT, but are less efficient than early binary features and unsuitable for matching in time-constrained applications. Finally, DeepBit [52] is a CNN-based approach that learns a binary descriptor in an unsupervised manner: an image patch and its geometrically transformed version are given as input to a Siamese network to learn a set of projection functions to provide invariance to the transformations; enforce minimal quantisation error between the real-value deep feature and the binary code to increase the

TABLE I

LOCAL IMAGE AND SPATIO-TEMPORAL FEATURES. GRAY CELLS DENOTE PROPERTIES NOT HANDLED BY THE METHOD. KEY – REF: REFERENCE; ROT: ROTATION; DIST: DISTANCE USED FOR MATCHING DESCRIPTORS; UNS: UNSUPERVISED; MS: MULTI-SCALE; SI: SCALE INVARIANT; V: SPACE-TIME VOLUME; T: TRACKING; CNN: CONVOLUTIONAL NEURAL NETWORK; CONCAT.: CONCATENATION; L: LEARNT; E: EUCLIDEAN; H: HAMMING; W: WEIGHTED HAMMING; DOG: DIFFERENCE OF GAUSSIANS; GP: GAUSSIAN PYRAMID; RI: ROTATION INVARIANT; IC: INTENSITY CENTROID; LG: LOCAL GRADIENT; DD: DATA DEPENDENT; F: FLOATING POINT; B: BINARY. ×: DESCRIPTOR DIMENSION RESULTING FROM A CONCATENATION OPERATION OR A SET REPRESENTATION.

| | | **Detection** | | | **Description** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **REF** | **Method** | **Approach** | **Scale** | | **Approach** | **ROT** | **Scale** | **Time** | | **Dimension** | **Storage** | **DIST** | **UNS** |
| | | | MS | SI | | | MS | V | T | | | | |
| [34] | DeepCompare | | | | CNN with pairs of labelled patches | DD | | | | 256 | F | L | |
| [33] | DeepDesc | | | | CNN with pairs of labelled patches | DD | | | | 128 | F | E | |
| [35] | TFeat | | | | CNN with triplets of labelled patches | DD | | | | 128 | F | E | |
| [36] | MR CNN | | | | CNN with scaled patches (3 layers) | DD | ✓ | | | 128 | F | E | |
| [37] | FCRN-PDN | L | | ✓ | Scale branches det. + labelled patch triplets | DD | | | | 128/256 | F | E | |
| [38] | LIFT | L | | ✓ | CNN with quadruplets of patches | L | | | | 128 | F | E | |
| [39] | LF-Net | L | | ✓ | CNN with pair of images + depth | L | | | | 256 | F | E | ✓ |
| [40] | SuperPoint | L | ✓ | | CNN with homographic adaptation | DD | | | | 256 | F | E | ✓ |
| [2] | SIFT | DoG | | ✓ | gradient orientations in a regular grid | LG | | | | 128 | F | E | ✓ |
| [27] | Daisy | Dense | | | convolved orientation maps | | | | | 200 | F | E | ✓ |
| [44] | LIOP | | | | local ordinal intensities | RI | | | | 144 | F | E | ✓ |
| [45] | OIOP | | | | overall ordinal intensities | RI | | | | 256 | F | E | ✓ |
| [45] | MIOP | | | | concat. of LIOP with OIOP + PCA | RI | | | | 128 | F | E | ✓ |
| [46] | SLS | DoG | | ✓ | linear subspace of SIFTs | LG | ✓ | | | 8256 | F | E | ✓ |
| [11] | DSP-SIFT | DoG | | ✓ | pooling of SIFTs across scales | LG | ✓ | | | 128 | F | E | ✓ |
| [12] | ASV | DoG | | ✓ | SIFTs/LIOPs stability across scales | LG | ✓ | | | 128/144 | F | E | ✓ |
| [25] | 3D-SIFT | Random | | | 3D gradient orientations | LG | | ✓ | | 256/2048 | F | E | ✓ |
| [19] | HOG3D | STIP[47] | ✓ | | 3D-SIFT with polyhedrons | LG | | ✓ | | 960 | F | E | ✓ |
| [20] | Daisy-3D | Dense | ✓ | | concat. of Daisys with optical flow | | | | ✓ | $7 \times 136$ | F | E | ✓ |
| [14] | BRIEF | | | | random set of pixel pairs | | | | | 128/256/512 | B | H | ✓ |
| [15] | ORB | GP | ✓ | | learnt set of pixel pairs | IC | | | | 256 | B | H | ✓ |
| [16] | BRISK | GP | | ✓ | deterministic set of pixel pairs | LG | | | | 512 | B | H | ✓ |
| [32] | LDB | | | | learnt set of sub-patch pairs | IC | | | | 256 | B | H | |
| [18] | LATCH | | | | learnt set of sub-patch triplets | IC | | | | 128/256/512 | B | H | |
| [48] | D-BRIEF | | | | linear comb. of box/Gaussian filters | DD | | | | 32 | B | H | |
| [49] | BinBoost | | | | learnt set of hash functions (boosting) | DD | | | | 64 | B | H | |
| [50] | RFD | | | | selected receptive fields + learnt thresholds | DD | | | | 293/598 | B | H | |
| [51] | BOLD | | | | online selection of stability bits | | | | | 512 | B | W | ✓ |
| [17] | MORB | GP | ✓ | | set of ORBs across scales | IC | ✓ | | | $8 \times 256$ | B | H | ✓ |
| [52] | DeepBit | | | | CNN with min quantis. + max entropy loss | RI | | | | 256 | B | H | ✓ |
| [53] | CDBin | | | | lightweight CNN with triplet loss | RI | | | | 256 | B | H | |
| [7] | LMED | | | | ORB selection over time | IC | | | ✓ | 256 | B | H | ✓ |
| [26] | STB | | | | optical flow and temporal gradients encoding | IC | | ✓ | ✓ | 188 | B | H | ✓ |
| [29] | T-DS | | | | temporally reduced ORBs (centroid + stability) | IC | | | ✓ | 512 | B | W | ✓ |
| | **MST** | GP | | ✓ | set of temporally reduced ORBs across scales | IC | ✓ | | ✓ | $5 \times 512$ | B | W | ✓ |

descriptiveness (quantisation loss); and evenly distribute the binary code to maximise the information capacity (entropy) for each bin (even-distribution loss). CDbin [53], instead, uses a lightweight CNN to reduce the number of parameters and increase the efficiency of training and testing, outperforming other state-of-the-art binary descriptors. In addition to quantisation and even-distribution loss, CDbin uses a supervised triplet loss to increase the discriminative power and a correlation loss to reduce the correlation among different bits.

Table I summarises the approaches discussed in this section. CNN-based descriptors outperform both histogram-based and binary descriptors on standard patch verification, image matching, or patch retrieval datasets [1][9][55]. However, histogram-based methods, such as RootSIFT [10], outperform CNN-based descriptors in generalising across datasets and applications [5][6] without requiring any training. Therefore histogram-based descriptors are still preferable for their robustness to geometric challenges. The computation time of extracting and matching both CNN-based and histogram-based descriptors, however, makes them less suitable for time-constrained applications, unless GPU accelerations are used (e.g. GPU-SIFT [56] or TFeat [35]).

## III. LOCALISATION AND RECONSTRUCTION

We present a generic framework for binary descriptors that exploits the movement of a camera to selectively accumulate and encode temporal information about the appearance of a 3D point in a compact representation at multiple scales. To enable multi-scale extraction, we design a feature suppression strategy that simultaneously enforces scale-invariance and favours a spatially uniform distribution during localisation.

### A. Grid-based and scale-invariant feature point localisation

Let a local image feature represent the patch around image location $\mathbf{x} \in \mathbb{R}^2$ with a $D$-dimensional descriptor $\mathbf{d} \in \{0,1\}^D$. The number and spatial distribution of interest points over an image typically depends on a decision on the corner response [15][57]. However, using only the corner response can

(a) (b)

Fig. 1. Comparison of two feature point suppression approaches. (a) Using only the cornerness response results in few dense regions; whereas (b) using a regular grid and the cornerness response leads to a more uniform feature point distribution that is desirable when matching across very different viewpoints and scales.

result in an undesirable concentration of interest points, thus reducing opportunities for matches from different viewpoints and scales (see Fig. 1(a)). Moreover, when interest points are localised independently for each scale, redundancies can occur that generate ambiguities in the extracted descriptor [58]. To retain a maximum number of interest points without tuning the threshold of the corner response, we propose a suppression approach that simultaneously considers the corner response function to select the strongest points across nearby scales over a Gaussian pyramid (scale-invariance [2][16]), and a regular grid to enforce uniformity in the interest point distribution over the image [7] (see Fig. 1 (b)).

Let $\mathbf{I}_k$ be the frame at time $k$ and $\mathcal{I}_k = \{\mathbf{I}_{k,s}\}_{s=0}^{S-1}$ be its (scale) pyramid [2][15], where each layer $\mathbf{I}_{k,s}$ is recursively down-sampled by a factor $\lambda$, up to scale $S$, with a Gaussian convolutional kernel, $g(\cdot)$:

$$\mathbf{I}_{k,s}(\lambda^{-1}) = g(\lambda^{-1}) * \mathbf{I}_{k,s-1}. \tag{1}$$

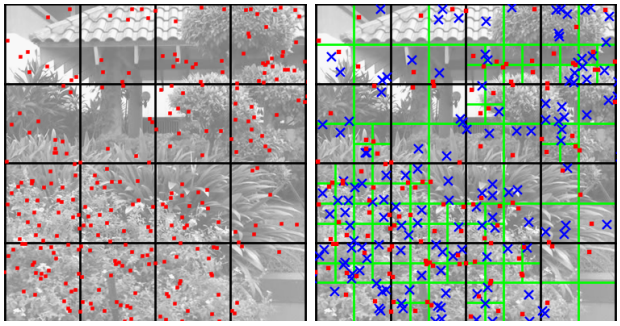
To allow the extraction of descriptors at multiple scales, we divide each $\mathbf{I}_{k,s}$ in a grid of $w \times w$ cells considering a scale-adaptive margin $B_s = \lambda^{S-s-1}G$ from the image borders, where $G \times G$ is the area around an interest point. We localise interest points with a good trade-off between repeatability and extraction time [15][57][58]. Next, we suppress non-maxima points across scales by comparing the response with the eight neighbours at the same scale and with the nine neighbours in the nearest scales [2][16].

As the Gaussian pyramid is obtained by following the terms of a geometric series as coefficients of proportionality based on the scale factor, $\lambda$, we proportionally distribute a number of localised interest points across scales, $F_s$, to determine a maximum number of features, $F$, as

$$F_s = \begin{cases} \frac{1-\lambda^{-1}}{1-\lambda^{-S+1}} F & \text{if } s = 0 \\ \lambda^{-1} F_{s-1} & \text{if } 0 < s < S-1 \\ \max\left(F - \sum_{q=0}^{S-2} F_q, 0\right) & \text{if } s = S-1, \end{cases} \tag{2}$$

where the resulting coefficients sum to 1.

Therefore we retain only $F_s$ interest points for each scale $s$ in an iterative way [7]. For each iteration, we sort the cells based on their feature density in an ascending order (cells without points are not considered). We then subdivide the cells that contain more than one interest point into four sub-cells and interest points are assigned to each sub-cell based on their location. The iterative procedure ends when the number of (sub-)cells is equal or greater than $F_s$ or all cells contain only one interest point. When a cell contains more than one interest point, we retain only the interest point with the highest corner response. Figure 2 illustrates the procedure for the retention of features based on their spatial distribution.



(a) (b)

Fig. 2. Feature suppression based on quadtree subdivision (suppressed features are represented with a blue cross). (a) Grid of cells superimposed on the image. (b) Cells with more than one point are split into four sub-cells (green blocks for each cell) and features (red dots) are accordingly assigned to each sub-cell. For each iteration, if the number of cells corresponds to the desired number of features, the features with higher Harris response [59] are retained in those cells that contain more than one feature. Cells without points are not counted for the desired number of features.

After localisation, we extract a descriptor for each interest point and then track the features. As our approach represents a 3D point associated to the trajectory of a feature, we will present our multi-scale spatio-temporal descriptor in Sec. IV. In the next subsection, we focus on how we form a feature track and reconstruct its 3D point.

### *B. Temporal feature point reconstruction*

Once feature points are localised and described, we estimate their trajectories over time. We use an iterative coarse-to-fine, local search by patch correlation through the scales of the image pyramid [30][60]. While we observed that frame-to-frame matching, as used during the initialisation of ORB-SLAM [7], generates intermittent feature tracks[1], the pyramidal local search allows feature tracks to survive longer. We reduce the risks of early termination by comparing the descriptor of the candidate features at the current frame with a reference descriptor selected adaptively as the one with the least median distance from all the descriptors in the feature track. We thus terminate the trajectory if the distance of the descriptors is larger than a threshold that represents the typical separation of matching and non-matching feature distributions in the space of the Hamming distances, e.g. $\gamma = 50$ [7][14]. As the camera moves, the number of matching features decreases over time and we detect new interest points every $n$ frames over a masked version of the frame where all the pixels around the locations of existing trajectories are set to zero. Then we initialise a new feature track for all the new interest points that are successfully tracked in the next frame.

[1]Frame-to-frame matching relies on the localisation strategy that selects different interest points for each frame and the matching is not constrained to a local area around each interest point in the previous frame.

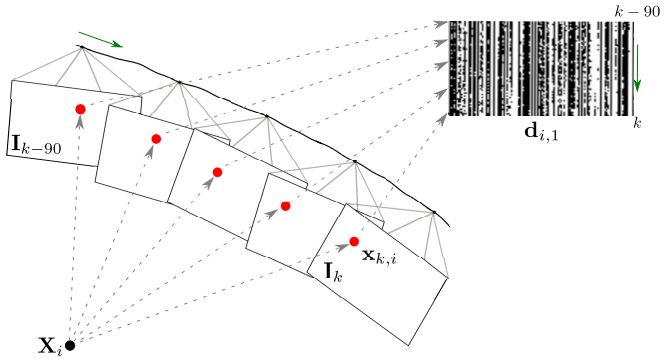



Fig. 3. Illustration of the accumulation of binary descriptors of a tracked feature point x representing a 3D point X.

Let us define the feature track as $\mathcal{T}_i = \{\mathbf{x}_{i,t_i}, \ldots, \mathbf{x}_{i,k_i}\}$, whose length is $L_i = k_i - t_i + 1$, where $t_i$ and $k_i$ are the indices of the first and last frame of the trajectory, respectively. Given the camera calibration information (e.g. obtained with Zhang's method [61]), we derive from $\mathcal{T}_i$ the position of $\mathbf{X}_i$ by N-view triangulation with Singular Value Decomposition (SVD) [62]:

$$\mathbf{X}_i = \tau(\mathbf{x}_{t_i,i}, \ldots, \mathbf{x}_{k_i,i}, \mathbf{C}_{t_i}, \ldots, \mathbf{C}_{k_i}, \theta), \tag{3}$$

where $\tau(\cdot)$ is the triangulation function; $\mathbf{C}_{t_i}, \ldots, \mathbf{C}_{k_i}$ are the relative camera poses (i.e. position and orientation, which we assume to be available through an Inertial Measurement Unit, Odometry, or Structure-from-Motion); and $\theta$ contains the intrinsic camera parameters, such as focal length and principal point (and distortion coefficients).

To account for uncertainties in the feature point detection, feature point tracking and triangulation steps, we validate the reconstructed 3D point with a maximum re-projection error of 5 pixels [5][7] and by constraining the depth to be positive [7][62].

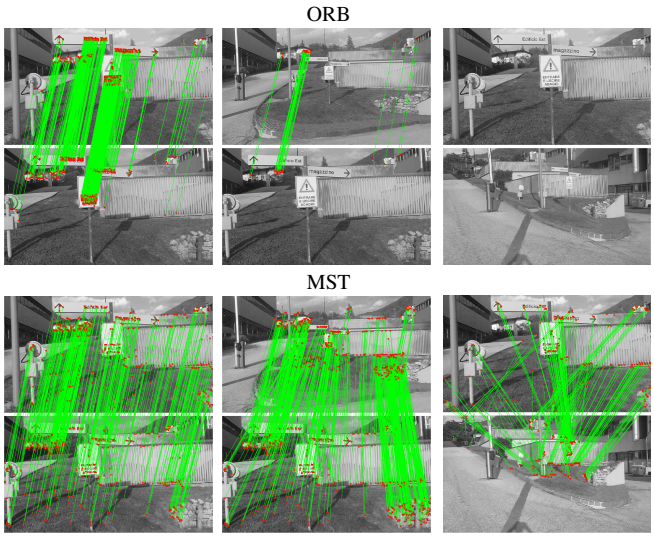
Fig. 3 illustrates the process of obtaining the binary spatio-temporal descriptor from a feature point $\mathbf{x}_i$ tracked over consecutive frames ($\mathcal{T}_i$), and representing the corresponding 3D point, $\mathbf{X}_i$.

## IV. Description Aggregation and Matching

The feature tracking and associated 3D local reconstruction produce valid spatio-temporal features that we temporally reduce into a fixed-length descriptor considering the most frequent and stable binary tests. To handle the unknown scale difference between features, we use a cross-scale matching strategy between multi-scale temporal descriptors with a weighted Hamming distance to consider the stability information.

### A. On handling scale variations

Descriptors of multi-scale approaches extracted *at the scale* where the interest point is localised [2][16][15] can be inaccurate when matching across images with severe scale variations [12]. Moreover, redundancies and ambiguities may arise if interest points are localised independently for each scale (e.g. ORB [15]), and can be avoided by suppressing non-maxima across scales [58] (e.g. SIFT [63] or BRISK [16]). Fig. 4 shows matching results with our approach compared to ORB features [15] with a similar viewpoint, different scale, and different viewpoint. Note that the number of ORB matches can be even lower if an interest point is associated with a reconstructed 3D point as our approach does.




(a) Similar (b) Scale difference (c) Viewpoint difference

Fig. 4. Matching performance of MST versus ORB [15] on *gate*. Green lines denote correct matches. When increasing the difference in scale (b) and viewpoint (c), the performance of ORB considerably decreases, while MST can handle the geometric variation. ORB features are matched using the nearest neighbour strategy with the distance ratio test [2], and a descriptor distance threshold. MST matches are based on the re-projection of the reconstructed 3D points in the selected frames and those that contains occluded points on the image are manually removed.

Descriptors can also be extracted at *multiple scales* of a Gaussian pyramid to capture multi-scale information of an interest point [11][12][46]. Coarser levels allow one to distinguish locally repeated patterns, whereas finer levels capture subtle changes thus helping to discriminate nearby points [36]. The Scale-less SIFT (SLS) descriptor [46] approximates SIFT descriptors [2] sampled at multiple scales with a linear subspace. Domain-Size Pooling SIFT (DSP-SIFT) [11] aggregates SIFT descriptors by pooling the values of each bin across scales. Accumulated Stability Voting (ASV) [12] thresholds the absolute difference between SIFT/LIOP descriptors of any pair of scales and accumulates the relative stability values into a compact representation. ASV selects one or multiple thresholds based on the principle of maximum entropy. In a second stage, ASV uses a further threshold to obtain a binary representation. These approaches inherit the same limitations of the histogram-based descriptors and are inadequate under severe viewpoint changes [17].

We therefore propose an alternative multi-scale and temporal extraction based on binary descriptors. Unlike our previous MORB descriptor [17] that applies a cross-scale geometric verification to remove ambiguities, we reconsider the feature suppression strategy to add scale-invariance [2] and uniform distribution of the features over the image [7]. While the efficiency in the extraction of the descriptor at multiple scales

decreases with the number of scales (a common limitation for SLS, DSP-SIFT, ASV, MORB, and our descriptor), using binary descriptors can mitigate this effect.

### B. Multi-scale temporal (MST) descriptor

We sample a patch around each point $\mathbf{x}_i$ of $\mathcal{T}_i$ at multiple scales with a pre-computed pattern $\mathcal{S}$ centred at $\mathbf{x}_{i,s,k}$ with $s = 1, \ldots, S$ and $k \in [t_i, k_i]$. To account for the rotation of the camera with respect to the 3D point, we rotate the patch towards the dominant orientation by $\varphi_{i,s,k}$ with respect to the centre of mass of the patch as defined by the intensity centroid [64]. We keep the size of the patch $G \times G$ fixed for each scale $s$ of $\mathcal{I}_k$ and define the sampling pattern as:

$$\mathcal{S} = \{\mathbf{u}_b = (\mathbf{u}_{b,1}, \mathbf{u}_{b,2})\}_{b=1}^{D}, \tag{4}$$

where $\mathbf{u}_{b,1}$ and $\mathbf{u}_{b,2}$ are pixel locations within the patch. After sampling using the rotated pattern:

$$\tilde{\mathcal{S}}_{i,s,k} = \{\mathbf{R}(\varphi_{i,s,k})\mathbf{u} : \mathbf{u} \in \mathcal{S}, \mathbf{R} \in SO(2)\}, \tag{5}$$

we generate the descriptor $\mathbf{d}_{i,s,k} \in \{0,1\}^D$, whose elements result from the binary test [14][15][16][31]:

$$\mathbf{d}_{i,s,k}(\mathbf{u}_b) = \begin{cases} 1 & \text{if } \mathbf{I}(\mathbf{u}_{b,1}) < \mathbf{I}(\mathbf{u}_{b,2}), \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The descriptor $\mathbf{d}_i$ thus represents a set of patch descriptors at multiple scales and over time:

$$\mathbf{d}_i = \{\mathbf{d}_{i,1,t_i}, \ldots, \mathbf{d}_{i,S,t_i}, \ldots, \mathbf{d}_{i,1,k_i}, \ldots, \mathbf{d}_{i,S,k_i}\}. \tag{7}$$

We propose to represent the interest point with a more compact and fixed-length descriptor that captures the most representative tests of each 3D point as seen by a moving camera (see Fig. 5).

For each scale $s$, we reduce $\mathbf{d}_{i,s}$ to a fixed-length vector $\mathbf{z}_{i,s} \in \{0,1\}^D$ by accumulating the binary test values over time and identifying the dominant binary value as

$$z_{i,s,d} = \begin{cases} 1 & \text{if } \frac{1}{L_i}\langle \mathbf{d}_{i,s,d}, \mathbf{1}\rangle > 0.5, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $\mathbf{d}_{i,s,d} \in \{0,1\}^{L_i}$ is the vector containing the temporal values of the element $d$, $\langle \cdot, \cdot \rangle$ is the (logical) dot product and 0.5 is the prior probability of the binary test being 1.

To account for noise during the temporal matching due to photometric and/or geometric changes, we allow some variations in the binary test outcome, at a rate lower than 20% of the length of the feature track (i.e. if the minimum length of a feature track is 5 frames, we allow only one change for each binary test). We then compute a second vector, $\mathbf{d}'_{i,s} \in \{0,1\}^{(L_i-1)\times D}$, that captures the temporal changes, i.e. instability, of the binary tests in $\mathbf{d}_{i,s}$ via a bit-wise XOR of two consecutive binary descriptors. As for $\mathbf{z}_{i,s}$, we reduce $\mathbf{d}'_{i,s}$ to $\mathbf{m}_{i,s} \in \{0,1\}^D$ as:

$$m_{i,s,d} = \begin{cases} 1 & \text{if } \frac{1}{L_i-1}\langle \mathbf{d}'_{i,s,d}, \mathbf{1}\rangle \leq 0.2, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

We refer to $\mathbf{z}_{i,s}$ and $\mathbf{m}_{i,s}$ as the vector of temporally dominant bits and the vector of temporally stable bits, respectively.

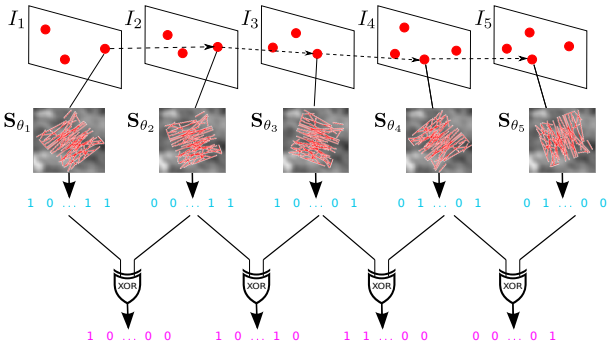



Fig. 5. Extraction of the temporal binary descriptor at a single scale. The location of the interest point in the first frame is tracked in successive frames. For each frame the rotated sampling pattern is extracted forming a set of binary vectors (cyan). We then compute the derivative (XOR operation) between consecutive binary vectors to estimate a second set of binary vectors (magenta) containing the frame-to-frame stability. For each set, we sum the vectors followed by thresholding to obtain the vector of the most frequent binary values and the vector of the most stable tests over time, respectively.

The dimensionality of the MST descriptor, $2 \times D \times S$, depends on the chosen number of scales, $S$, the length of the vector of temporally dominant bits and the vector of temporally stable bits, $2 \times D$. Note that $D$ depends on the dimensionality of the specific employed image-based binary descriptor. Moreover, the total number of binary tests performed by MST depends on the length, $L_i$, of the feature trajectory. For example, considering 5 scales, a binary descriptor such as ORB ($D = 256$), and a maximum length of 50 frames, the minimum number of binary tests is 6400 (the maximum is 64000), and the dimensionality of MST is 2560 bits.

### C. Matching

After estimating the multi-scale temporal descriptors, we aim to find a set of matches across cameras. As the scale at which features should be matched is unknown, we cannot directly apply for matching nearest neighbour [9] or bag of words [4]. For this reason we estimate the minimum cross-scale distance between feature pairs. To this end, let us introduce $\alpha$ and $\beta$ as indices of two cameras.

We first remove temporally unstable bits of $\mathbf{z}^{\alpha}_{i,s}$ and $\mathbf{z}^{\beta}_{j,l}$ (see Fig. 6) using a weighted Hamming distance [51]. Let the masked Hamming distance using only $\mathbf{m}^{\alpha}_{i,s}$ be defined as

$$\langle \mathbf{m}^{\alpha}_{i,s}, \mathbf{z}^{\alpha}_{i,s} \oplus \mathbf{z}^{\beta}_{j,l} \rangle, \tag{10}$$

where $\oplus$ is the XOR operator. Let the number of stable binary tests for $\mathbf{z}^{\alpha}_{i,s}$ be defined as

$$N^{\alpha}_{i,s} = \langle \mathbf{m}^{\alpha}_{i,s}, \mathbf{1} \rangle \tag{11}$$

and, similarly, $N^{\beta}_{j,l}$ for $\mathbf{z}^{\beta}_{j,l}$.

We then compute the dissimilarity, $h_{i,j}(s,l)$, between descriptor pairs as:

$$h_{i,j}(s,l) = \frac{N^{\alpha}_{i,s}\langle \mathbf{m}^{\alpha}_{i,s}, \mathbf{z}^{\alpha}_{i,s} \oplus \mathbf{z}^{\beta}_{j,l} \rangle + N^{\beta}_{j,l}\langle \mathbf{m}^{\beta}_{j,l}, \mathbf{z}^{\alpha}_{i,s} \oplus \mathbf{z}^{\beta}_{j,l} \rangle}{N^{\alpha}_{i,s} + N^{\beta}_{j,l}}, \tag{12}$$

and identify the minimum across scales, $\mathcal{S} = \{1, \ldots, S\}$, as

$$h_{i,j}(s^*, l^*) = \min_{s,l \in \mathcal{S}} h_{i,j}(s,l), \tag{13}$$

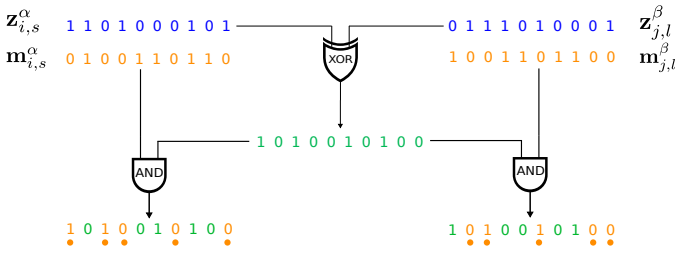



Fig. 6. Graphical representation of the descriptor matching using the temporally dominant and temporally stable vectors at a single scale between descriptors of two different cameras. After selecting the stable bits of the resulting difference vector between the temporally dominant vectors, the weighted Hamming distance is applied (see Eq.12 [1]). Unstable bits are denoted with an orange dot.

and $|s^* - l^*|$ is the scale offset between the interest points.

To remove possible ambiguities, we determine the final set of matches through nearest neighbour followed by the Lowe's ratio test that validates a match only if the similarity distance of the closest neighbour is sufficiently lower than the distance from the second nearest neighbour [2][9].

### D. Discussion

The proposed MST descriptor handles scale and viewpoint differences by exploiting multi-scale extraction with a grid-based and scale-invariant suppression strategy, and temporal variations obtained by tracking local binary image features. As our approach encodes the temporal information of feature trajectories in a compact descriptor, we differ from Daisy-3D [20], which concatenates tracked 2D Daisy features in a fixed window thus limiting the amount of information and variations captured by the spatio-temporal feature and requiring an expensive matching approach between cameras. We also differ from LMED [7], which uses the ORB binary descriptor and selects the single descriptor over time with the least median distance with respect to all the tracked ORB descriptors within the feature trajectory. While the chosen descriptor can reduce drifts in the feature tracks, this descriptor may not be suitable when matching features across cameras. Unlike STB [26], which describes the trajectory and temporal gradients of a fixed-size spatio-temporal volume, we obtain varying-length spatio-temporal features by directly accumulating image-based binary features, followed by a reduction to a compact, fixed-size representation. Moreover, we handle scale differences when matching different views through multi-scale extraction and representation. Finally, unlike BOLD [51], which computes the stability vector with small geometric variations of the sampling pattern, we determine the stability by exploiting the temporal variations within a feature track. The stability is thus used as a selector when computing the distance between MST descriptors. The proposed framework is validated in the next section.

## V. Validation

### A. Experimental setup

We compare our proposed spatio-temporal descriptor, MST against the method based on ORB-SLAM [7] for the processing of each sequence (feature track extraction and descriptor reduction) and a matching with the Bag of Binary Words (e.g. DBoW2 [4]). We also compare MST against (i) SetDesc, the set of image-based binary descriptors of a feature track without reduction; (ii) T-D, extracted at a single scale with a reduction of the set of binary descriptors with only the temporally dominant bit approach [29], (iii) T-DS, which complements T-D with a vector that contains the temporally stable bits [29], (iv) LMED, which selects the single binary descriptor from SetDesc that has the least median distance compared to all other descriptors within the feature track [7], and (v) MST-S, which corresponds to our spatio-temporal descriptor without the stability vectors.

To fairly compare all the descriptors, we obtain feature tracks with our approach and we then compute the corresponding descriptors. We use the most suitable dissimilarity measure for each descriptor when matching features. For SetDesc, we find the minimum Hamming distance between all possible pairs of single descriptors between the sets of descriptors belonging to two different sequences (*set2set min dist* [65]). However, as finding the minimum across both scales and time is computationally expensive, we extract and match the sets only at the original scale. Note that we expect SetDesc extracted also at multiple scales to achieve higher matching performance than without scale. For T-D and LMED, we use the standard Hamming distance as dissimilarity measure, while we use the weighted Hamming distance (Eq. 12) for T-DS and MST as their descriptors contain the additional stability vector. Finally, we consider the cross-scale matching approach between single-scale descriptor pairs for MST-S and MST.

We not only apply the binary spatio-temporal features to a range of image-based binary descriptors, namely ORB [15], BRIEF [14], LDB [32], LATCH [18], RFD [50], and the learned, CNN-based DeepBit [52], but we also adapt the overall framework to histogram-based descriptors, and include SIFT [2] as an example. In this latter case, we replace the intensity centroid method [64] with the SIFT orientation assignment [2]. When tracking features, after estimating the average SIFT descriptor within the trajectory, up to frame $k-1$, we estimate the ratio between the distance of the current SIFT with the average descriptor and the distance of the SIFT descriptor at the localised frame and the average descriptor. Following the inverse of the Lowe's distance ratio test [2], we terminate a trajectory if its ratio is lower than 0.6. For SIFT, we apply SetDesc and T-D as spatio-temporal features. Note that T-D corresponds to the average SIFT over time (similarly to the representation used in SfM pipelines [5]).

We use pairs of sequences, captured with hand-held cameras, from publicly available datasets: TUM-RGB-D SLAM [66]; *courtyard*[2] [67]; and *gate*, a dataset we collected and make available to the research community. From TUM-RGB-D SLAM we use two clips of 50 frames (640×480 pixels) with sufficient overlap from *desk* (with similar motion) and *office* (cameras move in opposite directions). From the first and fourth video of *courtyard*, we select the first 50 frames (800×450 pixels) after sub-sampling the videos from 50 to 25 fps. We select the first 100 frames (1280×720 pixels) of

[2]drone.sjtu.edu.cn/dpzou/project/coslam.php, accessed: March 2018


Fig. 7. A sample frame for each sequence of the four sets. Note the differences in viewpoint and/or scale between sequences within the same set.

the four sequences of *gate* after down-sampling the video to 10 fps from 30 fps. We pair the first sequence with each of the other three sequences and we refer to each pair as *gate-1*, *gate-2*, *gate-3*, respectively. Fig. 7 shows a frame for each sequence.

### B. Parameter settings and choices

We set the parameters using values from related works or corresponding implementations: the FAST threshold is 25 [57], the block size for the grid is $w = 30$ [7]. To extract the multi-scale descriptor, we consider a pyramid of $S = 5$ scales [15] with a scale factor $\lambda = 1.15$. The patch size depends on the chosen image-based binary descriptor (e.g. $G = 31$ for ORB [15]). Features are tracked with the pyramidal Kanade-Lucas-Tomasi tracker [30] available in OpenCV using a window size of 21 pixels, 5 scales and maximum 30 iterations. We discard tracked features whose distance from the image boundaries at the coarsest level is less than half of $G$, which ensures the extraction of the descriptors at multiple scales. To reduce uncertainty in the triangulation, we enforce the feature track to be at least 5 frames long assuming that there is enough camera motion (translation) [62]. In addition, we set the radius of the non-maxima suppression for the grid-based detection to 3 pixels; and we detect new interest points every $n = 5$ frames[3] using a $7 \times 7$ masking window around the location of each existing feature track.

### C. Performance measures

We quantify the number of correct matches over the number of estimated matches (precision, $P$); the number of correct matches over the number of ground truth correspondences (recall, $R$); and their harmonic mean ($F_1$ score):

$$F_1 = 2 \times \frac{P \times R}{P + R}, \tag{14}$$

and the average matching time per descriptor pair to evaluate the different spatio-temporal approaches.

$P$ and $R$ are generally used for features between image pairs with known ground truth homographies [1][9][69]. Therefore to compute $P$ and $R$ for feature trajectories, we annotate reference correspondences of feature tracks using multi-view geometry [62] and reconstruct the 3D point associated to each feature track using the absolute camera poses. We then geometrically verify that the projection of the point into the second view is within the image borders for at least five frames. Next we compute the root mean square residual (RMS) between feature track pairs from the two views and validate only pairs whose RMS is smaller than 5 pixels. We determine the number of unique correspondences (i.e. one feature track cannot be paired with more than one in another view) using the nearest neighbour approach. Then, we consider the nearest neighbour with the Lowe's ratio test [2][9] as similarity matching strategy between spatio-temporal descriptors to compute precision, recall and $F_1$ score. The threshold for the ratio test is 0.8 [7]. We determine correct matches as those matches whose RMS is smaller than 5 pixels. This ground truth (calibration data and camera poses) is available with the dataset. For *courtyard* and *gate*, the ground truth is annotated using COLMAP [5].

[3]We aim to reproduce the automatic keyframe selection strategy of Visual SLAM/Odometry methods (around 5-10 keyframes per second) [7][68].

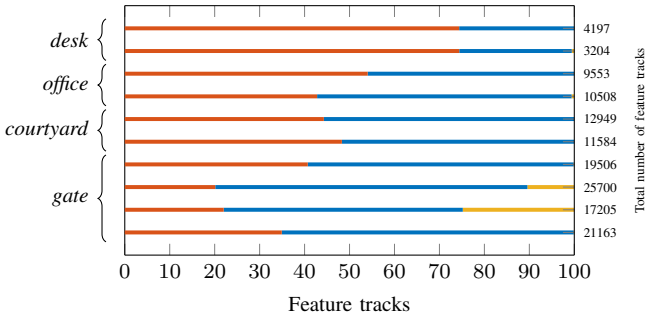



Fig. 8. Percentage of surviving feature tracks (■) after discarding tracks whose length is shorter than five frames (■) or invalidated by geometric tests (■). Note that the number of feature tracks invalidated by the geometric tests are less than 0.1% for most of the sequences.

### D. Results

Fig. 8 shows the percentage of survived trajectories, discarded feature tracks because of the short length, and tracks discarded by geometric tests. The total number of feature tracks is ∼135000. The high number of feature tracks denotes the frequent re-localisation of many interest points. Because of the short length, the method discards more than 50% of the feature tracks in most of the sequences except *desk* where the camera moves slowly. In *gate-3*, the geometric tests invalidate ∼25% of the feature tracks due to camera shaking.

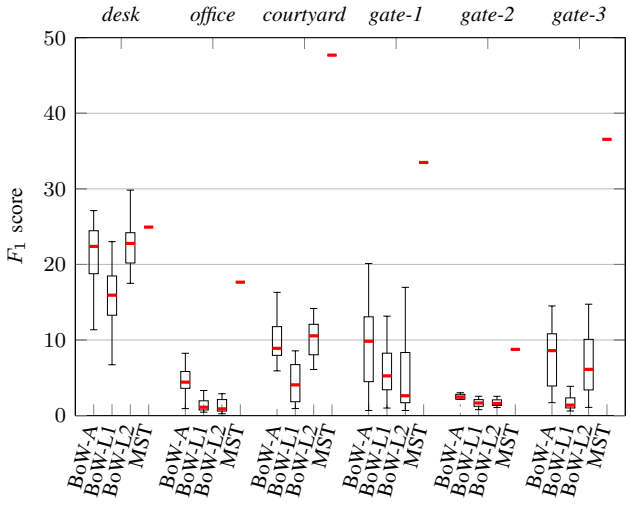



Fig. 9. $F_1$ score comparison between MST and the BoW approach. For each sequence pair, we show three cases for BoW. BoW-A: the best image match is estimated among all keyframes of both cameras. BoW-L2: the best image match is estimated between the last keyframe of the second camera against all the keyframes of the first camera; BoW-L1: the best image match is estimated between the last keyframe of the first camera against all the keyframes of the second camera. BoW-based matching results are obtained by running ORB-SLAM [7] over 30 runs for both camera sequences simultaneously.

Fig. 9 compares the $F_1$ score of our proposed method, MST, against BoW, the cross-camera matching based on ORB-SLAM and the Bag of Visual Binary Words [4][7]. We consider three variants of BoW: all the keyframes of camera 1 are compared against all the keyframes of the second camera 2 (BoW-A); the last keyframe of camera 1 is compared against all the keyframes of camera 2 (BoW-L1); and the last keyframe of camera 1 is compared against all the keyframes of camera 2 (BoW-L2). The last two variants recall scenarios where only one keyframe (usually the last) is sent/received by each camera [8][70]. To account for the non-deterministic nature of ORB-SLAM, we run ORB-SLAM 30 times for each sequence using the same settings of our approach. While BoW creates a feature vector using all the local features of a frame, the matching within ORB-SLAM limits the valid matches to features with a corresponding 3D point, similar to our MST. MST outperforms BoW on all sequence pairs. In *desk* where geometric variations are small, MST slightly outperforms BoW, while the benefit of our approach is clearly visible in *courtyard*, *gate-1* and *gate-3* where geometric differences are more challenging. In the most severe viewpoint differences of *office* and *gate-2*, MST outperforms BoW by more than 10% and 5%, respectively. Note that in *gate-2* the two cameras approach the same point of the scene from different viewpoints.

Fig. 10 compares the matching performance when varying $F$ to quantify the impact of the number of feature points localised in the first frame or during the re-detection. For *courtyard*, *office*, and *gate-3*, MST outperforms other descriptors independently of the number of localised features. In *desk*, where the scene has low texture and small geometric variations, SetDesc achieves the best performance. When $F = 500$, the performance of MST-S and MST is close to SetDesc, while

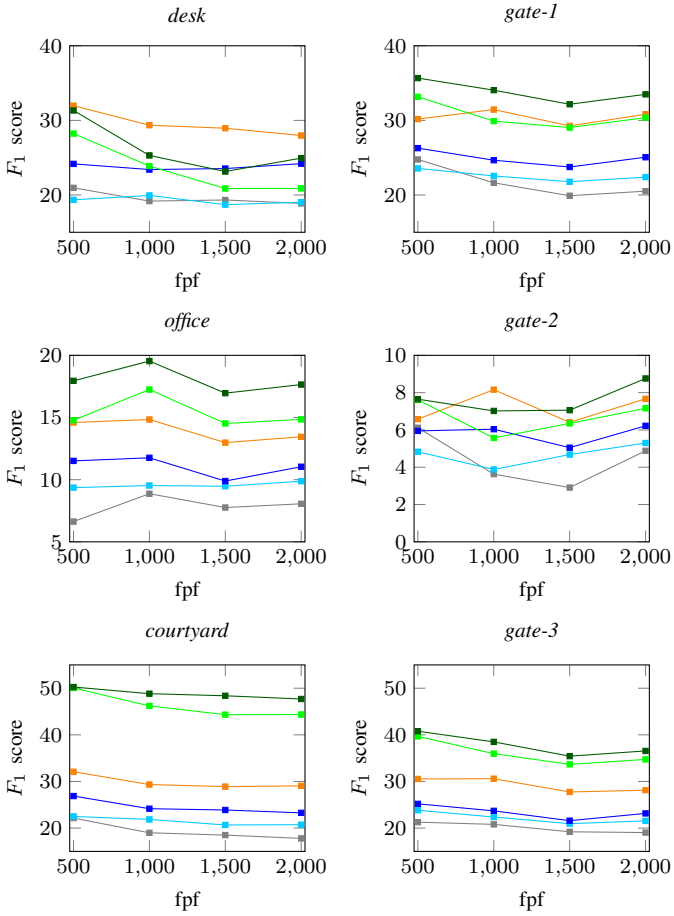



Fig. 10. Accuracy ($F_1$ score) when varying the maximum number of features per frame (fpf) using ORB [15]. The fpf depends on the localisation in the first frame and re-detection. Note the different scales of the vertical axes. Legend: SetDesc, LMED, T-D, T-DS, MST-S, MST.

when increasing $F$ the performance of MST converges to that of T-DS, suggesting that the multi-scale is not important in this scenario. We can also observe that, unlike the behaviour of the other approaches, the performance of SetDesc increases when $F = 1000$ in *gate-2* and *gate-1*. In *gate-2* SetDesc achieves the highest $F_1$ score. Overall, we can observe that increasing the maximum number of features localised or re-detected does not result in an increase of the performance, but on the contrary the performance tends to decrease in most of the sequence pairs for most of the approaches. For fair comparison, we do not fine-tune the number of features and we set $F = 2000$ across all sequences for the last comparison results.

Table II compares the matching performance of spatio-temporal descriptors extracted from the feature tracks. The number of reference correspondences is 1280 for *desk*, 2623 for *office*, 3448 for *courtyard*, 2427 for *gate-1*, 1357 for *gate-2*, 2378 for *gate-3*. We can observe that the additional stability vector of T-DS and MST leads to higher recall but lower precision than T-D and MST-S. Moreover, the multi-scale representation, MST-S and MST, allows to improve the performance of the proposed temporal reduction, i.e. T-D and T-DS. MST outperforms other approaches in terms of recall across all sequence pairs except *desk* that contains

TABLE II
MATCHING RESULTS WITH THE NEAREST NEIGHBOUR STRATEGY AND LOWE'S RATIO TEST USING ORB FEATURES. BEST RESULTS IN BOLD, SECOND BEST IN ITALIC. LEGEND: $M$: NUMBER OF MATCHES. $P$: PRECISION. $R$: RECALL. $F_1$: $F_1$ SCORE.

| | *desk* | | | | *office* | | | | *courtyard* | | | | *gate-1* | | | | *gate-2* | | | | *gate-3* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $M$ | $P$ | $R$ | $F_1$ | $M$ | $P$ | $R$ | $F_1$ | $M$ | $P$ | $R$ | $F_1$ | $M$ | $P$ | $R$ | $F_1$ | $M$ | $P$ | $R$ | $F_1$ | $M$ | $P$ | $R$ | $F_1$ |
| SetDesc | 444 | **54.28** | **18.84** | **27.97** | 560 | *38.21* | 8.16 | 13.45 | 853 | 73.27 | 18.13 | 29.06 | 895 | **57.21** | *21.10* | *30.82* | 338 | **19.23** | *4.79* | *7.67* | 880 | *52.05* | 19.26 | 28.12 |
| LMED | 321 | *47.04* | 11.81 | 18.87 | 453 | 27.37 | 4.73 | 8.06 | 632 | 57.44 | 10.53 | 17.79 | 693 | 46.18 | 13.19 | 20.51 | 282 | 14.18 | 2.95 | 4.88 | 668 | 43.41 | 12.20 | 19.04 |
| T-D | 328 | 46.65 | 11.96 | 19.04 | 454 | 33.48 | 5.79 | 9.88 | 671 | 63.64 | 12.38 | 20.73 | 700 | 50.00 | 14.42 | 22.39 | 265 | 16.23 | 3.17 | 5.30 | 741 | 45.34 | 14.13 | 21.55 |
| T-DS | 481 | 44.28 | 16.65 | 24.20 | 692 | 26.45 | 6.98 | 11.04 | 1021 | 50.93 | 15.08 | 23.27 | 1036 | 41.89 | 17.88 | 25.06 | 508 | 11.42 | 4.27 | 6.22 | 1112 | 36.33 | 16.99 | 23.15 |
| MST-S | 388 | 44.85 | 13.60 | 20.88 | 541 | **43.44** | *8.96* | *14.85* | 1214 | **85.17** | *29.99* | *44.36* | 892 | 56.50 | 20.77 | 30.37 | 319 | 18.81 | 4.42 | 7.16 | 1095 | **55.07** | *25.36* | *34.73* |
| MST | 533 | 42.40 | *17.67* | *24.94* | 834 | 36.57 | **11.63** | **17.65** | 1610 | *74.91* | **34.98** | **47.69** | 1293 | 48.18 | **25.67** | **33.49** | 584 | 14.55 | **6.26** | **8.76** | 1562 | 46.09 | **30.28** | **36.55** |

TABLE III
DIFFERENCE BETWEEN THE $F_1$ SCORE (%) OF SPATIO-TEMPORAL FEATURES WHEN EXTRACTING FEATURE TRAJECTORIES WITH AND WITHOUT 3D GEOMETRIC TESTS.

| **Method** | **Sequence pair** | | | | | |
|---|---|---|---|---|---|---|
| | *desk* | *office* | *courtyard* | *gate-1* | *gate-2* | *gate-3* |
| SetDesc | .00 | -.12 | -.01 | .00 | -.03 | .10 |
| LMED | -.02 | -.01 | -.08 | -.01 | -.01 | -.16 |
| T-D | .00 | .00 | .00 | .00 | .00 | .00 |
| T-DS | -4.80 | -3.00 | -3.10 | -3.90 | -5.20 | -6.60 |
| MST-S | -1.70 | -.89 | -.75 | -1.40 | -1.50 | -2.60 |
| MST | -.21 | -.01 | -.15 | -.16 | -.09 | -.26 |

sequences with limited motion in the same direction and similar viewpoint in an indoor environment with low texture. The higher recall also influences the performance of the $F_1$ score except for *gate-2* where the stability vector allows to estimate almost twice the number of matches with several false positives (85/584 for MST vs 60/319 for MST-S), considerably decreasing the precision. We can observe that due to the severe change in viewpoint between the cameras, *office* and *gate-2* are the most challenging sequence pairs with recall lower than 12% for all approaches.

We now evaluate the spatio-temporal features without the geometric tests, but still filtering out short feature tracks (see Fig. 8). Tab. III shows, for each spatio-temporal feature and for each sequence pair, the difference between the $F_1$ score with and without geometric tests. We can observe that adopting the geometric tests has a minimal impact on the accuracy for SetDesc, LMED, T-D, and MST across all sequence pairs, while T-DS and MST-S are the most sensitive to this step as their accuracy decreases up to 6% and less than 3% in $F_1$ score, respectively.

Fig. 11 shows correct matches obtained with MST. Reconstructed 3D points are re-projected in pairs of selected frames for *gate-1*, *gate-2*, and *gate-3* with changes in both scale and viewpoint. Fig. 12 quantifies the maximum viewpoint angle for MST features when estimated within each sequence and when correctly matched across cameras, for all sequence pairs. The viewpoint angle is computed using the cosine formula between the reconstructed 3D point and two camera locations, where the 3D point is observed. For each MST feature, we estimate the angle between each pair of views where the corresponding 3D point is visible, and we then find the view pair with the maximum angle. Most trajectories can handle up to 10 degrees of viewpoint difference, while there are features that

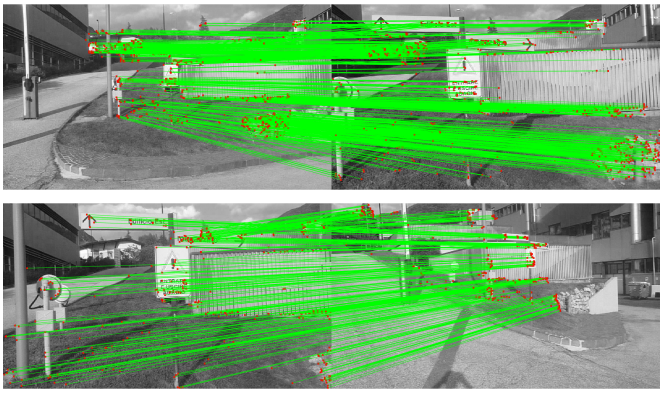


Fig. 11. Correct matches (green lines) with MST by re-projecting the 3D points (red dots) in a selected pairs of frames. Top: scale difference in *gate-3*; bottom: viewpoint and scale difference in *gate-1*.

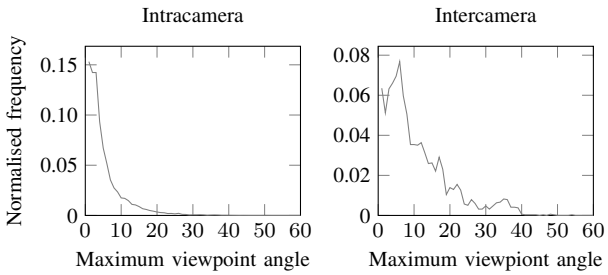



Fig. 12. Distribution of the maximum viewpoint angle (in degrees) for MST features within each camera for all testing sequences (on the left, intracamera), and for correct matching MST features between cameras for all sequence pairs (on the right, intercamera). The maximum viewpoint angle corresponds to the angle of the pair of views that is maximum across all the possible view pairs where the 3D point corresponding to the spatio-temporal feature is visible. Note the different scale of the y-axis. The total number of estimated MST features is 50791. The total number of correct matches is 3165.

can handle differences of up to 30 degrees. As feature tracking is performed with a validation strategy based on the image-based binary descriptor, the maximum viewpoint variation is constrained by the limitation in the geometric variations of the descriptor itself (e.g. ORB is not robust to viewpoint differences). The distribution of the maximum viewpoint angle for correctly matched MST features across cameras shows that the proposed approach can handle differences of up to 40 degrees.

Fig. 13 shows the average $F_1$ scores and matching times, while Table IV shows the total time to match the descriptors for each sequence pair. The total matching time depends on the

TABLE IV
TOTAL MATCHING TIME FOR EACH SEQUENCE PAIR AND FOR EACH SPATIO-TEMPORAL FEATURE (IN SECONDS). NOTE THAT SIFT AND ORB ARE NOT COMPARABLE DUE TO THE DIFFERENT NUMBER OF FEATURE TRAJECTORIES FOR EACH SEQUENCE. OBSERVE THE COMPARISON BETWEEN SPATIO-TEMPORAL FEATURES FOR EACH ROW.

| Seq. pair | Desc. | #FT | | Total matching time (s) | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Seq1 | Seq2 | SetDesc | LMED | T-D | T-DS | MST-S | MST |
| *desk* | SIFT | 1486 | 658 | 46 | | 10 | | | |
| | ORB | 1198 | 1005 | 55 | 13 | 14 | 14 | 16 | 23 |
| *office* | SIFT | 2402 | 726 | 45 | | 17 | | | |
| | ORB | 2305 | 2566 | 115 | 55 | 56 | 64 | 71 | 104 |
| *courtyard* | SIFT | 3214 | 3830 | 939 | | 151 | | | |
| | ORB | 4416 | 4950 | 331 | 186 | 188 | 216 | 241 | 358 |
| *gate-1* | SIFT | 2673 | 465 | 35 | | 11 | | | |
| | ORB | 7806 | 7000 | 1288 | 492 | 486 | 558 | 625 | 904 |
| *gate-2* | SIFT | 2673 | 1763 | 186 | | 47 | | | |
| | ORB | 7806 | 7185 | 1575 | 519 | 515 | 587 | 657 | 942 |
| *gate-3* | SIFT | 2673 | 1788 | 216 | | 53 | | | |
| | ORB | 7806 | 7211 | 1633 | 522 | 519 | 596 | 664 | 947 |

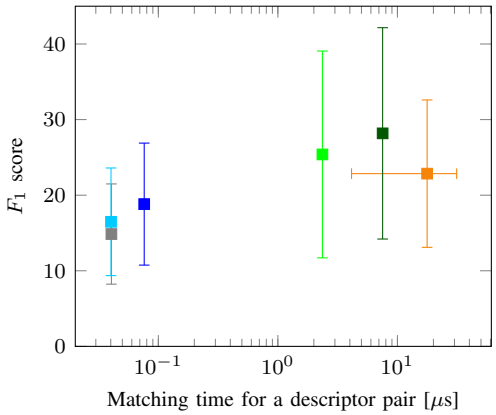



Fig. 13. Mean and standard deviation of accuracy and efficiency across all the sequence pairs. The number of initial localised ORB features set to 2000. Note the large standard deviation in the efficiency for SetDesc due to the varying length of the sets. Legend: ■ LMED, ■ T-D, ■ T-DS, ■ MST-S, ■ MST, ■ SetDesc.

number of feature trajectories in each sequence. Set representations such as SetDesc, MST-S, and MST have a higher $F_1$ score but, as expected, are slower than LMED, T-D, and T-DS, because of the *set2set min dist* strategy. The computational cost of MST-S and MST is quadratic with respect to the number of scales, $\mathcal{O}(S^2)$, whereas the computational cost of SetDesc depends on the length of the trajectories, thus resulting in a large (temporal) standard deviation. On average, the matching performance of these approaches is higher than LMED, T-D, and T-DS, but with a larger deviation. Note that the additional stability vector in T-DS and MST associated with the weighted Hamming distance (Eq. 12) doubles the matching time with respect to their counterparts, T-D and MST-S. As a reference for a histogram-based descriptor, we also report the total matching when applying SetDesc and T-D to SIFT. However, the timing between the two employed image-based descriptors are not comparable as the number of feature trajectories, as well as their length, differs from each other.

### E. Comparison of binary descriptors

The proposed spatio-temporal approaches are generic and can be applied to different image-based binary descriptors. As we model feature track extraction and spatio-temporal descriptor considering binary descriptors based on sampling patterns and dominant orientation, we analyse and compare the spatio-temporal approaches using BRIEF [14], ORB [15], LDB [32], and LATCH [18] as baselines. Note that we steer all binary reference descriptors according to the estimated orientation using the intensity centroid method [64]. We integrate the OpenCV implementation of BRIEF, ORB, and LATCH, and the author's implementation of LDB[4] in our own implementation.

While BRIEF and ORB compares intensity values of pixel pairs, LDB compares the mean intensity and the directional gradients of regular sub-windows within the patch with a multi-grid approach; and LATCH compares the norm of the difference between two sub-windows using a triplet of sampling points within the patch, with one point acting as anchor. It is noteworthy that most of the binary descriptors smooth the image (or scale level in an image pyramid) to reduce the sensitivity to noise in the intensity values [14][15][32], unless small windows are used (e.g. LATCH [18]).

We also include DeepBit [52] in the comparison, as a learnt CNN-based but non sampling-pattern based descriptor, and RFD [50] (both $\text{RFD}_\text{R}$ and $\text{RFD}_\text{G}$), as a binary descriptor based on receptive fields followed by thresholding. Note that the dimensionality of previous descriptors is 256 bits, while the dimensionality of $\text{RFD}_\text{R}$ is 293 and that of $\text{RFD}_\text{G}$ is 405[5]. Unlike previous descriptors, DeepBit cannot directly be employed within the full method, such as feature point tracking, and therefore we applied DeepBit on the patches belonging to feature tracks extracted using ORB features. We consider the 256 bit version trained on the *Liberty* (DB-L), *NotreDame* (DB-N), and *Yosemite* (DB-Y) landmarks of the UBC Phototourism dataset [55]. To also compare with a histogram-based descriptor, we provide results of SetDesc and T-D applied to SIFT [2].

Fig. 14 shows the $F_1$ score performance averaged across all sequence pairs using $F = 2000$. We can observe that RFD is a better choice for any of the spatio-temporal approaches given its higher accuracy, while DeepBit is the worst, followed by LATCH. The performance of LATCH and DeepBit shows how learning on a specific dataset (Phototourism) makes generalisation to other scenarios still a challenge. When using our multi-scale approach, MST, ORB, LATCH, and LDB become other valid alternatives to RFD. Note that selecting stable bits marginally improves the average performance of MST over MST-S.

Tab. V compares the timings[6] to extract the spatio-temporal features when employing the different image-based descriptor on three testing sequences (*desk*, *courtyard*, and *gate-1*) with varying image resolutions and content. We compare the impact

[4]http://lbmedia.ece.ucsb.edu/research/binaryDescriptor/web_home/web_home/index.html, accessed: Dec 2018

[5]http://www.nlpr.ia.ac.cn/fanbin/rfd.htm

[6]All the experiments are performed using a machine with Intel ®Core™ i7-4790S CPU @ 3.20GHz × 8, 15.6 GBi RAM, and running Ubuntu 18.04

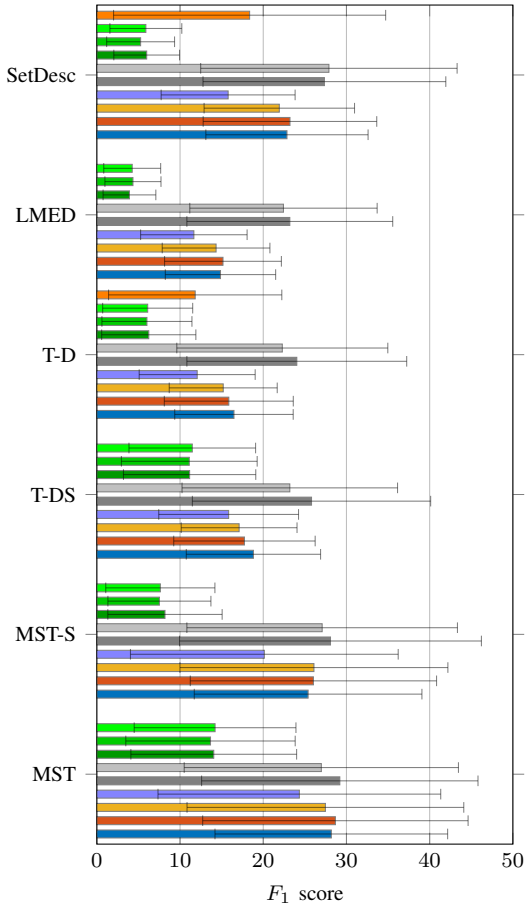



Fig. 14. Average $F_1$ score and standard deviation across all sequence pairs, targeting a maximum of 2000 local features per frame during localisation. Comparison between binary (ORB), histogram-based (SIFT), and CNN-based (DeepBit) descriptors. Note that for SIFT, we compute only the set of SIFTs over time (SetDesc) and the average within the set as reduction (T-D). Legend: SIFT, DeepBit (Yosemite), DeepBit (Notre-Dame), DeepBit (Liberty), $RFD_R$, $RFD_G$, LATCH, LDB, BRIEF, ORB.

TABLE V
EFFICIENCY ANALYSIS ON THREE TESTING SEQUENCES WITH DIFFERENT RESOLUTIONS. NOTE THAT SIFT IS NOT DESCRIBED AT MULTIPLE SCALES. LEGEND – DET.: DETECTION TIME PER FEATURE. DESC.: (MULTI-SCALE) DESCRIPTION TIME PER FEATURE. TRACK.: TRACKING TIME PER FEATURE. FRAME: AVERAGE TIME TO PROCESS A FRAME. VAL: TIMING FOR 3D GEOMETRIC TESTS. RED: TIMING FOR TEMPORAL REDUCTION. #FT: NUMBER OF FEATURE TRAJECTORIES.

| | | **Frame timings** | | | | **Post-processing timings** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **DET** ($\mu$s/feat) | **DESC** ($\mu$s/feat) | **TRACK** ($\mu$s/feat) | **FRAME** (s) | **#FT** bef. | **VAL** (s) | **RED** (s) | **#FT** after |
| *desk* | SIFT* | 11.75 | 25.98 | 50.95 | 0.04 | 3431 | 1.38 | 0.01 | 1486 |
| | ORB | 16.82 | 60.63 | 842.73 | 0.42 | 1751 | 1.08 | 1.19 | 1198 |
| | BRIEF | 10.17 | 173.92 | 988.33 | 0.21 | 514 | 0.49 | 0.50 | 423 |
| | LDB | 10.91 | 177.04 | 889.67 | 0.16 | 638 | 0.33 | 0.45 | 486 |
| | LATCH | 9.51 | 301.16 | 932.19 | 0.16 | 755 | 0.25 | 0.41 | 525 |
| | $RFD_R$ | 10.12 | 1645.10 | 2492.61 | 1.25 | 4159 | 0.95 | 1.11 | 992 |
| | $RFD_G$ | 9.90 | 14474.67 | 15066.39 | 7.46 | 5215 | 0.49 | 1.02 | 1055 |
| *courtyard* | SIFT* | 15.61 | 16.79 | 37.39 | 0.07 | 7001 | 3.61 | 0.03 | 3214 |
| | ORB | 11.73 | 30.90 | 318.05 | 0.37 | 12055 | 0.56 | 2.00 | 4416 |
| | BRIEF | 9.10 | 67.63 | 391.47 | 0.20 | 3751 | 0.31 | 1.10 | 2225 |
| | LDB | 8.74 | 111.65 | 386.26 | 0.15 | 4716 | 0.18 | 0.75 | 1887 |
| | LATCH | 9.18 | 257.21 | 494.12 | 0.17 | 5782 | 0.09 | 0.42 | 1262 |
| | $RFD_R$ | 23.36 | 1642.86 | 1946.34 | 1.86 | 15512 | 0.34 | 1.18 | 2346 |
| | $RFD_G$ | 23.82 | 14546.47 | 14463.49 | 13.93 | 17512 | 0.17 | 0.78 | 1624 |
| *gate* | SIFT* | 13.12 | 28.41 | 57.01 | 0.10 | 8373 | 11.22 | 0.02 | 2673 |
| | ORB | 13.36 | 30.54 | 726.64 | 1.09 | 19713 | 6.38 | 6.64 | 7806 |
| | BRIEF | 8.91 | 45.59 | 1248.11 | 1.86 | 12035 | 14.69 | 8.08 | 6316 |
| | LDB | 9.33 | 74.26 | 737.62 | 0.91 | 15365 | 4.89 | 5.60 | 6818 |
| | LATCH | 8.82 | 202.29 | 726.02 | 0.80 | 18403 | 2.86 | 4.54 | 6438 |
| | $RFD_R$ | 23.83 | 1623.85 | 2293.06 | 2.64 | 27017 | 3.71 | 4.02 | 4560 |
| | $RFD_G$ | 23.88 | 14429.77 | 14690.71 | 15.61 | 31392 | 1.41 | 2.84 | 3632 |

of SIFT, ORB, BRIEF, LDB, LATCH, RFD on the overall extraction of the (multi-scale) spatio-temporal features in terms of detection time per image feature, the multi-scale description time per image feature, the tracking time per image feature (including both KLT, descriptor extraction, and descriptor validation), the average time per frame, and the post-processing time consisting of the 3D geometric tests and temporal reductions. To make the comparison fair, all the binary descriptors are integrated within the same implementation, except DeepBit that extracts the descriptors from the patches of the final feature trajectories obtained with ORB. We refer the reader to the analysis of the running times provided by [52], which shows that the processing of the patches in batches makes the extraction slow and not comparable with other binary descriptors in our application. As SIFT is also integrated and adapted in the framework, we report its results as reference.

We can observe that even though RFD is the most accurate in Fig. 14, the average frame processing time is highly affected, especially due to the high extraction time of the descriptor at multiple scales[7]. Even if ORB is the fastest among the sampling-pattern based approaches in describing each feature at multiple scales, LDB and LATCH require less time, on average, to process each frame. The single scale extraction of the SIFT descriptor achieves the fastest processing of a frame, on average. Then, it is important to note how each image-based descriptor affects the number of estimated feature trajectories before the 3D geometric tests and temporal reduction, and that this number largely varies. Moreover, each feature trajectory varies in length affecting the final timing to validate in 3D, which is more noticeable in sequence 1 of *gate* that contains 100 frames instead of 50 and has a higher resolution (1280×720). Note that ORB has the highest number of feature trajectories after the 3D geometric tests across all the sequences, except *desk* where SIFT obtains a higher number. The temporal reduction is done for all validated trajectories and for LMED, T-D, T-DS, MST-S, and MST, all in once. Again, this timing is affected by both the number of feature trajectories and their varying length. SIFT is the fastest because the temporal reduction is performed only for T-D.

## VI. CONCLUSION

We presented a novel spatio-temporal multi-scale descriptor that accounts for viewpoint and scale variations across moving

[7]Slow extraction time was also observed in [13].

cameras in non-planar scenes. The proposed descriptor encodes, at multiple scales, temporal dominant and stable binary values of the neighbourhood of a 3D point from tracked binary features. The description matching function uses a cross-scale strategy to handle scale variations. Experiments showed the advantage of the proposed approach over alternative approaches for reducing the temporal descriptors. Moreover, we showed that our approach is generic and can be applied to several existing image-based binary descriptors.

Future work includes investigating descriptor-reduction methods across scales to reduce the computational time of the cross-scale matching and the application of our approach to local feature descriptors for RGB-D or depth data [71][72] for higher viewpoint-invariance.

## ACKNOWLEDGMENTS

The authors would like to thank Kevin Lin, author of DeepBit, for providing their pre-trained models, and Fabio Poiesi for providing reference camera poses.

## REFERENCES

[1] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: a benchmark and evaluation of handcrafted and learned local descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 21–26 Jul. 2017.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[3] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, 17–22 Jun. 2006.

[4] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robotics*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[5] J. L. Schönberger and J. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 27–30 Jun. 2016.

[6] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 21–26 Jul. 2017.

[7] R. Mur-Artal, J. Montiel, and J. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[8] P. Schmuck and M. Chli, "Multi-UAV collaborative monocular SLAM," in *Proc. IEEE Int. Conf. Robotics Autom.*, Singapore, 29 May–3 Jun. 2017.

[9] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[10] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 16–21 Jun. 2012.

[11] J. Dong and S. Soatto, "Domain-size pooling in local descriptors: DSP-SIFT," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 7–12 Jun. 2015.

[12] T.-Y. Yang, Y.-Y. Lin, and Y.-Y. Chuang, "Accumulated Stability Voting: A robust descriptor from descriptors of multiple scales," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 27–30 Jun. 2016.

[13] F. Bellavia and C. Colombo, "Rethinking the sGLOH descriptor," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 931 – 944, Apr. 2018.

[14] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in *Proc. Eur. Conf. Comput. Vis.*, Heraklion, Crete, Greece, 5–11 Sep. 2010.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 6–13 Nov. 2011.

[16] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 6–13 Nov. 2011.

[17] A. Xompero, O. Lanz, and A. Cavallaro, "MORB: a multi-scale binary descriptor," in *Proc. IEEE Conf. Image Process.*, Athens, Greece, 7–10 Oct. 2018.

[18] G. Levi and T. Hassner, "LATCH: learned arrangements of three patch codes," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Lake Placid, NY, USA, 7–9 Mar. 2016.

[19] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. Brit. Mach. Vis. Conf.*, London, UK, 7–10 Sep. 2009.

[20] E. Trulls, A. Sanfeliu, and F. Moreno-Noguer, "Spatiotemporal descriptor for wide-baseline stereo reconstruction of non-rigid and ambiguous scenes," in *Proc. Eur. Conf. Comput. Vis.*, Firenze, Italy, 7–13 Oct. 2012.

[21] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *IEEE Int. Workshop Vis. Surveill. Perform. Eval. Tracking and Surveill.*, Beijing, China, 15–16 Oct. 2005.

[22] I. Laptev, C. Marszalek, M. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, 23–28 Jun. 2008.

[23] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. Brit. Mach. Vis. Conf.*, Newcastle, UK, 3–6 Sep. 2008.

[24] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. Eur. Conf. Comput. Vis.*, Marseille, France, 12–18 Oct. 2008.

[25] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional SIFT descriptor and its application to action recognition," in *Proc. ACM Int. Conf. Multimedia*, Augsburg, Germany, 25–29 Sep. 2007.

[26] R. Leyva, V. Sanchez, and C. Li, "Compact and low-complexity binary feature descriptor and fisher vectors for video analytics," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 6169–6184, Dec 2019.

[27] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.

[28] T. Schmidt, R. Newcombe, and D. Fox, "Self-supervised visual descriptor learning for dense correspondence," *IEEE Robotics Autom. Lett.*, vol. 2, no. 2, pp. 420–427, Apr. 2017.

[29] A. Xompero, O. Lanz, and A. Cavallaro, "Multi-camera matching of spatio-temporal binary features," in *Int. Conf. Inf. Fusion*, Cambridge, UK, 10–13 Jul. 2018.

[30] J. yves Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker," Intel Corporation, Microprocessor Research Labs, Tech. Rep., 2000.

[31] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 16–21 Jun. 2012.

[32] X. Yang and K. T. Cheng, "Local difference binary for ultrafast and distinctive feature description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 188–194, Jan. 2014.

[33] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, 13–16 Dec. 2015.

[34] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 7–12 Jun. 2015.

[35] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks." in *Proc. Brit. Mach. Vis. Conf.*, York, UK, 19–22 Sep. 2016.

[36] R. Mitra, J. Zhang, S. Narayan, S. Ahmed, S. Chandran, and A. Jain, "Improved descriptors for patch matching and reconstruction," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Venice, Italy, 22–29 Oct. 2017.

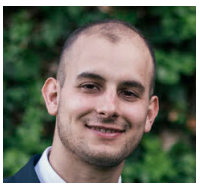
[37] H. Altwaijry, A. Veit, S. J. Belongie, and C. Tech, "Learning to detect and match keypoints with deep architectures," in *Proc. Brit. Mach. Vis. Conf.*, York, UK, 19–22 Sep. 2016.

[38] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned Invariant Feature Transform," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 8–16 Oct. 2016.

[39] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "LF-Net: Learning local features from images," in *Adv. Neural Inf. Process. Syst.*, Montréal, Canada, 3–8 Dec. 2018.

[40] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Salt Lake City, UT, USA, 18–22 Jun. 2018.

[41] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, USA, 27 Jun.–2 Jul. 2004.
[42] F. Bellavia, D. Tegolo, and C. Valenti, "Keypoint descriptor matching with context-based orientation estimation," *Image Vis. Computing*, vol. 32, no. 9, pp. 559–567, Sep. 2014.
[43] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Proc. Eur. Conf. Comput. Vis.*, Graz, Austria, 7–13 May 2006.
[44] Z. Wang, B. Fan, and F. Wu, "Local intensity order pattern for feature description," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 6–13 Nov. 2011.
[45] Z. Wang, B. Fan, G. Wang, and F. Wu, "Exploring local and overall ordinal information for robust feature description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2198–2211, Nov. 2016.
[46] T. Hassner, S. Filosof, V. Mayzels, and L. Zelnik-Manor, "SIFTing through scales," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1431–1443, Jul. 2017.
[47] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, no. 2-3, pp. 107–123, Sep. 2005.
[48] T. Trzcinski and V. Lepetit, "Efficient discriminative projections for compact binary descriptors," in *Proc. Eur. Conf. Comput. Vis.*, Firenze, Italy, 7–13 Oct. 2012.
[49] T. Trzcinski, C. M. Christoudias, and V. Lepetit, "Learning image descriptors with boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, Mar. 2015.
[50] B. Fan, Q. Kong, T. Trzcinski, Z. Wang, C. Pan, and P. Fua, "Receptive fields selection for binary feature description," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2583–2595, Jun. 2014.
[51] V. Balntas, L. Tang, and K. Mikolajczyk, "Binary Online Learned Descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, Mar. 2018.
[52] K. Lin, J. Lu, C. Chen, J. Zhou, and M. Sun, "Unsupervised deep learning of compact binary descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–14, May 2018.
[53] J. Ye, S. Zhang, T. Huang, and Y. Rui, "CDbin: Compact discriminative binary descriptor learned with efficient neural network," *IEEE Trans. Circuits Syst. Video Techn.*, pp. 1–13, Jan. 2019.
[54] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, "LDAHash: Improved matching with smaller descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 66–78, Jan. 2012.
[55] S. Winder and M. Brown, "Learning local image descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, 17–22 Jun. 2007.
[56] C. Wu, "SiftGPU: a GPU implementation of scale invariant feature transform (SIFT)," 2011, http://cs.unc.edu/~ccwu/siftgpu.
[57] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, Graz, Austria, 7–13 May 2006.
[58] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Found. and Trends in Comput. Graph. and Vis.*, vol. 3, no. 3, pp. 177–280, Jun. 2008.
[59] C. Harris and M. Stephens, "A combined corner and edge detector," in *$4^{th}$ Alvey Vis. Conf.*, Manchester, UK, 31 Aug./2 Sep. 1988.
[60] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 21–23 Jun. 1994.
[61] Z. Zhang, "A flexible new technique for camera Calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
[62] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.
[63] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Kerkyra, Greece, 20–27 Sep. 1999.
[64] P. L. Rosin, "Measuring corner properties," *Comput. Vis. Image Understanding*, vol. 73, no. 2, pp. 291–307, Feb. 1999.
[65] T. Hassner, V. Mayzels, and L. Zelnik-Manor, "On SIFTs and their scales," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 16–21 Jun. 2012.
[66] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE Int. Conf. Intell. Robot Syst.*, Vilamoura, Portugal, 7–12 Oct. 2012.
[67] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 354–366, Feb. 2013.
[68] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
[69] J. Heinly, E. Dunn, and J. Frahm, "Comparative evaluation of binary features," in *Proc. Eur. Conf. Comput. Vis.*, Firenze, Italy, 7–13 Oct. 2012.
[70] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular SLAM with multiple micro aerial vehicles," in *Proc. IEEE Int. Conf. Intell. Robot Syst.*, Tokyo, Japan, 3–7 Nov. 2013.
[71] A. Zeng, S. Song, M. Niessner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: Learning local geometric descriptors from rgb-d reconstructions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 21–26 Jul. 2017.
[72] G. Georgakis, S. Karanam, Z. Wu, J. Ernst, and J. Košecká, "End-to-end learning of keypoint detector and descriptor for pose invariant 3d matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 18–22 Jun. 2018.



**Alessio Xompero** received the MSc degree in Telecommunication Engineering from the University of Trento, Trento, Italy, in 2015. He is currently a research assistant and a Ph.D. candidate with the Centre of Intelligent Sensing at Queen Mary University of London, UK, supervised by Prof. Andrea Cavallaro. From 2017 to 2018, he was a visiting student at Fondazione Bruno Kessler (FBK), Trento, Italy, supervised by Dr. Oswald Lanz. His research interests include local image and spatio-temporal features, audio-visual object tracking, and multi-view reconstruction.

**Oswald Lanz** received the MSc degree in mathematics and a PhD in computer science from the University of Trento, Trento, Italy. He was head of the computer vision research unit at FBK and is now senior researcher at FBK and head of research at Covision Lab.

**Andrea Cavallaro** is Professor of Multimedia Signal Processing and the founding Director of the Centre for Intelligent Sensing at Queen Mary University of London, UK. He is Fellow of the International Association for Pattern Recognition (IAPR) and Turing Fellow at the Alan Turing Institute, the UK National Institute for Data Science and Artificial Intelligence. He received his Ph.D. in Electrical Engineering from the Swiss Federal Institute of Technology (EPFL), Lausanne, in 2002. He was a Research Fellow with British Telecommunications (BT) in 2004/2005 and was awarded the Royal Academy of Engineering teaching Prize in 2007; three student paper awards on target tracking and perceptually sensitive coding at IEEE ICASSP in 2005, 2007 and 2009; and the best paper award at IEEE AVSS 2009. Prof. Cavallaro is editor-in-chief of Signal Processing: Image Communication; chair of the IEEE Image, Video, and Multidimensional Signal Processing Technical Committee; IEEE Signal Processing Society Distinguished Lecturer; and an elected member of the IEEE Video Signal Processing and Communication Technical Committee. He is Senior Area Editor for the IEEE Transactions on Image Processing and Associate Editor for the IEEE Transactions on Circuits and Systems for Video Technology. He is a past Area Editor for the IEEE Signal Processing Magazine (2012-2014) and past Associate Editor for the IEEE Transactions on Image Processing (2011-2015), IEEE Transactions on Signal Processing (2009-2011), IEEE Transactions on Multimedia (2009-2010), IEEE Signal Processing Magazine (2008-2011) and IEEE Multimedia. He is a past elected member of the IEEE Multimedia Signal Processing Technical Committee and past chair of the Awards committee of the IEEE Signal Processing Society, Image, Video, and Multidimensional Signal Processing Technical Committee. Prof. Cavallaro has published over 270 journal and conference papers, one monograph on Video tracking (2011, Wiley) and three edited books: Multi-camera networks (2009, Elsevier); Analysis, retrieval and delivery of multimedia content (2012, Springer); and Intelligent multimedia surveillance (2013, Springer).