2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Learning Binary Features Online from Motion Dynamics for Incremental Loop-Closure Detection and Place Recognition

Guangcong Zhang[1], Mason J. Lilly[2], and Patricio A. Vela[1]

Abstract— **This paper proposes a simple yet effective approach to learn visual features online for improving loop-closure detection and place recognition, based on bag-of-words frameworks. The approach learns a codeword in the bag-of-words model from a pair of matched features from two consecutive frames, such that the codeword has temporally-derived perspective invariance to camera motion. The learning algorithm is efficient: the binary descriptor is generated from the mean image patch, and the mask is learned based on discriminative projection by minimizing the intra-class distances among the learned feature and the two original features. A codeword is generated by packaging the learned descriptor and mask, with a masked Hamming distance defined to measure the distance between two codewords. The geometric properties of the learned codewords are then mathematically justified. In addition, hypothesis constraints are imposed through temporal consistency in matched codewords, which improves precision. The approach, integrated in an incremental bag-of-words system, is validated on multiple benchmark data sets and compared to state-of-the-art methods. Experiments demonstrate improved precision/recall outperforming state of the art with little loss in runtime.**

I. Introduction

Long-distance visual Simultaneous Localization and Mapping experiences drift that require sophisticated approaches to handle [24], [25]. When the robot revisits a place, the drift can be greatly reduced by imposing geometric constraints in the posterior optimization (e.g. bundle adjustment) [16]. Identifying when a robot has returned to a previously visited place is referred as loop-closure detection. It plays a key role in visual SLAM systems.

Appearance-based methods have become prevalent in visual loop closure detection, due to their independence from the robot's estimated location [8]. Key research in appearance-based methods examines what kinds of visual features best describe the scene. Various features have been used in loop-closure detection, from floating point arithmetic based features such as SIFT [15] and SURF [3], to binary-encoded features [7], [14], [18]. In the typical bag-of-words loop-closure system, a feature is extracted from a single frame after being matched with the previous frame, then potentially used as a codeword in the vocabulary. Such codewords may not be invariant to the perspective transformation due to robot motion.

[1]Guangcong Zhang and Patricio A. Vela are with School of Electrical & Computer Engineering, and Institute of Robotics and Intelligent Machines, Georgia Institute of Technology, North Ave NW, Atlanta, GA 30332, USA. Mason Lilly is with Transylvania University, Lexington, KY 40508, USA. Mason Lilly's work was done when on a summer internship at Georgia Tech. {zhanggc, pvela}@gatech.edu, jmlilly16@transy.edu.

To compensate for perspective effects, visual loop-closure systems often require a similar trajectory profile to trigger a loop-closure. The loop-closing image sequence views the revisited scene from a similar perspective. The key idea in this paper is to learn the codewords by learning feature descriptors invariant to the perspective transformations induced by robot motion. With such codewords, visual features of the same object subject to perspective distortions are more likely to trigger loop-closure hypotheses and improve recall. We use binary features due to their overall advantages demonstrated in loop-closure applications, efficient computation with high precision-recall (PR) [11], [12].

Learning binary features is done by treating the image patches from a matched pair together with the mean patch as a single class, then optimizing the binary test by minimizing the intra-class distance and maximizing the inter-class distance through Linear Discriminants Analysis (LDA) [2], [5], [6]. Furthermore, because a codeword is learned from two consecutive images, if a frame is retrieved as a loop-closure hypothesis, its previous or next frame should also be a hypothesis. Based on this, our algorithm imposes temporal constraints in the hypothesis selection, which further improves the precision. In addition, mathematical analysis shows that the codewords learned by our method have nice geometric properties, which theoretically supports the proposed approach.

The major contributions of our paper include:

- an efficient algorithm based on LDA for learning codewords invariant to perspective transformations from robot motion, involving only matrix additions on image patches and bit-wise operations on binary vectors;
- theoretical justification for the geometric properties in the learned codeword, demonstrating that the learned codewords can be interpreted to be "centroids" in the space induced by the modified Hamming distance;
- integration into the incremental bag-of-word loop-closure detection system with additional simple hypothesis constraints that demonstrate improved PR with trivial runtime loss on various benchmark data sets.

II. Related Work

Research effort has sought to improve the pipeline of appearance-based methods, involving: (1) the visual features chosen as codewords; (2) loop-closure retrieval models, e.g. probabilistic model in Fab-Map [8], bag of visual words model [1], [11]; (3) data structures for vocabulary storage and search, e.g. Chow-Liu tree [8], vocabulary tree for binary features [11]; and (4) online incremental design in order to

978-1-4673-8026-3/16/$31.00 ©2016 IEEE

get rid of offline training, e.g. IBuILD system [12]. This review focuses on the visual features used in the loop closure detection, which is highly driven by the development of feature descriptors. Early work in visual descriptors based on floating-point arithmetic, such as SIFT [15] and SURF [3], are typically too expensive in computation to fulfill real-time loop-closure detection. The milestone work Fab-map [8] mitigates this problem by using quantized SURF descriptors, which are encoded in binary vectors. Such an approximation trades precision-recall with runtime. In [1], the authors use raw SIFT descriptors with an additional feature space of local color histogram. With a tree-structure vocabulary in their bag-of-word model, high frame rates are attained with the feature retrieval of logarithmic-time complexity in codeword number. The work in [13] attains real-time performance by using compact randomized tree signatures. With the developments of binary descriptors such as BRIEF [7], BRISK [14], ORB [18], binary features have been widely adopted for loop-closure detection (often with bag-of-word models) due to their fast computation and comparable precision/recall to SIFT and SURF. The new standard, a binary features based bag-of-word system, is presented in [11]. The IBuILD system [12] further improves the binary bag-of-word recipe by designing an online incremental system without the need for prior feature training. Our system improves upon IBuILD.

Beside standard feature descriptors, related research has sought to improve descriptors through additional learning processes. These learning approaches include LDA [6], [5], [22], [2], boosting [21], Principal Component Analysis (PCA) [23], Domain-Size Pooling [10], etc. Our work is inspired by [2], in which LDA is applied by learning a mask **locally** to minimize the intra-class distance of binary descriptors. The algorithm synthesizes samples from each image by rotating the patch, and treats the original patch as a pivot patch along with the auxillary samples to form a single class for LDA. To contrast, our method first uses the synthesized patch as the pivot patch and the original patches as auxillary patches. More importantly, instead of learning the invariance to some heuristic transformations, our method learns the transformation invariance for the actual robot motion.

III. Learning Binary Codeword Invariant to Frame-by-Frame Motion Dynamics

This section first presents basics about binary feature descriptors and the LDA method for optimizing the descriptor by learning a mask. Then it details the proposed algorithm for learning features from motion dynamics. The geometric properties will be discussed in detail. Our notation uses boldface for a vector or matrix (e.g., $\mathbf{x}$), and normal font for a scalar binary or real value (e.g., x_i).

A. Binary Descriptors and Intra-class Distance

1) Binary Descriptors for a visual feature:
Binary descriptors [7], [14], [18] follow the same basic formulation. Given an image intensity patch I, a binary descriptor is encoded as a binary vector $\mathbf{x}$, composed of L bits $x_i \in \mathbb{B}$ (typically $L = 512$). Often after a smoothing operation, each bit in $\mathbf{x}$ is generated by binary tests $\{[\mathbf{a_i}, \mathbf{b_i}]\}_{i=1}^{L}$

$$x_i = \begin{cases} 1 & \text{if } I(\mathbf{a_i}) < I(\mathbf{b_i}) \\ 0 & \text{otherwise} \end{cases}, \quad \forall i = 1 \dots L \tag{1}$$

where each $\mathbf{a_i} = \left[u_{a_i}, v_{a_i}\right]^\top$ (and similarly $\mathbf{b_i}$) is a pixel position. The binary test patterns are usually generated offline by training on large data sets. Here, we use the BOLD binary test pattern [2].

2) Intra-class distance:
The distance between two binary descriptors is measured by the Hamming distance d_{H} with bit-xor operation $\oplus$:

$$\begin{aligned} d_{\mathrm{H}}\left(\mathbf{x}^{(k)}, \mathbf{x}^{(k')}\right) &= \mathbf{x}^{(k)} \oplus \mathbf{x}^{(k')} \\ &= \sum_{i=1}^{L} x_l^{(k)} \oplus x_l^{(k')} = \sum_{i=1}^{L} \left(x_l^{(k)} - x_l^{(k')}\right)^2 \end{aligned} \tag{2}$$

For a set of image patches $\{I^{(k)}\}$ from the same class and the corresponding binary descriptors $\{\mathbf{x}^{(k)}\}$, the expected intra-class distance is:

$$\begin{aligned} &\mathbb{E}\left[d_{\mathrm{H}}(\{\mathbf{x}^{(k)}\})\right] = \frac{1}{L}\sum_{l=1}^{L} \mathbb{E}\left[d_{\mathrm{H}}(\{x_l^{(k)}\})\right] \\ &= \frac{1}{LK^2}\sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{k'=1}^{K} d_{\mathrm{H}}(x_l^{(k)}, x_l^{(k')}) \\ &= \frac{1}{LK^2}\sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{k'=1}^{K} \left(x_l^{(k)} - x_l^{(k')}\right)^2 \\ &= \frac{1}{LK^2}\sum_{l=1}^{L}\left(2\sum_{k=1}^{K}\sum_{k=1}^{K}\left(x_l^{(k)}\right)^2 - 2\sum_{k=1}^{K}\sum_{k'=1}^{K} x_l^{(k)} x_l^{(k')}\right) \\ &= \frac{1}{L}\sum_{l=1}^{L} 2\mathbb{E}\left[x_l^2\right] - 2\mathbb{E}\left[x_l\right]^2 \end{aligned} \tag{3}$$

B. Learning Codewords from Motion Dynamics

1) Minimizing intra-class distances with binary masks:
Eq. 3 shows that minimizing the intra-class distance can be done by masking the binary coordinates with high variance, effectively projecting out the highly variable coordinates. We will therefore package learned codewords into a feature ensemble consisting of a feature descriptor and a binary mask $\mathbf{D} = \{\mathbf{x}, \mathbf{y}\} \in \mathcal{D}_{\mathrm{MH}}$, where the mask $\mathbf{y}$ is defined

$$y_i \cong \begin{cases} 1 & \text{if } (\wedge_k x_i^{(k)} = 1) \vee (\vee_k x_i^{(k)} = 0) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

for each coordinate $i \in \{1 \dots L\}$. The distance between two feature ensembles, with non-zero masks, is defined by the "masked Hamming distance" d_{MH}:

$$\begin{aligned} d_{\mathrm{MH}}(\mathbf{D}_1, \mathbf{D}_2) &= \frac{|\mathbf{y}_2||\mathbf{x}_1 \oplus \mathbf{x}_2 \cap \mathbf{y}_1| + |\mathbf{y}_1||\mathbf{x}_1 \oplus \mathbf{x}_2 \cap \mathbf{y}_2|}{|\mathbf{y}_1| + |\mathbf{y}_2|} \\ &= \frac{|\mathbf{y}_2|}{|\mathbf{y}_1| + |\mathbf{y}_2|}\mathfrak{d}_2^1 + \frac{|\mathbf{y}_1|}{|\mathbf{y}_1| + |\mathbf{y}_2|}\mathfrak{d}_1^2 \end{aligned} \tag{5}$$

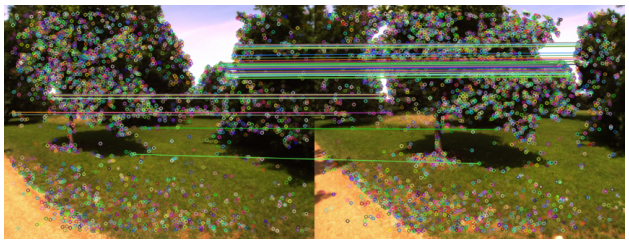

Fig. 1. For each pair of matched frame, we extract a 48×48 patch I_1 from the previous frame, and I_2 from the current frame. These two will be used to learn a codeword.

Algorithm 1: Learning codewords from motion dynamics.

Data: $I_1, I_2 \in \mathbb{R}^{M \times N}$ – imaged patches from a pair of matched features in two consecutive frames; $\{[\mathbf{a_i}, \mathbf{b_i}]\}$ – lists of binary tests positions

Result: $\mathbf{D}_m = \{\mathbf{x}_m, \mathbf{y}_m\}$ – Codeword of length L invariant to perspective transformation between I_1, I_2

$I_m \leftarrow \frac{1}{2}(I_1 + I_2)$; // $\mathcal{O}(MN)$

$\mathbf{x}_m \leftarrow$ **BinaryTests** $(I_m, \{[\mathbf{a_i}, \mathbf{b_i}]\})$; // Eq.1, $\mathcal{O}(L)$

$\mathbf{y}_m \leftarrow$ **MaskLearning** $(\{I_m, I_1, I_2\})$; // Eq.7, $\mathcal{O}(L)$

where $|\cdot|$ is the number of 1s in a binary vector; $\mathfrak{d}_j^i \triangleq |\mathbf{x}_i \oplus \mathbf{x}_j \cap \mathbf{y}_i|$ and $\mathfrak{d}_j^i \neq \mathfrak{d}_i^j$.

The distance metric defined in Eq. 5 has two significant properties: (1) $d_{\text{MH}}(\mathbf{D}_1, \mathbf{D}_2) \in [0, L]$. The lower bound is straightforward, and the upper bound is simply given by $d_{\text{MH}}(\mathbf{D}_1, \mathbf{D}_2) \leq \frac{2|\mathbf{y}_1||\mathbf{y}_2|}{|\mathbf{y}_1|+|\mathbf{y}_2|} \leq \frac{2|\mathbf{y}_1||\mathbf{y}_2|}{2\sqrt{|\mathbf{y}_1||\mathbf{y}_2|}} = \sqrt{|\mathbf{y}_1||\mathbf{y}_2|} \leq L$. (2) When $\mathbf{y}_1, \mathbf{y}_2$ are all 1s, then $d_{\text{MH}}(\mathbf{D}_1, \mathbf{D}_2) \equiv d_{\text{H}}(\mathbf{x}_1, \mathbf{x}_2)$, which means the masked Hamming distance defined also accommodates the Hamming distance. The masked Hamming distance is not a metric since the coincidence axiom and the triangular inequality do not hold.

2) Learning codewords invariant to cross-frame motion: Given a pair of matched image patches I_1, I_2 from two consecutive frames (as depicted in Fig. 1), the mean patch I_m is used to generate the codeword $\mathbf{D}_m = \{\mathbf{x}_m, \mathbf{y}_m\}$,

$$I_m = \frac{1}{2}(I_1 + I_2). \tag{6}$$

I_m has the structural information of I_1, I_2. Binary tests on I_1, I_2, and I_m, per Eq. 1, generate the binary vectors $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_m$. The masks $\mathbf{y}_1$ and $\mathbf{y}_2$ are set to all 1s. The mask $\mathbf{y}_m$ is computed by minimizing the intra-class distance among $\{I_m, I_1, I_2\}$:

$$y_{m,i} = \begin{cases} 1 & \text{if } \bigcap_{k\in\{1,2,m\}} \neg(I_k(\mathbf{a_i}) < I_k(\mathbf{b_i})) = 1 \\ & \text{or } \bigcap_{k\in\{1,2,m\}} (I_k(\mathbf{a_i}) < I_k(\mathbf{b_i})) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The algorithm defining the mean codeword is summarized in Algorithm 1. Fig. 2 depicts two binary tests with $\{I_m, I_1, I_2\}$. For non-zero variance in the same test across three patches, the corresponding coordinate in $\mathbf{x}_m$ will be masked out through $\mathbf{y}_m$, as illustrated in Fig. 3.

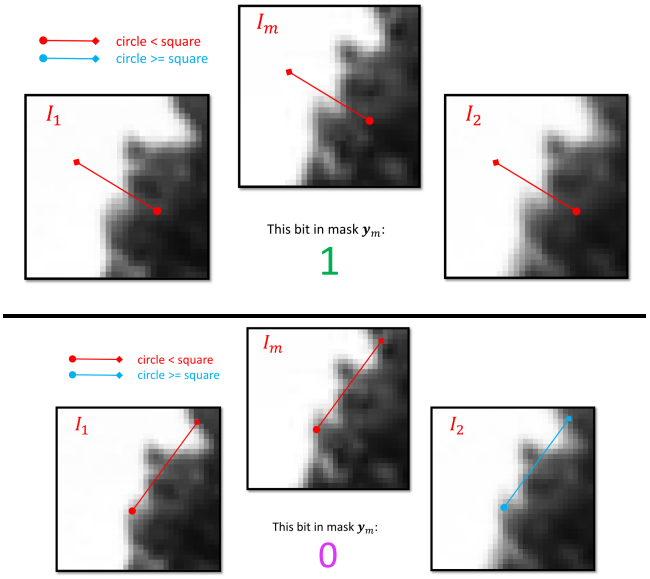

Fig. 2. Illustration of binary tests and mask learning for I_m from I_1, I_2.

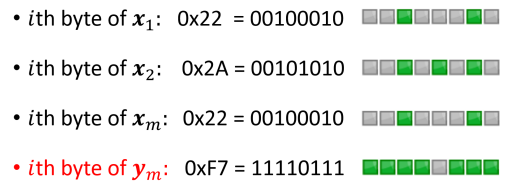

Fig. 3. Codeword learning from the perspective of bit-wise operations.

C. Geometric Properties of Learned Codewords

Relative to the source codewords $\mathbf{D}_1 = \{\mathbf{x}_1, \mathbf{y}_1\}$ and $\mathbf{D}_2 = \{\mathbf{x}_2, \mathbf{y}_2\}$ both in $\mathcal{D}_{\text{MH}}$, the learned mean codewords have the following two nice geometric properties in the space $\mathcal{D}_{\text{MH}}$ induced by d_{MH} (note that $\forall\ \mathbf{D}_k \in \mathcal{D}_{\text{MH}},\ |\mathbf{y}_k| \neq 0$):

1) $\mathbf{D}_m$ can be viewed as the **topological centroid** of $\mathbf{D}_1$ and $\mathbf{D}_2$, as depicted in Fig. 4;
2) Given any other point $\mathbf{D}_k \in \mathcal{D}_{\text{MH}}$, $\mathbf{D}_m$ **preserves the localities** between $\mathbf{D}_k$ and $\mathbf{D}_m$, and between $\mathbf{D}_k$ and $\mathbf{D}_1, \mathbf{D}_2$.

These properties are shown to hold in Theorems 1 and 2.

***Theorem** 1:* Let $\mathbf{D}_m$ be the codeword generated from $\mathbf{D}_1$ and $\mathbf{D}_2$, then

$$\begin{aligned} &d_{\text{MH}}(\mathbf{D}_m, \mathbf{D}_1) \leq d_{\text{MH}}(\mathbf{D}_1, \mathbf{D}_2) \quad \text{and} \\ &d_{\text{MH}}(\mathbf{D}_m, \mathbf{D}_2) \leq d_{\text{MH}}(\mathbf{D}_1, \mathbf{D}_2) \end{aligned} \tag{8}$$

Proof: From Eq. 7, $\mathfrak{d}_1^m = 0$. The masked Hamming distance reduces to:

$$\begin{aligned} d_{\text{MH}}(\mathbf{D}_m, \mathbf{D}_1) &= \frac{|\mathbf{y}_m|}{|\mathbf{y}_1| + |\mathbf{y}_m|}\mathfrak{d}_m^1 + \frac{|\mathbf{y}_1|}{|\mathbf{y}_1| + |\mathbf{y}_m|}\mathfrak{d}_1^m \\ &= \frac{|\mathbf{y}_m|}{|\mathbf{y}_1| + |\mathbf{y}_m|}\mathfrak{d}_m^1 \end{aligned} \tag{9}$$

Also, because I_m is the mean patch of I_1 and I_2, $|\mathbf{x}_m \oplus \mathbf{x}_1| \leq |\mathbf{x}_2 \oplus \mathbf{x}_1|$, which further implies $\mathfrak{d}_m^1 = |(\mathbf{x}_m \oplus \mathbf{x}_1) \cap \mathbf{y}_1| \leq |(\mathbf{x}_2 \oplus \mathbf{x}_1) \cap \mathbf{y}_1| = \mathfrak{d}_2^1$. In addition,

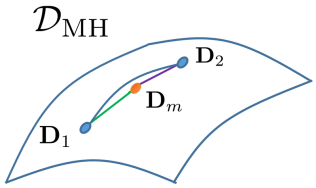

Fig. 4. Topologically $\mathbf{D}_m$ is centroid of $\mathbf{D}_1$ and $\mathbf{D}_2$ in $\mathcal{D}_{\mathrm{MH}}$.

$\mathbf{y}_1 = \mathbf{y}_2 \Longrightarrow \mathfrak{d}_1^2 = \mathfrak{d}_2^1$. Therefore,

$$\begin{aligned} d_{\mathrm{MH}}(\mathbf{D}_m, \mathbf{D}_1) &\leq \frac{|\mathbf{y}_m|}{|\mathbf{y}_1| + |\mathbf{y}_m|}\mathfrak{d}_2^1 \leq \mathfrak{d}_2^1 \\ &\leq \frac{|\mathbf{y}_2|}{|\mathbf{y}_1| + |\mathbf{y}_2|}\mathfrak{d}_2^1 + \frac{|\mathbf{y}_1|}{|\mathbf{y}_1| + |\mathbf{y}_2|}\mathfrak{d}_2^1 \\ &\leq \frac{|\mathbf{y}_2|}{|\mathbf{y}_1| + |\mathbf{y}_2|}\mathfrak{d}_2^1 + \frac{|\mathbf{y}_1|}{|\mathbf{y}_1| + |\mathbf{y}_2|}\mathfrak{d}_1^2 \\ &\leq d_{\mathrm{MH}}(\mathbf{D}_1, \mathbf{D}_2) \end{aligned} \tag{10}$$

Likewise, $d_{\mathrm{MH}}(\mathbf{D}_m, \mathbf{D}_2) \leq d_{\mathrm{MH}}(\mathbf{D}_1, \mathbf{D}_2)$ also holds. ■

***Theorem** 2:* Let $\mathbf{D}_m$ be the codeword generated from D_1 and D_2, then $\forall\ \mathbf{D}_k \in \mathcal{D}_{\mathrm{MH}}$, the following inequality holds

$$d_{\mathrm{MH}}(\mathbf{D}_k, \mathbf{D}_1) + d_{\mathrm{MH}}(\mathbf{D}_k, \mathbf{D}_2) \geq \lambda d_{\mathrm{MH}}(\mathbf{D}_k, \mathbf{D}_m), \tag{11}$$

where

$$\lambda = \left(\frac{|\mathbf{y}_k|}{L + |\mathbf{y}_k|}\right)\left[1 + \frac{\min(|\mathbf{y}_m|, |\mathbf{y}_k|)}{\max(|\mathbf{y}_m|, |\mathbf{y}_k|)}\right] \in [0, 1] \tag{12}$$

Proof: First consider codewords with only one bit. There are two cases:

Case 1: If $\mathbf{x}_1$ and $\mathbf{x}_2$ have the same value, WLOG assume this value is 1, then $\mathbf{x}_m = \mathbf{y}_m = 1$. Since $\mathbf{x}_k, \mathbf{y}_k \in \{0, 1\}$, there are four situations as listed in Table I.

Case 2: If $\mathbf{x}_1$ and $\mathbf{x}_2$ have the different value, WLOG assume $\mathbf{x}_1 = 1, \mathbf{x}_2 = 0$, then $\mathbf{y}_m = 0$ and $\mathbf{x}_m$ can be 1 or 0. Due to the symmetry, we only need to consider either one of these situations. Let $\mathbf{x}_m = 0$. There are again four situations listed in Table II.

Therefore, in any one-dimensional case it holds that

$$\begin{aligned} d_{\mathrm{MH}}(m, k) &\leq d_{\mathrm{MH}}(1, k) + d_{\mathrm{MH}}(2, k) \\ &\Longleftrightarrow \mathfrak{d}_k^m + \mathfrak{d}_m^k \leq \mathfrak{d}_k^1 + \mathfrak{d}_1^k + \mathfrak{d}_k^2 + \mathfrak{d}_2^k \end{aligned} \tag{13}$$

For any descriptor with L dimensions, $\mathfrak{d}_k^m + \mathfrak{d}_m^k \leq \mathfrak{d}_k^1 + \mathfrak{d}_1^k + \mathfrak{d}_k^2 + \mathfrak{d}_2^k$ still holds, because $\mathfrak{d}_j^i$ is a summation over all dimensions without weighting. For the sum of the two masked Hamming distances in the theorem statement,

$$\begin{aligned} &d_{\mathrm{MH}}(\mathbf{D}_1, \mathbf{D}_k) + d_{\mathrm{MH}}(\mathbf{D}_2, \mathbf{D}_k) \\ &= \frac{|\mathbf{y}_k|\mathfrak{d}_k^1}{|\mathbf{y}_1| + |\mathbf{y}_k|} + \frac{|\mathbf{y}_1|\mathfrak{d}_1^k}{|\mathbf{y}_1| + |\mathbf{y}_k|} + \frac{|\mathbf{y}_k|\mathfrak{d}_k^2}{|\mathbf{y}_2| + |\mathbf{y}_k|} + \frac{|\mathbf{y}_2|\mathfrak{d}_2^k}{|\mathbf{y}_2| + |\mathbf{y}_k|} \\ &= \frac{|\mathbf{y}_k|\mathfrak{d}_k^1}{L + |\mathbf{y}_k|} + \frac{L\mathfrak{d}_1^k}{L + |\mathbf{y}_k|} + \frac{|\mathbf{y}_k|\mathfrak{d}_k^2}{L + |\mathbf{y}_k|} + \frac{L\mathfrak{d}_2^k}{L + |\mathbf{y}_k|}, \end{aligned}$$

TABLE I

DISTANCES OF CODEWORDS WITH ONE BIT (CASE#1). FOR SIMPLICITY WE USE $d_{\mathrm{MH}}(i, j)$ TO DENOTE $d_{\mathrm{MH}}(\mathbf{D}_i, \mathbf{D}_j)$.

$\mathbf{D}_1$	$\mathbf{D}_2$	$\mathbf{D}_m$	$\mathbf{D}_k$	$d_{\mathrm{MH}}(1,k)$	$d_{\mathrm{MH}}(2,k)$	$d_{\mathrm{MH}}(m,k)$
(1,1)	(1,1)	(1,1)	(1,1)	0	0	0
(1,1)	(1,1)	(1,1)	(1,0)	0	0	0
(1,1)	(1,1)	(1,1)	(0,1)	1	1	1
(1,1)	(1,1)	(1,1)	(0,0)	1	1	1

TABLE II

DISTANCES OF CODEWORDS WITH ONE BIT (CASE#2).

$\mathbf{D}_1$	$\mathbf{D}_2$	$\mathbf{D}_m$	$\mathbf{D}_k$	$d_{\mathrm{MH}}(1,k)$	$d_{\mathrm{MH}}(2,k)$	$d_{\mathrm{MH}}(m,k)$
(1,1)	(0,1)	(0,0)	(1,1)	0	1	1
(1,1)	(0,1)	(0,0)	(1,0)	0	1	0
(1,1)	(0,1)	(0,0)	(0,1)	1	0	0
(1,1)	(0,1)	(0,0)	(0,0)	1	0	0

the property $|\mathbf{y}_k| \leq |\mathbf{y}_1| \equiv |\mathbf{y}_2| = L$, implies

$$\begin{aligned} &d_{\mathrm{MH}}(\mathbf{D}_1, \mathbf{D}_k) + d_{\mathrm{MH}}(\mathbf{D}_2, \mathbf{D}_k) \\ &\geq \frac{|\mathbf{y}_k|}{L + |\mathbf{y}_k|}\left(\mathfrak{d}_k^1 + \mathfrak{d}_1^k + \mathfrak{d}_k^2 + \mathfrak{d}_2^k\right) \\ &\geq \frac{|\mathbf{y}_k|}{L + |\mathbf{y}_k|}\left(\mathfrak{d}_k^m + \mathfrak{d}_m^k\right) \\ &\geq \frac{|\mathbf{y}_k|}{L + |\mathbf{y}_k|} \cdot \frac{|\mathbf{y}_k| + |\mathbf{y}_m|}{\max(|\mathbf{y}_k|, |\mathbf{y}_m|)} \\ &\qquad \cdot \left(\frac{\max(|\mathbf{y}_k|, |\mathbf{y}_m|)}{|\mathbf{y}_k| + |\mathbf{y}_m|}\mathfrak{d}_k^m + \frac{\max(|\mathbf{y}_k|, |\mathbf{y}_m|)}{|\mathbf{y}_k| + |\mathbf{y}_m|}\mathfrak{d}_m^k\right) \\ &\geq \frac{|\mathbf{y}_k|}{L + |\mathbf{y}_k|} \cdot \frac{|\mathbf{y}_k| + |\mathbf{y}_m|}{\max(|\mathbf{y}_k|, |\mathbf{y}_m|)} \\ &\qquad \cdot \left(\frac{|\mathbf{y}_m|}{|\mathbf{y}_k| + |\mathbf{y}_m|}\mathfrak{d}_k^m + \frac{|\mathbf{y}_k|}{|\mathbf{y}_k| + |\mathbf{y}_m|}\mathfrak{d}_m^k\right) \\ &\geq \left(\frac{|\mathbf{y}_k|}{L + |\mathbf{y}_k|}\right)\left[1 + \frac{\min(|\mathbf{y}_m|, |\mathbf{y}_k|)}{\max(|\mathbf{y}_m|, |\mathbf{y}_k|)}\right] d_{\mathrm{MH}}(\mathbf{D}_k, \mathbf{D}_m) \\ &\geq \lambda \cdot d_{\mathrm{MH}}(\mathbf{D}_k, \mathbf{D}_m) \end{aligned}$$

It is easy to see that $\lambda > 0$ ($\lambda \neq 0$ because $|\mathbf{y}_k| \neq 0$). On the other hand, $\lambda = \left(\frac{1}{L/|\mathbf{y}_k| + 1}\right)\left[1 + \frac{\min(|\mathbf{y}_m|, |\mathbf{y}_k|)}{\max(|\mathbf{y}_m|, |\mathbf{y}_k|)}\right] \leq \left(\frac{1}{L/L + 1}\right)\left[1 + \frac{\max(|\mathbf{y}_m|, |\mathbf{y}_k|)}{\max(|\mathbf{y}_m|, |\mathbf{y}_k|)}\right] = 1$. Thus, $\lambda \in [0, 1]$. ■

***Remark** 1:* Theorem 1 leads to the interpretation of $\mathbf{D}_m$ as a topological centroid.

***Remark** 2:* The limits $\lambda \in [0, 1]$ in Theorem 2 are conservative since they do not factor that $\mathbf{D}_1, \mathbf{D}_2$ are matched features. In reality, $\mathbf{x}_1 \approx \mathbf{x}_2 \Longrightarrow |\mathbf{y}_m| \approx L$. In practice, $\mathbf{D}_k$ will also be generated from a matched pair. Therefore, $|\mathbf{y}_k| \approx L \approx |\mathbf{y}_m|$ also, to conclude $\lambda \approx \frac{1}{2}(1 + 1) = 1$.

***Remark** 3:* Intuitively, Theorem 2 shows that the locality is preserved by using $\mathbf{D}_m$ as a proxy. The theorem can be interpreted as "*if $\mathbf{D}_k$ is far away from $\mathbf{D}_m$, then $\mathbf{D}_k$ is also far away from $\mathbf{D}_1$ and (or) $\mathbf{D}_2$*". This is important to loop-closure: if the matching between two codewords $\mathbf{D}_k$ and $\mathbf{D}_m$ is rejected and no loop-closure hypothesis is triggered, then a real loop-closure is not likely to exist because $\mathbf{D}_k$ must be unable to match with the original features $\mathbf{D}_1, \mathbf{D}_2$ up to the

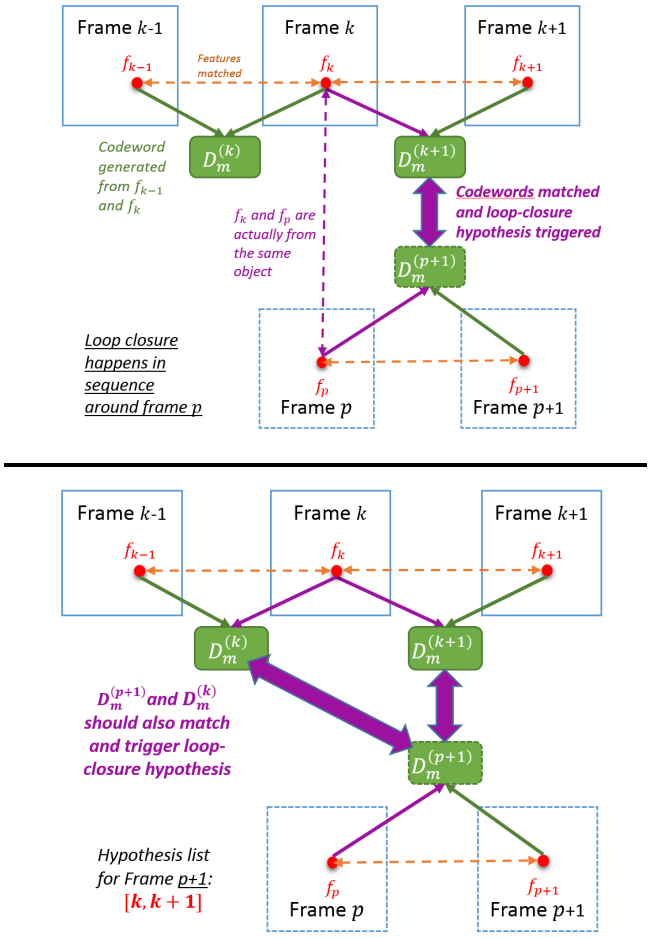

Fig. 5. Illustration of intrinsic temporal constraint on the loop-closure hypotheses.

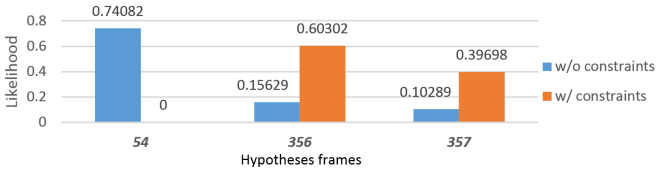

Fig. 6. Temporal constraints rejects the false positive hypothesis frame 54. In this example, the true loop-closure frame is 356.

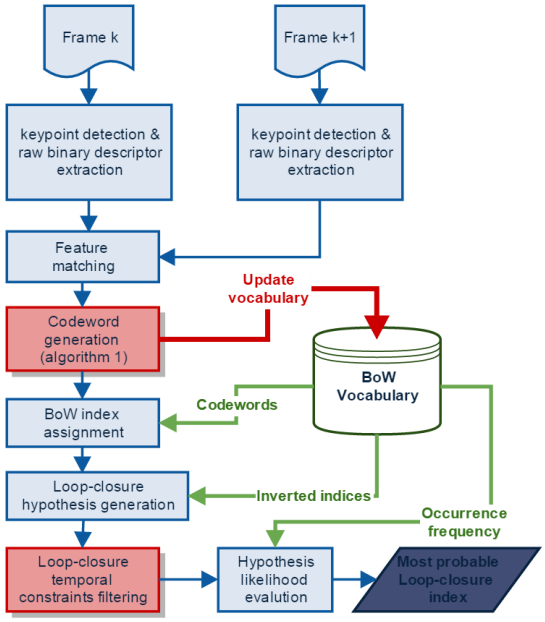

Fig. 7. Diagram of the final loop-closure detection system.

factor λ.

D. Temporal Constraints on Loop-closure Hypotheses

Here we discuss a technique for improving the detection precision, which comes naturally from the codeword learning process. Fig. 5 illustrates a loop-closure happening around frame p, which revisits the same place captured previously around frame k in the sequence. A loop-closure hypothesis closing frame $p+1$ with frame $k+1$ is triggered in the bag-of-words framework, because these two frames share plenty of matched codewords. Assume two codewords $\mathbf{D}_m^{(k+1)}$ and $\mathbf{D}_m^{(p+1)}$ are matched due to the fact that (at least) features f_k and f_p are strongly matched with each other.

If feature f_k is stable across frames $k-1$ and k, and it is also matched between these two frames, then the codeword $\mathbf{D}_m^{(k)}$ generated from f_k for frame k should also match strongly with $\mathbf{D}_m^{(p+1)}$. If there are enough such stable features across frames $k-1$ and k, a loop-closure hypothesis should also be triggered by matching frames $p+1$ with frame k. This results in at least two *consecutive* frame indices existing in the hypothesis list. Therefore, we impose a temporal constraint on the loop-closure hypotheses for a frame,

Constraint: *A loop-closure hypothesis with frame k is accepted, if and only if either frame $k-1$ or frame $k+1$ is also retrieved as a hypothesis.*

The temporal constraint on hypotheses reduces the false positive rate and leads to improved detection precision. Fig.6 illustrates an example of using the temporal constraint. True loop-closure exists in frame 356. The hypothesis list *without* temporal constraints includes frame 54 (with likelihood 0.74), frame 356 (with likelihood 0.16), and frame 357 (with likelihood 0.10). The temporal constraint rejects frame 54 (since neither frame 53 nor 55 are in the hypothesis list), the false positive hypothesis. After renormalization, the temporally constrained hypotheses become frame 356 (with likelihood 0.60) and frame 357 (with likelihood 0.40). Frame 356 is therefore retrieved as the final loop-closure index.

IV. Loop Closure Detection System

Our final system design is based the state-of-the-art system IBuILD [12], but with the integration of the proposed algorithm, as depicted in Fig. 7. The system is an incremental bag-of-word system without any prior offline training process. Here we discuss some details of the system.

- The initial feature matching is based on raw binary descriptors extracted from visual keypoints (FAST is used). The raw binary descriptors are from the same binary tests which are used from the codeword learning.

A local search on the next frame is then performed to find the best matched features. This step would come for free in an actual visual SLAM system.

- Masked Hamming distance is used for distance evaluation in all modules except for the initial feature matching.
- The codeword generation is based on Algorithm 1 . In actual implementation, $\mathbf{x}_1$ and $\mathbf{x}_2$ are already known from the previous steps. They are directly used to create the mask $\mathbf{y}_m$ instead of the raw image patches I_1, I_2.
- A codeword merging step is performed when the learned codewords are used to update the vocabulary. This step will iterate through all the codewords generated for the current frame and find the codeword pairs within matching thresholds. These pairs will then be merged according to Algorithm 1 but treating the two codewords to merge as $\mathbf{D}_1$ and $\mathbf{D}_2$. Contrast this merge to [12] which takes the "numerical centroid". For two binary vectors, the "numerical centroid" is equivalent to the bit-wise OR operation.
- The temporal constraint discussed in Section III-D filters the hypothesis list before the loop-closure likelihood calculations. It is performed with a single-pass scan on the hypothesis list.
- The hypothesis with the highest likelihood is output as the final loop-closure index. We perform a temporal consistency check of $k = 2$ (like [11], Section VI.C).

V. Evaluation

Evaluation used benchmark datasets. To get reasonable parameters, we first performed a coarse-grain parameter sweep on the `CityCentre` set. The experiments on the other data sets involved modifying the matching threshold only. Performance is compared to three other methods: FabMap 2.0 [9], Bag-of-binary-words [11], and IBuILD [12]. The machine used was a Linux desktop with Intel Core i5 quadcore 2.8GHz CPU and 8 GB memory.

A. Datasets and evaluation tool

The data sets used include four challenging sets: `CityCentre` [8], `Malaga09 6L` [4], `New College` [20], `Ford Campus 2` [17]. Table III summarizes their characteristics. Figures 8 to 11 visualize some matched feature pairs that lead to learned codewords. In the `Ford Campus 2` set, part of the vehicle is always visible. We kept only the paired keypoints with pixel coordinate $v \leq 1200$. For benchmarking, the evaluation scripts and ground truth files from [11] are used.

B. Experiment and parameter sweep on CityCentre set

We chose `CityCentre` set for a coarse-grain parameter sweep because it is a difficult dataset to get high recall with 100% precision. The dimensions L of tests and masks are fixed to be 512. The parameter sweep evaluated four parameters: (1) matching threshold Ψ for d_{MH}, (2) keypoints detection threshold Υ, (3) maximum number of matched pairs allowed Γ, and (4) number of local frames excluded

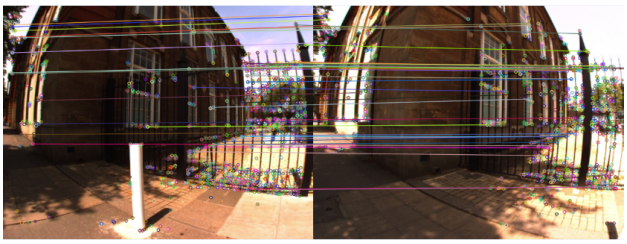

Fig. 8. Example frames in `CityCentre` data set.

Fig. 9. Example frames in `Malaga09 6L` data set.

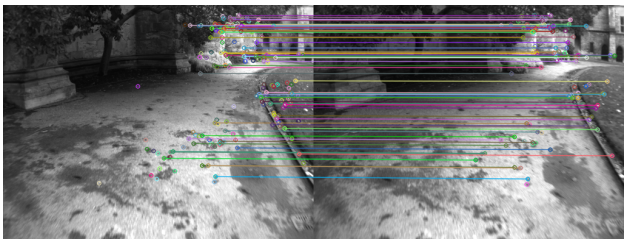

Fig. 10. Example frames in `New College` data set.

Fig. 11. Example frames in `Ford Campus 2` data set.

T_{local}. Among these parameters, we observed T_{local} can cover a large range, i.e. 20–50, with little change in the results. Moreover, Γ mainly controls the size/growth of vocabulary by limiting the accepted pairs of initially matched features. Since it trades off efficiency and precision/recall, we set the value to 100.

The most important parameters are matching threshold Ψ and detection threshold Υ. Figure 12 plots the precision-recall curves with $\Psi = [8, 10, 12, 15, 18, 20, 22, 25]$ under $\Upsilon = [20, 35, 50]$. The best recall with 100% precision happens when $\Psi = 18, \Upsilon = 35$, as listed in Table IV .

TABLE III

DATASETS USED FOR EVALUATION

Dataset	Environment	Distance (m)	Sensor position	Image resolution	Number of frames
CityCentre [8]	Outdoor, urban, dynamic	2025	Lateral	640×480	2474 (left & right cameras)
Malaga09 6L [4]	Outdoor, slightly dynamic	1192	Frontal	1024×768	869
New College [20]	Outdoor, dynamic	2260	Frontal	512×384	5266
Ford Campus 2 [17]	Outdoor, urban, slightly dynamic	4004	Frontal	600×1600	1182

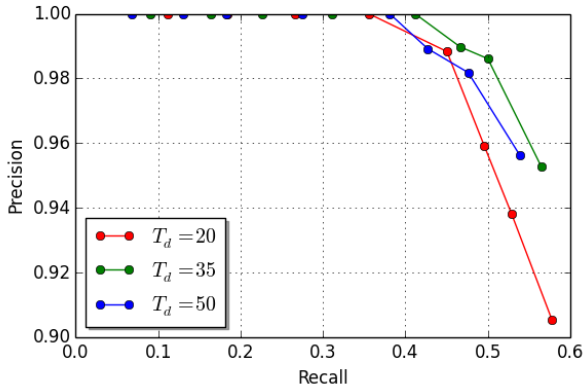

Fig. 12. The precision-recall under different detection thresholds on `CityCentre` set. Each precision-recall curve of a certain detection threshold is plotted by changing the matching threshold for the masked Hamming distances.

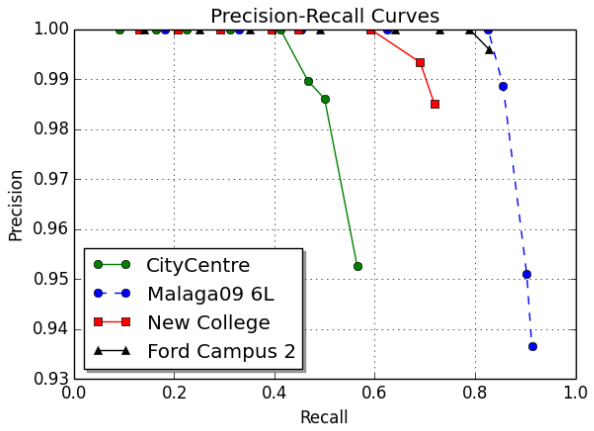

Fig. 13. The precision-recall on different data sets using the proposed approach.

TABLE IV

THE PARAMETER SET WHICH GIVES THE BEST RECALL WITH **100%** **PRECISION** ON CITYCENTRE SET.

Parameters	Values
Matching threshold for d_{MH}, Ψ	18
Keypoints detection threshold, Υ	35
Maximum number of matched pairs allowed, Γ	100
Binary test dimensions, L	512
Number of local frames excluded, T_{local}	20

C. Experiments on all four datasets

In these experiments, the precision and recall are evaluated by changing Ψ, but keeping $\Upsilon = 35, \Gamma = 100, T_{\mathrm{local}} = 20$. The precision-recall curves of our approach is depicted in Figure 13.

Finally, a comparison is provided of our approach to the other approaches [9], [11], [12], focusing on the best recall at 100% precision. The results of other approaches are directly from the referred publications. In particular, for bag-of-binary-words method, the `New College` and `Ford Campus 2` datasets are used as the training sets with parameter searching, while `CityCentre` and `Malaga09 6L` are used as testing set with fixed parameters. For FabMap 2.0 on `Malaga09 6L`, only 462 images are used [11]. It can be observed that except for the `Ford Campus 2`, our approach has the highest recalls at 100% precision.

D. Timing

We collected the timing statistics for learning a codeword as in Algorithm 1 from `CityCentre` experiments. The timing result under normal system process priority is listed in Table VI, and indicates a high efficiency and stability. In our implementation, the bit-wise operations are handled using C++ `std::transform` function with bit operation structure (e.g. `std::bit_xor`) for `uchar`. The number of 1s in a binary vector is counted directly using a look-up table indexed by `uchar` values.

VI. CONCLUSION

This work describes a method to learn binary codewords online for loop-closure detection. The codewords are learned efficiently in an LDA fashion from matched feature pairs in two consecutive frames, such that the learned codewords encode temporal perspective invariance from the observed motion dynamics. The geometric properties of the learned codewords are mathematically justified. The temporal consistency from the nature of learned codewords is further exploited to cull loop-closure hypotheses. The incremental system is evaluated with precision/recall and timing results to quantify the effectiveness and efficiency of the approach.

TABLE V
(USED 462 IMAGES)

Dataset	*Performance (precision/recall) of different approaches*			
	Fab-map 2.0 [9]	**Bag of binary words [11]**	**IBuILD [12]**	**Ours**
CityCentre	100% / 38.77%	100% / 30.61%	100% / 38.92%	100% / 41.18%
Malaga09 6L	100% / 68.52%	100% / 74.75%	100% / 78.13%	100% / 82.61%
New College	Not available	100% / 55.92%	Not available	100% / 59.20%
Ford Campus 2	Not available	100% / 79.45%	Not available	100% / 78.92%

TABLE VI
TIME (IN $\mathbf{10^{-6}}$ SEC.) USED FOR LEARNING ONE CODEWORD USING ALGORITHM 1.

Stat.	Mean	Standard Dev.	Min	Max
Time Used	14.60	0.76	13.10	16.04

ACKNOWLEDGMENT

We'd like to thank Summer Undergraduate Research in Engineering (SURE) grant (NSF award number: EEC-1263049) for supporting Mason Lilly. We sincerely thank the authors of [12] for sharing the main components of his IBuILD implementation, and thank the authors of [11] for providing their evaluation scripts and ground truth.

REFERENCES

[1] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *Robotics, IEEE Transactions on*, 24(5):1027–1037, 2008.

[2] V. Balntas, L. Tang, and K. Mikolajczyk. BOLD-binary online learned descriptor for efficient image matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2367–2375, 2015.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3):346–359, 2008.

[4] J. Blanco,F. Moreno, and J. Gonzalez. A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, 27(4):327–351, 2009.

[5] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):43–57, 2011.

[6] H. Cai, K. Mikolajczyk, and J. Matas. Learning linear discriminant projections for dimensionality reduction of image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):338–352, 2011.

[7] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision*, pages 778–792. Springer, 2010.

[8] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[9] M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0 *The International Journal of Robotics Research*, 30(9):1100–1123, 2011.

[10] J. Dong and S. Soatto. Domain-size pooling in local descriptors: DSP-SIFT. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5097–5106. IEEE, 2015.

[11] D. Gálvez-López and J. Tardós. Bags of binary words for fast place recognition in image sequences. *Robotics, IEEE Transactions on*, 28(5):1188–1197, Oct 2012.

[12] S. Khan and D. Wollherr. IBuILD: Incremental bag of binary words for appearance based loop closure detection. In *IEEE International Conference on Robotics and Automation*, pages 5441–5447, May 2015.

[13] K. Konolige, J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. *The International Journal of Robotics Research*, 2010.

[14] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *IEEE International Conference on Computer Vision*, pages 2548–2555. IEEE, 2011.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[16] R. Mur-Artal, J. Montiel, and J. D. Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *arXiv preprint arXiv:1502.00956*, 2015.

[17] G. Pandey, J. McBride, and R. Eustice. Ford campus vision and lidar data set. *The International Journal of Robotics Research*, 30(13):1543–1552, 2011.

[18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to sift or surf. In *IEEE International Conference on Computer Vision*, pages 2564–2571. IEEE, 2011.

[19] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1573–1585, 2014.

[20] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman. The new college vision and laser data set. *The International Journal of Robotics Research*, 28(5):595–599, 2009.

[21] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit. Boosting binary keypoint descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2874–2881. IEEE, 2013.

[22] T. Trzcinski and V. Lepetit. Efficient discriminative projections for compact binary descriptors. In *European Conference on Computer Vision*, pages 228–242. Springer, 2012.

[23] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 178–185. IEEE, 2009.

[24] G. Zhang and P. A. Vela. Good features to track for visual slam. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1373–1382, 2015.

[25] G. Zhang and P. A. Vela. Optimally observable and minimal cardinality monocular SLAM. In *IEEE International Conference on Robotics and Automation*, pages 5211–5218, 2015.