Multi-camera Matching of Spatio-Temporal Binary Features

Alessio Xompero
Centre for Intelligent Sensing
Queen Mary University of London
a.xompero@qmul.ac.uk

Oswald Lanz
Technologies for Vision
Fondazione Bruno Kessler
lanz@fbk.eu

Andrea Cavallaro
Centre for Intelligent Sensing
Queen Mary University of London
a.cavallaro@qmul.ac.uk

***Abstract*—Local image features are generally robust to different geometric and photometric transformations on planar surfaces or under narrow baseline views. However, the matching performance decreases considerably across cameras with unknown poses separated by a wide baseline. To address this problem, we accumulate temporal information within each view by tracking local binary features, which encode intensity comparisons of pixel pairs in an image patch. We then encode the spatio-temporal features into fixed-length binary descriptors by selecting temporally dominant binary values. We complement the descriptor with a binary vector that identifies intensity comparisons that are temporally unstable. Finally, we use this additional vector to ignore the corresponding binary values in the fixed-length binary descriptor when matching the features across cameras. We analyse the performance of the proposed approach and compare it with baselines.**

***Index Terms*—Spatio-temporal features, ORB, Feature matching**

I. INTRODUCTION

Local spatio-temporal features are used for object and scene recognition, human action recognition [1][2], video matching and retrieval [3], and wide baseline reconstruction [4]. Spatio-temporal feature detectors localise interest points in spatial, temporal, and scale domains [1][5]. Spatio-temporal descriptors encode appearance, motion (*e.g.* optical flow), and statistics (*e.g.* image gradients) of the spatial and temporal neighbours of the interest points [1].

In applications such as visual Simultaneous Localisation and Mapping (SLAM) [6][7][8][9], Structure from Motion [10] or stereo reconstruction [4], local features are extracted independently for each image and matched/tracked in multiple views. Online approaches such as ORB-SLAM [6] obtain spatio-temporal features by tracking local binary features (*e.g.* ORB [11]). ORB-SLAM reduces the spatio-temporal feature to a compact representation by selecting the descriptor with the least median distance from all others [6].

View matching is very challenging across freely moving cameras that observe the scene from different viewpoints. The feature similarity normally decreases with the increase of viewpoint, scale, and illumination changes. Moreover, features visible in one view may be occluded in another view, thus leading to matching ambiguities.

In this paper we investigate the problem of extracting and matching local spatio-temporal descriptors with uncalibrated and unsynchronised cameras under large viewpoint changes. We propose a spatio-temporal descriptor for feature point trajectories (*tracklets*) that captures the temporal changes of an interest point. We extract a sequence of ORB [11] descriptors and temporally pool the sequence to a compact fixed-length binary descriptor of dominant values. We also extract a second descriptor that discriminates temporally unstable binary tests and acts as a selector of the pooled descriptor for feature matching.

This paper is organised as follows. Section II reviews spatio-temporal detectors and descriptors as well as spatio-temporal features for 3D reconstruction. Section III describes the proposed spatio-temporal descriptor, its reduction, and the cross-view matching. Section IV discusses the experimental results. Finally, in Section V we draw conclusions.

II. BACKGROUND

In this section we briefly overview spatio-temporal detectors and descriptors, and we focus in particular on binary features for real-time applications.

Spatio-temporal detectors include Harris3D [5], Cuboid [12], Hessian [13], and dense sampling [1]. These detectors find space-time interest points given by local maxima of a response function, such as the Harris response [14] for Harris3D, the Gabor filters-based response for Cuboid, and the Hessian saliency measures for Hessian. Harris3D and Hessian are an extension of the space-time domain of the Harris [14] and SURF [15] detectors. All these detectors also consider the scale for both spatial and temporal domains to detect the interest points. Dense sampling does not search for local maxima of a response function and defines the location of the interest points in a regular 5-dimensional grid, which accounts for space, time, spatial scale and temporal scale, with a 50% overlap between volumes.

Spatio-temporal descriptors are 3D patches surrounding an interest point and divide the volume into smaller volumetric cells. Examples include Cuboid [12], HOG/HOF [16], HOG3D [17], Extended SURF (eSURF) [13], and 3D-SIFT [18]. Cuboid computes the gradient for each pixel followed by Principal Component Analysis to reduce the dimension of the feature vector. HOG/HOF computes normalised histograms of spatial gradient and normalised histograms of optical flow with a fixed number of bins and concatenates them to form a single

feature vector. 3D-SIFT and HOG3D extend to the spatio-temporal domain the quantisation of the histogram of gradients used in SIFT [19]. 3D-SIFT represents the gradients in polar coordinates and quantises them in histograms by meridians and parallels. This solution leads to singularity problems near the poles [17]. HOG3D overcomes this issue by using polyhedrons and projections of the gradient vectors onto the axes that connect the centre of the polyhedron to the centre of each face of the polyhedron. eSURF extends the SURF [15] descriptor by representing each cell of the 3D patch with a weighted sum of uniformly sampled responses of Haar wavelets. All these approaches use a fixed volume to extract the descriptor for a given video, thus making the matching across different viewpoints a difficult problem.

Daisy-3D [4] is a spatio-temporal description for dense 3D reconstruction with a wide baseline stereo camera in the presence of non-rigid objects and occlusions. Daisy-3D captures the temporal evolution of the spatial structure of an interest point by tracking dense 2D Daisy features [20] with optical flow priors, and concatenates the temporal descriptors. Spatio-temporal features are then matched between cameras by computing an average distance of sub-descriptors within a small window, followed by a global optimisation to enforce spatio-temporal consistency for depth estimation. The dimension of the temporal descriptors is large and therefore the Daisy-3D matching is computationally expensive. Moreover, to deal with dynamic objects in the scene, the Daisy-3D matching assumes synchronised cameras.

To obtain spatio-temporal features, most online approaches for self-localisation and 3D reconstruction rely on the extraction and tracking of local image features, such as Scale Invariant Feature Transform (SIFT) [19], Speeded Up Robust Features (SURF) [15], or Binary Robust Invariant Elementary Features (BRIEF) [21].

Binary features are preferred for real-time applications because of their extraction and matching efficiency. Examples of binary features include Oriented FAST and Rotated BRIEF (ORB) [11], Binary Robust Invariant Scale Key-point (BRISK) [22], or Fast REtinA Keypoint (FREAK) [23]. Binary features describe a small patch around an interest point with comparisons of intensity values of pixel pairs of a sampling pattern. The sampling pattern is obtained either in a deterministic way [22], in a probabilistic way [21], or through learning [11][23]. ORB features are more compact, faster to extract and can achieve a good accuracy in image feature matching benchmarks compared to the more complex SIFT features [24][6]. For this reason, ORB features are used in several pipelines, such as ORB-SLAM [6] and the Multi-UAV Collaborative SLAM [9]. Nevertheless, their performance decreases under severe geometric changes, such as scale and viewpoint, which typically occur when multiple cameras move freely.

III. Spatio-temporal descriptor and matching

A. Localisation and descriptor extraction

Let $\mathbf{I}_k$ be a (gray-scale) frame at time k captured by an uncalibrated and moving camera with unknown poses. We apply the FAST corner detector [25] in each $\mathbf{I}_k$ and retain the F features with the highest Harris response [14], which are at feature locations $\{\mathbf{x}_{1,k}, \ldots, \mathbf{x}_{f,k}, \ldots, \mathbf{x}_{F,k}\}$.

After smoothing $\mathbf{I}_k$ with a 2D Gaussian filter of size $W = 7$ and standard deviation $\sigma = 2$, we extract a descriptor $\mathbf{d}_\mathbf{p}$ for each feature location using the ORB [11] sampling pattern on a $G \times G$ patch $\mathbf{p} = \rho(\mathbf{I}_k, \mathbf{x}_{f,k}, G)$ centred at each feature location $\mathbf{x}_{f,k}$:

$$\mathbf{d}_\mathbf{p} = [\tau_\mathbf{p}(\mathbf{u}_1, \mathbf{v}_1), \ldots, \tau_\mathbf{p}(\mathbf{u}_q, \mathbf{v}_q), \ldots, \tau_\mathbf{p}(\mathbf{u}_{256}, \mathbf{v}_{256})], \quad (1)$$

where $\mathbf{u}_q$ and $\mathbf{v}_q$ are the positions of each pixel pair defined by the sampling pattern $\mathbf{S}$, with $q = 1, ..., 256$. The sampling pattern $\mathbf{S}$ consists of learnt pixel pairs with high variance and low correlation in their binary derivative [11].

The function $\tau_\mathbf{p}(\cdot, \cdot)$ is a binary test on the intensity values $\mathbf{p}(\mathbf{u}_q)$ and $\mathbf{p}(\mathbf{v}_q)$ in patch $\mathbf{p}$ of each pixel pair $\mathbf{u}_q$ and $\mathbf{v}_q$ of the sampling pattern:

$$\tau_\mathbf{p}(\mathbf{u}_q, \mathbf{v}_q) = \begin{cases} 1 & \text{if} \quad \mathbf{p}(\mathbf{u}_q) < \mathbf{p}(\mathbf{v}_q), \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

To account for in-plane rotations, we compute the orientation angle $\theta_\mathbf{p}$ of the patch with respect to its centre of mass as defined by the intensity centroid method [26]. After applying the rotation $\mathbf{R}(\theta_\mathbf{p}) \in SO(2)$ to the sampling pattern $\mathbf{S}$: $\mathbf{S}_\mathbf{p} = \mathbf{R}(\theta_\mathbf{p})\mathbf{S}$, the descriptor is $(\mathbf{x}, \theta, \mathbf{d})_{f,k}$, which encodes the location, $\mathbf{x}$, orientation, θ, and ORB descriptor $\mathbf{d}$ of the local image feature at frame k.

B. Descriptor tracking and reduction

We track the features between frame $\mathbf{I}_k$ and $\mathbf{I}_{k-1}$ by matching their descriptors with a nearest neighbour approach followed by a validation strategy to allow only one-to-one matches. For each feature from frame k we select the three closest features in frame $k-1$ by using as dissimilarity measure the Hamming distance: $\mathbf{d}_{f,k} \oplus \mathbf{d}_{g,k-1}$, where $\oplus$ is the bit-wise XOR operator. After ranking all candidate matches according to their Hamming distance, we discard matches whose feature in $\mathbf{I}_k$ are outside a gate of radius $r = 10$ pixels (as in the KLT tracker [27]) of the feature in $\mathbf{I}_{k-1}$. We also discard matches with a feature with higher similarity in another match.

The resulting spatio-temporal feature is $\mathcal{T}_i = \{(\mathbf{x}, \theta, \mathbf{d})_{i,k}\}_{k=k_{i_1}}^{k_{i_l}}$, where k_{i_1} and k_{i_l} are the first and last frames where the feature is detected (see Fig. 1). The sequence of image locations $\{\mathbf{x}_{k_{i_1}}, \ldots, \mathbf{x}_{k_{i_l}}\}$ denotes the trajectory (or *tracklet*) of the spatio-temporal feature, with length $L_i = k_{i_l} - k_{i_1}$. The spatio-temporal descriptor, $\mathbf{d}_i \in \{0,1\}^{L_i \times 256}$ is the temporal concatenation of the ORB descriptors: $\mathbf{d}_i = [\mathbf{d}_{i,k_{i_1}}, \ldots, \mathbf{d}_{i,k_{i_l}}]$.

We reduce $\mathbf{d}_i$ to a fixed-length descriptor $\mathbf{z}_i \in \{0,1\}^{256}$ with $\mathbf{z}_i = [z_{1,i}, \ldots, z_{q,i}, \ldots, z_{256,i}]$ by accumulating the binary test values over time (pooling) and applying a threshold to determine the final binary test value (voting),

$$z_{q,i} = \begin{cases} 1 & \text{if} \quad \frac{1}{L_i}\langle \mathbf{d}_{q,i}, \mathbf{1} \rangle > 0.5, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

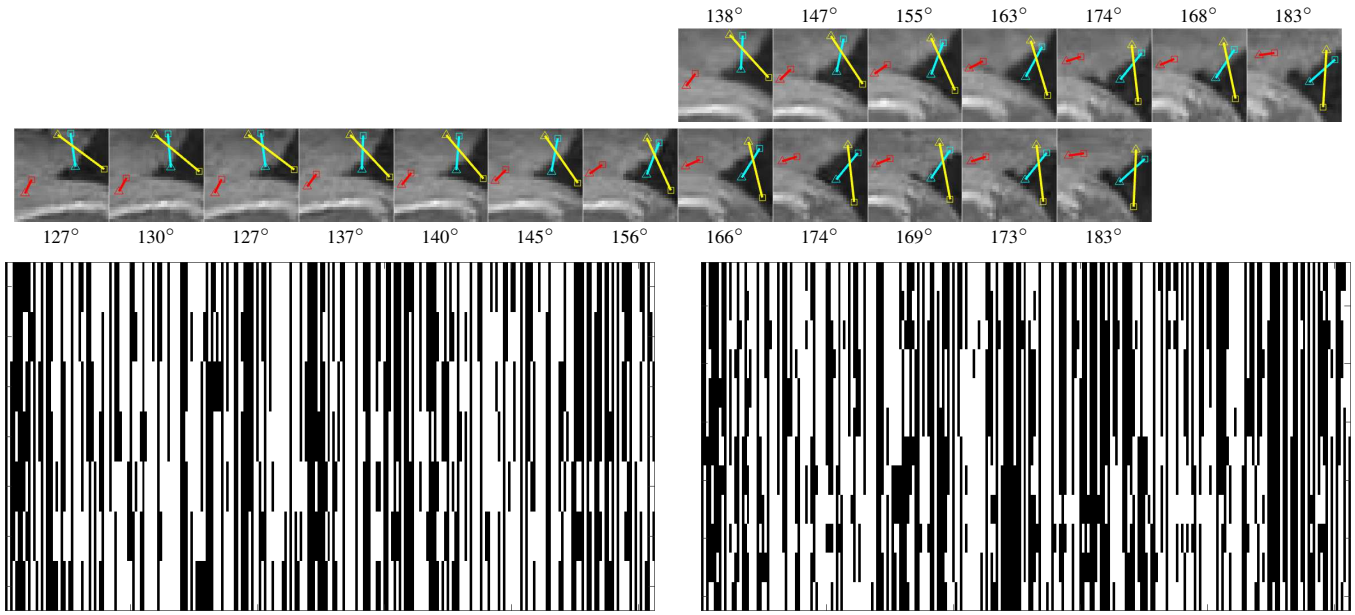

Fig. 1. On top, sample patch orientation changes from frame 7 to frame 20 (from left to right) for the tracked ORB descriptor in one camera (first row) and the corresponding tracked ORB descriptor in an another camera (second row). For each patch we show its orientation in degrees and 3 sample rods (red, cyan, yellow) from the ORB sampling pattern. At the bottom, the corresponding temporal ORB descriptors (differently from the patches, time is in a top-down representation), where we can see that some binary tests remain mostly stable on the vertical signals (black is a 0 and white is a 1).

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

To account for noise in the temporal matching caused *e.g.* by photometric changes or image blur, we allow some variations in the binary test outcome, at a rate lower than 20% of the length of the spatio-temporal feature. We therefore compute a second descriptor, $\mathbf{d}'_i \in \{0,1\}^{(L_i-1)\times 256}$, that captures the temporal changes of the binary tests in $\mathbf{d}_i$ and contains the bit-wise XOR of two consecutive ORB descriptors. We reduce $\mathbf{d}'_i$ to $\mathbf{m}_i \in \{0,1\}^{256}$ with $\mathbf{m}_i = [m_{1,i}, \ldots, m_{q,i}, \ldots, m_{256,i}]$ that contains the stability information of $\mathbf{z}_i$:

$$m_{q,i} = \begin{cases} 1 & \text{if} \quad \frac{1}{L_i-1}\langle \mathbf{d}'_{q,i}, \mathbf{1} \rangle \leq 0.2, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

C. Cross-view matching

Let i be the index of a spatio-temporal feature in one view ($\mathbf{z}_i$) and j the index of a spatio-temporal feature in another view ($\mathbf{z}_j$). To improve the feature matching across views, we remove temporally unstable binary tests of $\mathbf{z}_i$ and $\mathbf{z}_j$ by applying in turn the additional descriptors $\mathbf{m}_i$ and $\mathbf{m}_j$ to the XOR operation between $\mathbf{z}_i$ and $\mathbf{z}_j$ through the weighted Hamming distance [28].

Let $M_i = \langle \mathbf{m}_i, \mathbf{1} \rangle$ be the number of stable binary tests for $\mathbf{z}_i$ and M_j for $\mathbf{z}_j$. Let $\langle \mathbf{m}_i, \mathbf{z}_i \oplus \mathbf{z}_j \rangle$ be the masked Hamming distance using only $\mathbf{m}_i$. We compute the final dissimilarity measure between two descriptors as a weighted linear combination of two masked Hamming distances:

$$h(i,j) = \frac{M_i \langle \mathbf{m}_i, \mathbf{z}_i \oplus \mathbf{z}_j \rangle + M_j \langle \mathbf{m}_j, \mathbf{z}_i \oplus \mathbf{z}_j \rangle}{M_i + M_j}. \tag{5}$$

The set of putative matches is therefore determined by a similarity matching strategy such as threshold-based or nearest neighbour [29]. The ratio test between the distance of the first and second nearest neighbours can also be computed to remove possible ambiguities [19].

IV. Experiments

A. Experimental setup

We compare ST-ORB, P-ST-ORB, Mask-P-ST-ORB, and LMED. ST-ORB corresponds to the high-dimensional, temporally concatenated ORB descriptor, $\mathbf{d}_i$. P-ST-ORB corresponds to the reduced binary descriptor $\mathbf{z}_i$, while Mask-P-ST-ORB complements P-ST-ORB with $\mathbf{m}_i$. LMED is proposed within ORB-SLAM [6] and selects the single ORB descriptor within ST-ORB with the least median Hamming distance with respect to all the other single ORB descriptors. Even if LMED was proposed for tracking ORB features with a single camera, we analyse here its performance for cross-view matching.

To extract ORB descriptors [11] we use their OpenCV 3.3 implementation with default parameters: the FAST threshold is 20, the number of features is $F = 500$, and the patch size is $G = 31$. Moreover, we set the number of scales to 1.

We use the most suitable dissimilarity measure for each descriptor when matching features. For ST-ORB, we compute the Hamming distance of each pair of single ORB descriptors between $\mathbf{d}_i$ and $\mathbf{d}_j$. Then we use the minimum among all Hamming distances as dissimilarity measure between the two ST-ORB descriptors. For P-ST-ORB and LMED, we use the Hamming distance. For Mask-P-ST-ORB, we use the weighted Hamming distance (see Eq. (5)).

B. Dataset

We use images from three datasets: *coslam_courtyard*, *freiburg_office* and *freiburg_desk*. Fig. 2 shows five frames for each camera and for each dataset.

The data of *coslam_courtyard*[1] [30] are four videos recorded with a hand-held camera in a university courtyard. Starting from a similar position in front of a panel, each video was acquired by moving the camera around the area with different paths and returning to the initial position at the end of the recording. From the first and fourth sequences, we select the first 50 frames after sub-sampling the videos from 50 to 5 fps. As there is no camera calibration data provided with the dataset, we evaluate the methods only qualitatively.

For quantitative evaluation with ground-truth data, we use the TUM-RGB-D SLAM dataset [31] that contains monocular sequences acquired indoors with a Kinect. The Kinect was either handheld or mounted on a robot. Ground-truth camera poses were acquired with a motion capture system. From the dataset, we select two sequences with enough texture for detecting and tracking features, and with loop closures or different movements of the camera around the same scene: *freiburg_office* and *freiburg_desk*. For each sequence, we then select two portions of 50 frames with non-overlapping frames to simulate the motion of two cameras looking at the same portion of the scene from different viewpoints. For *freiburg_office*, we select the frames from 114 to 163 and from 2305 to 2354. The scenario consists of two cameras moving slowly around a cluttered desk and without strong viewpoint changes. For *freiburg_desk*, we select the frames from 97 to 147 and from 390 to 340. The scenario consists of two cameras moving in opposite directions around an office desk with more severe changes in scale and viewpoint. Note that some images in the datasets are affected by blur.

C. Performance evaluation

Inspired by [29], we evaluate the spatio-temporal feature matching by exploiting the depth images and ground-truth poses provided with the TUM-RGB-D SLAM dataset.

Given two or more sequences acquired with an RGB-D camera, we relate each RGB pixel to its corresponding depth pixel. Using projective geometry [32], we reconstruct the 3D structure of the scene in a common reference system, as the ground-truth poses are provided by a motion capture system. We can then determine spatio-temporal features for each video stream as well as ground-truth correspondences[2].

For each spatio-temporal feature $\mathcal{T}_i$, we compute a 3D location $\mathbf{X}_i$ as the median of the set of 3D points estimated from the back-projection of the image locations $\left\{\mathbf{x}_{i,k_{i_1}}, \ldots, \mathbf{x}_{i,k_{i_l}}\right\}$ and properly scale them using the associated values in the depth images. The median helps to remove false 3D estimations caused by noise or errors in the tracking of the spatio-temporal features. After obtaining a set of reconstructed 3D points for each video stream, we apply a brute force approach between the two sets and we then define the ground-truth correspondences as the set of all 3D point pairs whose Euclidean distance is lower than 3 cm.

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

[2] Given different sampling rates for RGB and depth, we consider the same depth image for two RGB images that are temporarily the closest to the depth image.

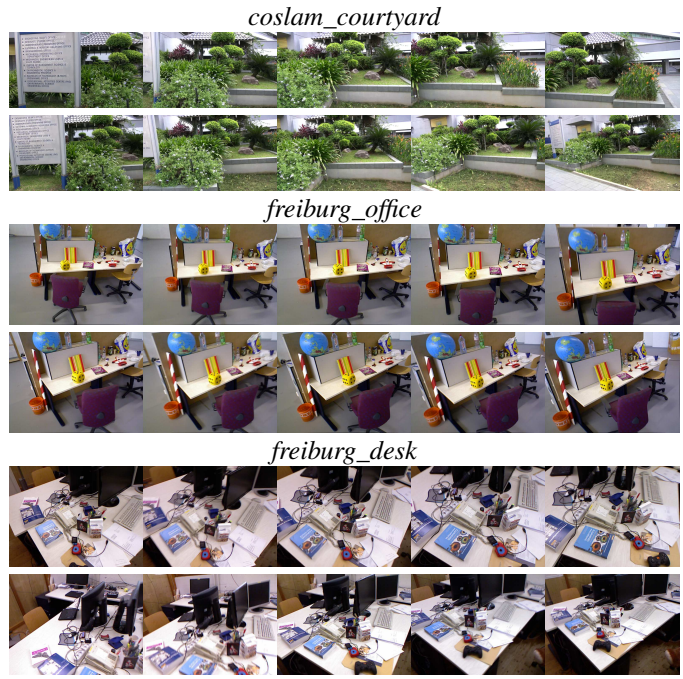

Fig. 2. Frames 0, 10, 20, 30, and 40 (from left to right) of the two camera sequences for the *coslam_courtyard* (top), *freiburg_office* (middle) and *freiburg_desk* (bottom) datasets.

Given the set of matches with a sufficiently high similarity (putative matches), we define a correct match as the tracklet pair that is also a ground-truth correspondence. Using the ground-truth correspondences, putative matches and correct matches, we compute precision, recall, F-score, and matching score (as in [29]). Precision is the ratio between the number of correct matches and the total number of matches. Recall is the ratio between the number of correct matches and the number of ground-truth correspondences. F-score$= 2\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$ is the harmonic mean between precision and recall. The matching score is the ratio between the number of correct matches and the minimum between the number of tracklets in one view and the other view.

D. Results on the outdoor dataset

For the *coslam_courtyard* dataset, we consider the nearest neighbour with ratio test as similarity matching strategy. For each spatio-temporal descriptor in the second camera, we search for the two nearest descriptors in the first camera, and we select the match only if the distance ratio of two nearest neighbours is below a threshold (we use the value 0.8 as in [24]). As there are fewer than 100 matches, we manually annotate true and false positives and we report the results in Tab. I. The number of spatio-temporal features estimated is 200 for the first camera and 343 for the second camera. The

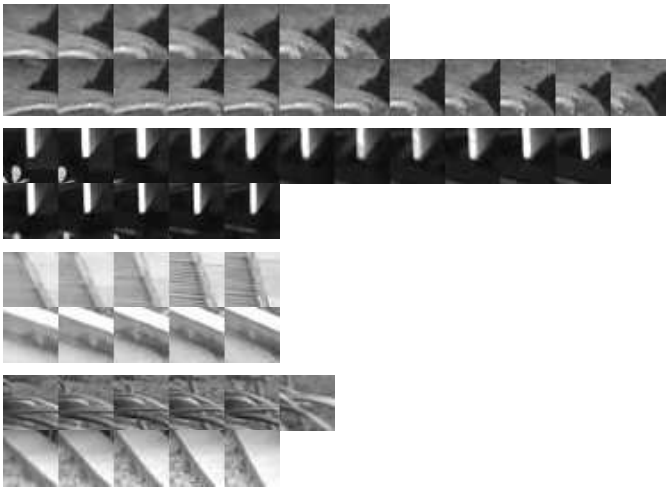

Fig. 3. Example of true matches and false matches with the Mask-P-ST-ORB and with Hamming distance lower than 50. Notice the different length of the temporal patches. First match (true positive): first tracklet from frame 15 to 21 and second tracklet from frame 8 to frame 19. Second match (true positive): first tracklet from frame 37 to 47 and second tracklet from frame 41 to frame 45. Third match (false positive): first tracklet from frame 43 to 47 and second tracklet from frame 46 to frame 50, Hamming distance equal to 38. Fourth match (false positive): first tracklet from frame 10 to 15 and second tracklet from frame 22 to frame 26, Hamming distance equal to 46.

number of matches estimated by each method is similar to each other, but LMED finds much fewer true positives than the other approaches. We can see that Mask-P-ST-ORB can achieve and slightly outperform the performance of the exhaustive ST-ORB. Fig. 3 shows few examples of true and false matches with a Hamming distance lower than 50 using Mask-P-ST-ORB. Even if true positives are well matched, we can still see the limitations of the spatio-temporal descriptor that hardily discriminates the pavement from the leaves only based on the intensities (fourth example at the bottom of Fig. 3).

E. Results with synthetic tracklets

For the *freiburg_office* and *freiburg_desk* datasets, we analyse a set of 'synthetic' tracklets for each video stream to reduce possible errors in the extraction of the spatio-temporal features and focus the evaluation mainly on the cross-view matching. The 'synthetic' tracklets are generated as follows.

For each camera and for each frame, we detect FAST [25] corner points and we back-project each point $\mathbf{x}_{f,k}$ to its corresponding 3D point $\mathbf{X}_{f,k} \in \mathbb{R}^3$ using the associated depth value $s_{f,k}$, the camera pose $\mathbf{C}_k \in SE(3)$ for frame k, and the camera calibration matrix $\mathbf{K}$,

$$\mathbf{X}_{f,k} = \pi^{-1}(\mathbf{x}_{f,k}, \mathbf{C}_k, \mathbf{K}, s_{f,k}), \tag{6}$$

where $\pi(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the projective transformation of a pinhole camera model [32]. The camera calibration matrix $\mathbf{K}$ contains the intrinsic parameters, such as the focal length and the principal point.

As the back-projection of the feature locations is independent for each frame, there will be duplicated 3D points. Starting from the first 3D point, we therefore remove all those successive 3D points whose distance is lower than 2 cm from the candidate point until we obtain a set of unique 3D points.

TABLE I
MATCHING RESULTS ON THE *coslam_courtyard* DATASET USING THE NEAREST NEIGHBOUR WITH RATIO TEST MATCHING STRATEGY. GROUND-TRUTH IS NOT AVAILABLE AND WE MANUALLY ANNOTATE TRUE AND FALSE POSITIVES. TP: TRUE POSITIVES.

Method	# Matches	# TP	Precision
ST-ORB	67	37	.55
LMED	51	17	.33
P-ST-ORB	64	36	.56
Mask-P-ST-ORB	56	32	**.57**

To generate the tracklets, we then project each 3D point in each frame of each camera. If the image point is within the image bounds, we validate which of the four neighbour pixels after approximation of the image point coordinates is closer to the 3D point by back-projecting each pixel again in 3D:

$$\|\pi^{-1}\left(\pi(\mathbf{X}_f, \mathbf{C}_k, \mathbf{K}), s_{f,k}\right) - \mathbf{X}_f\| < 0.02. \tag{7}$$

If none of the four pixels passes the validation test, the visibility of the 3D point in frame k is set to 0. This means that either that 3D point is occluded or the estimated depth value was inaccurate. Moreover, we allow only image points that have a positive Harris response [14] to avoid flat areas or points along an edge. As last step, we accept valid tracklets only if the number of visible image points is greater than four in at least either of the two video streams. This procedure attempts to make the synthetic tracklets as close as possible to the real tracklets, removing possible tracking errors and/or splitting single tracklets in multiple instances.

We evaluate the methods with the generated 'synthetic' tracklets on the *freiburg_desk*. Fig. 4 shows the distributions of the Hamming distance for true and false positives for each method[3]. The methods under analysis do not easily discriminate true and false positives as viewpoint and scale changes are especially challenging in this scenario. Moreover, there are several repetitive patterns (*e.g.* on the keyboard) that create ambiguities in finding correct correspondences. Nevertheless, there are also true positives whose appearance are not so similar, making the match hard to estimate.

Tab. II and Fig. 5 show the results of the methods by varying the threshold on the Hamming distance from 0 to 128 and also by varying the number of matches from 0 to 5000. We can see that ST-ORB achieves the best performance because of the exhaustive matching strategy that can find the single ORB descriptors with the most similar appearance. Mask-P-ST-ORB outperforms P-ST-ORB and LMED in terms of recall, however it has a high number of false positives, showing that the proposed descriptor is not discriminative enough. Nevertheless, also ST-ORB and P-ST-ORB become less accurate when increasing the threshold of the Hamming distance.

[3]A similar behaviour was presented in BRIEF [21] on a planar scene with increasing viewpoint change.

TABLE II

NUMBER OF TRUE POSITIVES (TP), NUMBER OF FALSE POSITIVES (FP), PRECISION (P), RECALL (R), F-SCORE (F) AND MATCHING SCORE (MS) FOR *freiburg_desk* AT DIFFERENT HAMMING DISTANCE (HD) THRESHOLDS. THE NUMBER OF TRACKLETS IS 2516 IN CAMERA a AND 3308 IN CAMERA b. THE NUMBER OF GROUND-TRUTH CORRESPONDENCES IS **2447**. TRACKLETS ARE GENERATED FROM 3D POINTS USING THE DEPTH IMAGES. WHEN THE NUMBER OF FALSE POSITIVES IS TOO HIGH, WE HIGHLIGHT THE VALUE IN RED.

HD	Method	# TP	# FP	P	R	F	MS
20	ST-ORB	128	564	.18	**.05**	**.08**	**.05**
	LMED	3	9	**.25**	.00	.00	.00
	P-ST-ORB	3	15	.17	.00	.00	.00
	Mask-P-ST-ORB	44	8071	.01	.02	.01	.02
30	ST-ORB	515	4107	.11	**.21**	**.15**	**.20**
	LMED	19	87	**.18**	.01	.01	.01
	P-ST-ORB	19	151	.11	.01	.01	.01
	Mask-P-ST-ORB	147	33669	.00	.06	.01	.06
40	ST-ORB	1034	29684	.03	**.42**	.06	**.41**
	LMED	51	540	**.09**	.02	.03	.02
	P-ST-ORB	66	1134	.06	.03	.04	.03
	Mask-P-ST-ORB	314	102177	.00	.13	.01	.12
50	ST-ORB	1484	176029	.01	**.61**	.02	**.59**
	LMED	120	3644	**.03**	.05	**.04**	.05
	P-ST-ORB	158	7102	.02	.06	.03	.06
	Mask-P-ST-ORB	630	267332	.00	.26	.00	.25
60	ST-ORB	1841	717154	.00	**.75**	.01	**.73**
	LMED	212	18728	**.01**	.09	**.02**	.08
	P-ST-ORB	309	32584	**.01**	.13	**.02**	.12
	Mask-P-ST-ORB	974	616485	.00	.40	.00	.39

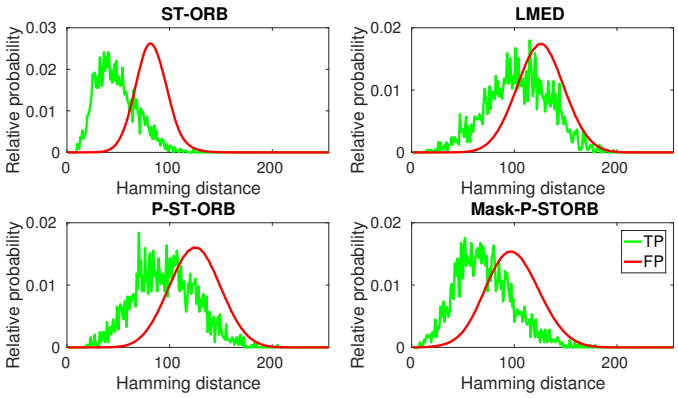

Fig. 4. Distribution of the Hamming distances for corresponding features (true positives, TP) and non-corresponding features (false positives, FP) for ST-ORB, LMED, P-ST-ORB, and Mask-P-ST-ORB with 'synthethic' tracklets.

Because each method can obtain the best performance at different thresholds of the Hamming distances and to make the comparison fair, we compare the methods by varying the number of matches from 0 to 5000. We can observe that Mask-P-ST-ORB has very low performance in the first 5000 matches, while the other methods confirm the previous results.

F. Results with real tracklets

We compare all the methods on both *freiburg_office* and *freiburg_desk* datasets using the spatio-temporal features obtained with the tracking approach discussed in Sec. IV-F. Fig. 6, top shows the distributions of the Hamming distance for corresponding (true positives) and non-corresponding (false positives) features. As *freiburg_office* has a limited change

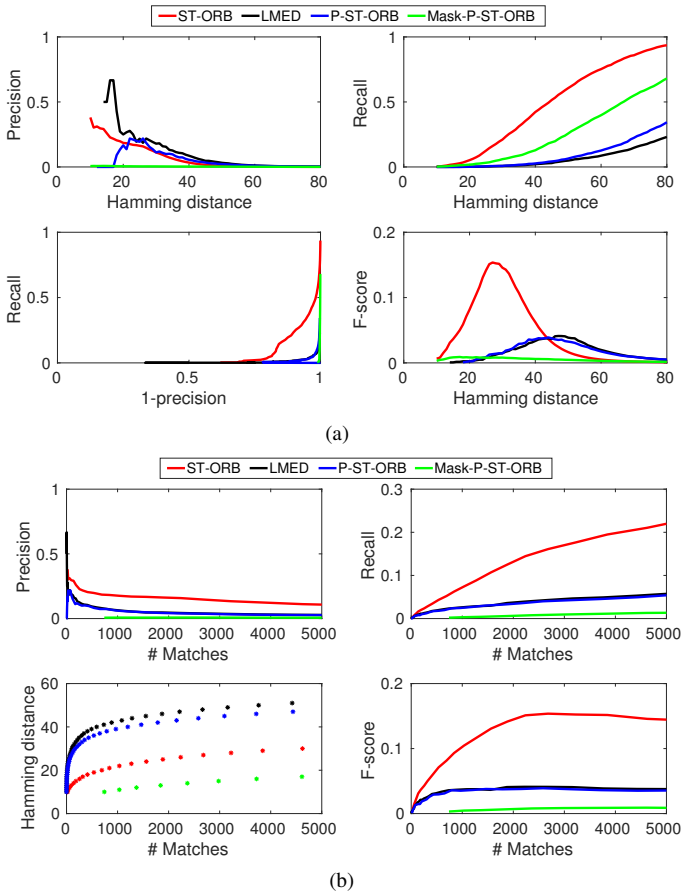

Fig. 5. Performance results with 'synthetic' tracklets on the *freiburg_desk* dataset. (a) Precision, recall, recall vs 1-precision, and F-score curves with the threshold on the Hamming distance varying from 20 to 80. (b) Precision, recall and F-score curves with the number of matches varying from 0 to 5000. The relation between Hamming distance and the number matches is also shown.

in viewpoint and the camera motion is quite slow, the distributions are slightly better separated, while they cannot be distinguished in the *freiburg_desk* dataset. Fig. 6 also shows the performance results by varying the Hamming distance (Fig. 6, middle) and by varying the number of matches (Fig. 6, bottom). ST-ORB is the best in both datasets, while P-ST-ORB and Mask-P-ST-ORB outperform LMED. However, the performance of Mask-P-ST-ORB is similar to P-ST-ORB, showing that masking the temporally unstable binary tests is unnecessary in this case. Therefore, the first reduction is sufficient to compact the high-dimensional ST-ORB descriptor.

V. CONCLUSION

We investigated the problem of matching spatio-temporal features extracted from videos acquired by independently moving cameras. We proposed a spatio-temporal binary descriptor obtained by tracking ORB [11] features and concatenating their descriptors. As matching the high-dimensional descriptors is computationally expensive, we accumulated the spatio-temporal features into a fixed-length binary descriptors by pooling and selecting the temporally dominant values. We

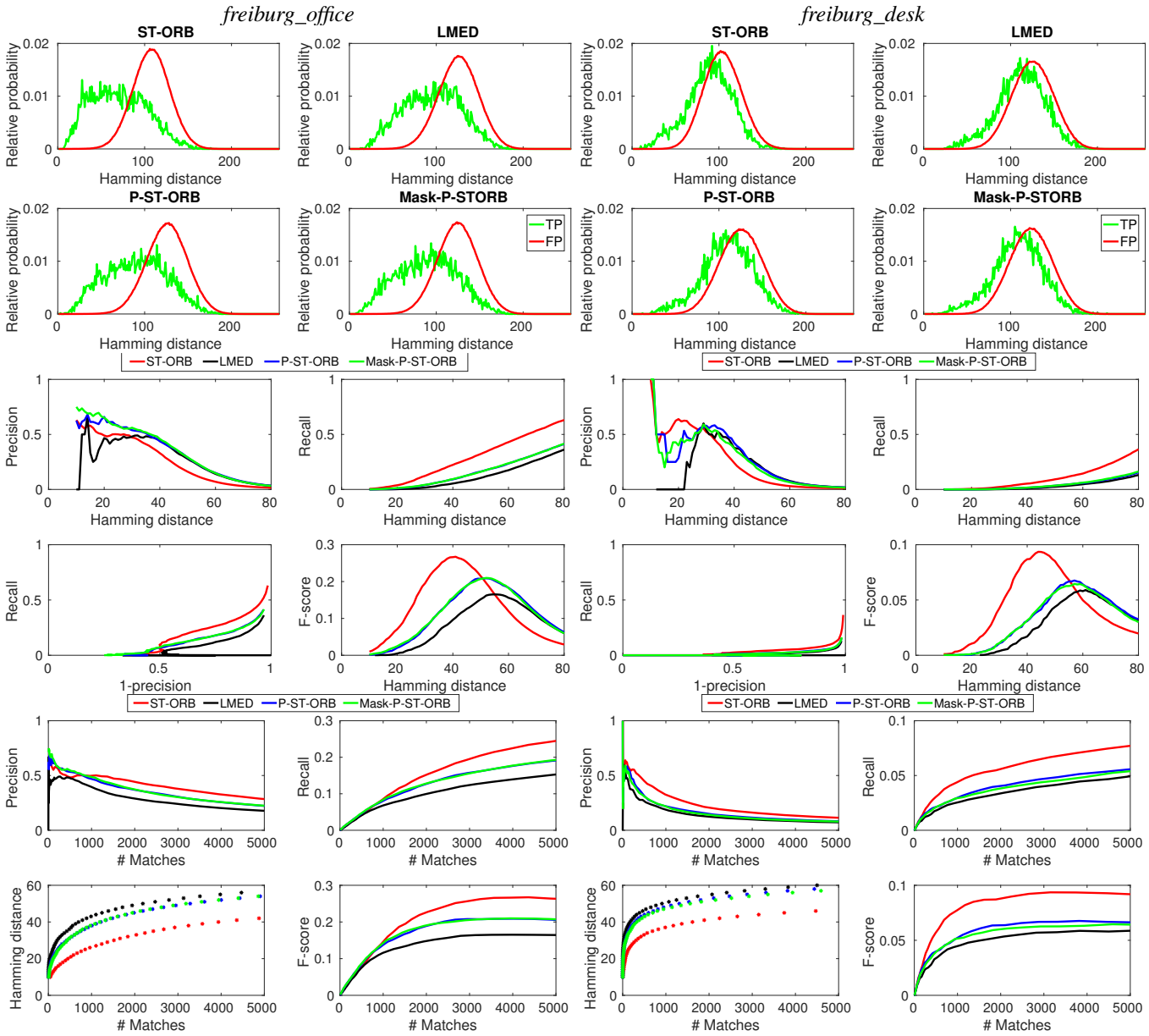

Fig. 6. Performance results with real tracklets on the *freiburg_office* (left columns) and *freiburg_desk* (right columns) datasets. On top (first two rows), distribution of the Hamming distances for corresponding features (true positives, TP) and non-corresponding features (false positives, FP) for ST-ORB, LMED, P-ST-ORB, and Mask-P-ST-ORB. In the middle (third and fourth rows), precision, recall, recall vs 1-precision, and F-score curves with the threshold on the Hamming distance varying from 20 to 80. At the bottom (fifth and sixth rows), precision, recall and F-score curves with the number of matches varying from 0 to 5000. The relation between Hamming distance and the number matches is also shown.

also complemented this descriptor with an additional binary descriptor by encoding the temporal stability of each binary test and ignoring those binary values in the first descriptor when matching features across cameras. Experiments showed that our descriptor outperforms LMED, the method proposed in ORB-SLAM [6]. As future work, we will investigate an effective reduction approach that considers the viewpoint and preserves the matching efficiency.

References

[1] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. of The British Machine Vision Conference*, London, UK, 7–10 Sept. 2009.

[2] Z. Gao, W. Nie, A. Liu, and H. Zhang, "Evaluation of local spatial–temporal features for cross-view action recognition," *Neurocomputing*, vol. 173, no. P1, pp. 110–117, Jan. 2016.

[3] O. Alatas, O. Javed, and M. Shah, "Compressed Spatio-temporal Descriptors for Video Matching and Retrieval," in *Proc. of the IEEE Conference on Pattern Recognition*, Cambridge, UK, 26 Aug. 2004.

[4] E. Trulls, A. Sanfeliu, and F. Moreno-Noguer, "Spatiotemporal Descriptor for Wide-Baseline Stereo Reconstruction of Non-rigid and Ambiguous Scenes," in *Proc. of the European Conference on Computer Vision*, Firenze, Italy, 7–13 Oct. 2012.

[5] I. Laptev, "On Space-Time Interest Points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, Sept. 2005.

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

[7] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles," in *Proc. of The IEEE International Conference on Intelligent Robot Systems*, Tokyo, Japan, 3–7 Nov. 2013.

[8] L. Riazuelo, J. Civera, and J. M. M. Montiel, "C^2TAM: A Cloud framework for Cooperative Tracking and Mapping," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 401 – 413, Apr. 2014.

[9] P. Schmuck and M. Chli, "Multi-UAV Collaborative Monocular SLAM," in *Proc. of The IEEE International Conference on Robotics and Automation*, Singapore, Singapore, 29 May/3 June 2017.

[10] J. L. Schönberger and J. Frahm, "Structure-from-motion revisited," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proc. of the IEEE International Conference on Computer Vision*, Barcelona, Spain, 6–13 Nov. 2011.

[12] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *The IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, China, 15–16 Oct. 2005.

[13] G. Willems, T. Tuytelaars, and L. Van Gool, "An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector," in *Proc. of the European Conference on Computer Vision*, Marseille, France, 12–18 Oct. 2008.

[14] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. of the fourth Alvey Vision Conference*, Manchester, UK, 31 Aug./ 2 Sept. 1988.

[15] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Proc. of the European Conference on Computer Vision*, Graz, Austria, 7–13 May 2006.

[16] I. Laptev, C. Marszalek, M. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 23–28 June 2008.

[17] A. Klaser, M. Marszałek, and C. Schmid, "A Spatio-Temporal Descriptor Based on 3D-Gradients," in *Proc. of The British Machine Vision Conference*, Leeds, UK, 1–4 Sept. 2008.

[18] P. Scovanner, S. Ali, and M. Shah, "A 3-Dimensional SIFT Descriptor and its Application to Action Recognition," in *Proc. of The ACM International Conference on Multimedia*, Augsburg, Germany, 25–29 Sept. 2007.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[20] E. Tola, V. Lepetit, and P. Fua, "DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, May 2010.

[21] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in *Proc. of the European Conference on Computer Vision*, Heraklion, Crete, Greece, 5–11 Sept. 2010.

[22] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," in *Proc. of the IEEE International Conference on Computer Vision*, Barcelona, Spain, 6–13 Nov. 2011.

[23] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast Retina Keypoint," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 16–21 June 2012.

[24] J. Heinly, E. Dunn, and J. Frahm, "Comparative evaluation of binary features," in *Proc. of the European Conference on Computer Vision*, Firenze, Italy, 7–13 Oct. 2012.

[25] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," in *Proc. of the European Conference on Computer Vision*, Graz, Austria, 7–13 May 2006.

[26] P. L. Rosin, "Measuring corner properties," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 291–307, 1999.

[27] J. Shi and C. Tomasi, "Good features to track," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 21–23 June 1994.

[28] V. Balntas, L. Tang, and K. Mikolajczyk, "Binary Online Learned Descriptors," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, Mar. 2017.

[29] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[30] D. Zou and P. Tan, "CoSLAM: Collaborative Visual SLAM in Dynamic Environments," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 354–366, Feb. 2013.

[31] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *Proc. of The IEEE International Conference on Intelligent Robot Systems*, Vilamoura-Algarve, Portugal, 7–12 Oct. 2012.

[32] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.