An Online Sparsity-Cognizant Loop-Closure Algorithm for Visual Navigation

Yasir Latif*, Guoquan Huang†, John Leonard†, and José Neira*
*Instituto de Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza, Zaragoza, Spain
Email: {ylatif, jneira}@unizar.es
†Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology, Cambridge, MA 02139, USA
Email: {gqhuang, jleonard}@mit.edu

***Abstract*—It is essential for a robot to be able to detect revisits or *loop closures* for long-term visual navigation. A key insight is that the loop-closing event inherently occurs sparsely, i.e., the image currently being taken matches with only a small subset (if any) of previous observations. Based on this observation, we formulate the problem of loop-closure detection as a *sparse, convex* ℓ_1-minimization problem. By leveraging on fast convex optimization techniques, we are able to efficiently find loop closures, thus enabling real-time robot navigation. This novel formulation requires no offline dictionary learning, as required by most existing approaches, and thus allows *online incremental* operation. Our approach ensures a *global, unique* hypothesis by choosing only a single globally optimal match when making a loop-closure decision. Furthermore, the proposed formulation enjoys a *flexible* representation, with *no* restriction imposed on how images should be represented, while requiring only that the representations be close to each other when the corresponding images are visually similar. The proposed algorithm is validated extensively using public real-world datasets.**

I. Introduction

With a growing demand for autonomous robots in a range of applications, such as search and rescue [30, 8], space and underwater exploration [7], it is essential for the robots to be able to navigate accurately for an extended period of time in order to accomplish the assigned tasks. To this end, the ability to detect revisits (i.e., *loop closure* or place recognition) becomes necessary, since it allows the robots to bound the errors and uncertainty in the estimates of their positions and orientations (poses). In this work, we particularly focus on loop closure during visual navigation, i.e., given a camera stream we aim to efficiently determine whether the robot has previously seen the current place or not.

Even though the problem of loop closure has been extensively studied in the visual-SLAM literature (e.g., see [13, 19]), most existing algorithms typically require the *offline* training of visual words (dictionary) from *a priori* images acquired previously in the same environment. This clearly is not always the case when a robot operates in an unknown environment. In general, it is difficult to reliably find loops in visual-appearance space. One particular challenge is the perceptual aliasing, i.e., while images may be similar in appearance, they might be coming from different places. To mitigate this issue, both temporal (i.e., loops will only be considered closed if there are other loops closed nearby) and geometric constraints (i.e., if a loop has to be considered closed, a valid transformation must exist between the matched images) can be employed [19]. In particular, the method proposed in [19] decides on the quality of a match *locally*: If the match with the highest score in some distance measure is away from the second highest, it is considered a valid candidate. However, such local information may lead to incorrect decisions even if the temporal and geometric consistency checks are applied. This is due to the fact that both temporal and geometric conditions can easily fail in a highly self-similar environment (e.g., corridors in a hotel)

To address the aforementioned issues, in this paper we introduce a *general online* loop-closure approach for vision-based robot navigation. In particular, by realizing that loops typically occur intermittently in a navigation scenario, we, for the first time, formulate loop-closure detection as a sparse ℓ_1-minimization problem that is convex. By leveraging on the fast convex optimization techniques, we subsequently solve the problem efficiently and achieve real-time frame-rate generation of loop-closure hypotheses. Furthermore, the proposed formulation enjoys *flexible* representations and can generate loop-closure hypotheses regardless of what the extracted features represent. That is, any discriminative information, such as descriptors, Bag of Words (BoW), or even whole images, can be used for detecting loops. Lastly, we shall stress that our proposed approach declares a loop that is valid only when it is *globally unique*, which ensures that if perceptual aliasing is being caused by more than one previous image, no loop closure will be declared. Although this is conservative in some cases, since a false loop closing can be catastrophic while missing a loop closure generally is not, ensuring such global uniqueness is necessary and important, in particular, in highly self-similar environments.

II. Related Work

The problem of loop-closure detection has been extensively studied in the SLAM literature and various solutions have been proposed over the years for visual navigation (e.g., see [20] and references therein). In this work, we focus only on visual-appearance-based methods. In particular, Cummins and Newman [13] proposed FAB-MAP, a probabilistic appearance-based approach using visual BoW for place recognition, and

showed it to work robustly over trajectories up to 1000 km. Similarly, Galvez-Lopez and Tardos [19] introduced a Binary-BoW (BBoW)-based method, which detects FAST [28] keypoints and employs a variation of the BRIEF [6] descriptors to construct the BoW. A verification step is further enforced to geometrically check the features extracted from the matched images. It should be pointed out that both methods [13, 19] learn the BoW dictionaries beforehand, which is used later for detecting loop closures when the robots actually operates in the field. In contrast, the proposed approach builds the dictionary *online* as the robot explores an unknown environment, while at the same time efficiently detecting loops (if any). Moreover, rather than solely relying on the descriptors-based BoW, our method is flexible and can utilize *all* pixel information to discriminate places even in presence of dynamic objects (encoded as sparse errors), any descriptor that can represent similar places, or any combination of such descriptors.

Recently, some work has focused on loop closure under extreme changes in the environment, such as different weather conditions or different times of the day. In particular, Milford and Wyeth [24] proposed SeqSLAM, which is able to localize with drastic lighting and weather changes by matching sequences of images with each other as opposed to single images. Churchill and Newman [12] introduced the experience-based maps that learn the different appearances of the same place as it gradually changes in order to perform long-term localization. In addition, new images are also discovered in order to attain better localization [26]. In these cases, if the information invariant to such changes can be obtained, the proposed formulation can be utilized to obtain loop-closure hypotheses. Thus, this work essentially focuses on finding loop closures given some discriminative descriptions (e.g., descriptors, and whole images), without explicitly assuming a specific type of image representations.

III. Sparse Optimization for Loop Closure

In this section, we formulate loop-closure detection as a sparse optimization problem based on a sparse and redundant representation. Such representations have been widely used in computer vision for problems such as denoising [18], deblurring [17], and face recognition [11]. Similarly, in SLAM, [9, 10] used an ℓ_1-norm based formulation for back-end optimization. To the best of our knowledge, no prior work has yet investigated this technique for loop closure in robot navigation. The key idea of this approach is to represent the problem *redundantly*, from which a *sparse* solution is sought for a given observation.

Specifically, suppose we have the current image represented by a vector $\boldsymbol{b} \in \mathcal{R}^n$, which can be either the vectorized full raw image or the sparse descriptors extracted from the image. Suppose that we also have a dictionary denoted by $\mathbf{B} = [\boldsymbol{b}_1 \ \cdots \ \boldsymbol{b}_m] \in \mathcal{R}^{n \times m}$, which consists of m basis vectors of the same type as $\boldsymbol{b}$. Thus, solving the linear equation $\mathbf{B}\boldsymbol{x} = \boldsymbol{b}$ yields the representation of $\boldsymbol{b}$ in the basis $\mathbf{B}$ in the form of the vector $\boldsymbol{x} \in \mathcal{R}^m$. Elements of $\boldsymbol{x}$ indicate which basis vectors, $\boldsymbol{b}_i$, best explain $\boldsymbol{b}$ and how much the corresponding contributions are. A zero contribution simply implies that the corresponding basis vector is irrelevant to $\boldsymbol{b}$. One trivial example of a dictionary is the $n \times n$ identity matrix, $\mathbf{B} = \mathbf{I}_n$, which gives us the same representation, i.e., $\mathbf{I}_n\boldsymbol{x} = \boldsymbol{b} \Rightarrow \boldsymbol{x} = \boldsymbol{b}$. It is important to note that we have made *no* assumption about what the dictionary contains, and in general, any arbitrary basis including random matrices or wavelet basis can be used as a dictionary.

We know that a general vector $\boldsymbol{b}$ can be represented by a basis matrix $\mathbf{B}$ if and only if it belongs to the range space of $\mathbf{B}$, i.e., $\exists \boldsymbol{x} \neq \mathbf{0}, \ \text{s.t.} \ \mathbf{B}\boldsymbol{x} = \boldsymbol{b}$. Carefully choosing these bases may result in a sparse representation, i.e., $\boldsymbol{x}$ is sparse. This is often the case in practice, because the signal is naturally sparse when represented in a certain basis. For instance, when representing an image using the wavelet basis, there are only a few coefficients that are nonzero. Nevertheless, a vector may not be representable by the basis, for example, if the basis matrix (dictionary) is rank-deficient and the vector is in its null space. To exclude such singular cases, in this work, we assume that the image vector $\boldsymbol{b}$ is always representable by the basis matrix which is of full row rank (i.e., $\text{rank}(\mathbf{B}) = n$). Moreover, we allow the basis to be redundant, that is, we may have more (not necessarily orthogonal) basis vectors than the dimension of the image vector (i.e., $m > n$). Note that this assumption can be satisfied by carefully designing the dictionary (see Section IV). In general, a redundant dictionary leads to the sparse representation of the current image, which is what we seek and describes the sparse nature of the loop-closure events (i.e., occurring sparsely).

Consider that we have $m > n$ basis vectors, then $\mathbf{B}\boldsymbol{x} = \boldsymbol{b}$ becomes an under-determined linear system and has infinitely many solutions. Therefore, we have to regularize it in order to attain a unique solution by specifying a desired criterion that the solution should satisfy. Typically, this regularization takes the form of looking for a solution, $\boldsymbol{x}^\star$, that leads to the minimum reconstruction error in the ℓ_2-norm sense, which is the least-squares formulation:

$$\min_{\boldsymbol{x}} \ \|\mathbf{B}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \quad \Rightarrow \quad \boldsymbol{x}^\star = \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}\boldsymbol{b} \qquad (1)$$

Note that ℓ_2-norm is widely used in practice in part because of the closed-form unique solution, while leading to a *dense* representation, i.e., almost all of the elements of $\boldsymbol{x}^\star$ are non-zero and thus all the basis vectors are involved in representing the current image vector.

Due to the fact that loop-closure events often occur sparsely, we instead aim to find the *sparsest* possible solution under the condition that it best explains the current image vector. Intuitively, by assuming that the basis dictionary consists of all the previous observed images and thus is redundant, we are looking for the smallest possible subset of previous images that can best explain our current image. The smallest such subset would contain just a single image which is "closest" to the current image (in appearance or descriptor space) under the assumption that there exists a unique image from the past which matches the current image.

To that end, we can employ the ℓ_0-norm to quantify the sparsity of a vector, i.e., $\|\boldsymbol{x}\|_0 = \text{Card}(\boldsymbol{x} : \forall i, x_i \neq 0)$, the total number of non-zero elements in $\boldsymbol{x}$. Note that a vector with d non-zero elements is called d-sparse. Thus, the problem of

loop closure can be formulated as follows:

$$\min_{\boldsymbol{x}} \quad \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad \mathbf{B}\boldsymbol{x} = \boldsymbol{b} \tag{2}$$

The above problem is a combinatorial optimization problem which in general is NP-hard [1], because all of the possible d-sparse vectors have to be enumerated to check whether they fulfill the constraint. To address this computational-intractability issue, we relax the ℓ_0-norm in (2) to ℓ_1-norm, which is defined as the summation of absolute values of the elements of $\boldsymbol{x}$, $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|$, since it is well-known that ℓ_1-norm results in a sparse solution [15], i.e.,

$$\min_{\boldsymbol{x}} \quad \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \mathbf{B}\boldsymbol{x} = \boldsymbol{b} \tag{3}$$

The problem (3) assumes the perfect reconstruction without noise, which clearly is not the case in practice. Hence, we introduce a sparse noise (error) term along with the basis vector to explain the current image, i.e.,

$$\min_{\boldsymbol{x},\boldsymbol{e}} \quad \|\boldsymbol{x}\|_1 + \|\boldsymbol{e}\|_1 \quad \text{subject to} \quad \mathbf{B}\boldsymbol{x} + \boldsymbol{e} = \boldsymbol{b} \tag{4}$$

$$\Rightarrow \min_{\boldsymbol{\alpha}} \quad \|\boldsymbol{\alpha}\|_1 \quad \text{subject to} \quad \mathbf{D}\boldsymbol{\alpha} = \boldsymbol{b} \tag{5}$$

where $\mathbf{D} := \begin{bmatrix} \mathbf{I}_n & \mathbf{B} \end{bmatrix}$ and $\boldsymbol{\alpha} := \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{x} \end{bmatrix}$. Note that we normalize the basis vectors when building the dictionary $\mathbf{D}$, and thus $\mathbf{I}_n$ can be considered as the noise basis along each of the n directions of the data space, and $\boldsymbol{e}$ becomes an indication variable for which noise components dominate the reconstruction error. This allows us to normalize $\boldsymbol{x}$ and $\boldsymbol{e}$ together when computing contributions of data and noise bases (see Section IV-A). We stress that this new formulation of loop closure (5) takes advantage of the fact that ℓ_1-norm automatically promotes sparsity, as opposed to the more commonly used ℓ_2-norm [3]. By finding the minimum ℓ_1-norm solution of (5), we are in effect seeking an explanation of our current image that uses the fewest basis vectors from the redundant set of basis. This problem is also known as atomic decomposition [18], since $\boldsymbol{b}$ is decomposed into its constituent atoms in the dictionary.

IV. Closing Loops via ℓ_1-minimization

In the preceding section, we have formulated loop closure as a sparse convex ℓ_1-minimization problem. In this section, we present in detail how this novel formulation can be utilized in monocular-vision-based robot navigation. In what follows, we first explain how a dictionary can be constructed incrementally online, and then subsequently how such a dictionary can be used at each time step to generate loop-closure hypotheses.

A. Building the Dictionary

The first step in the process of detecting loops is to build a set of basis vectors that make up the dictionary. Unlike state-of-the-art loop-closure detection methods (e.g., [19]), the proposed approach does *not* learn the dictionary offline before the start of the experiment. Instead, the dictionary is built exclusively for the current experiment, as the robot moves and collects images, that is, incrementally *online* as images arrive.

As a new image becomes available, a mapping function, $\boldsymbol{f} : \mathcal{R}^{(r,c)} \rightarrow \mathcal{R}^n$, transforms the image of resolution $r \times c$ to a unit vector of dimension n. Due to the flexibility of representation enjoyed by our proposed approach, this function is general and can be either a whole image descriptors such as HOG [14] and GIST [25], or local descriptors such as SIFT [21] and SURF [4] computed over the image. That is, the basis vectors can represent any information that helps distinguish between two images. The proposed method can be considered as data association in a high-dimensional space carried out by (approximately) reconstructing a given vector from a subset of unit vectors in that space. As such, this approach is agnostic to what these vectors physically represent. For this reason, versatility of representation is inherent to our algorithm, allowing representation ranging from whole images to descriptors, BoW, or even mean and variance normalized images over time for matching sequences across different times of day or changing weather conditions. Also, since we use ℓ_2-norm of the error to measure how good the reconstruction is, any descriptor whose distance between two vectors is measured in term of ℓ_2-norm, can be naturally incorporated in the proposed approach.

In order to ensure full row rank of the dictionary matrix $\mathbf{D}$, we initialize the dictionary with an identity matrix $\mathbf{I}_n$, which also accounts for the basis of the noise $\boldsymbol{e}$ [see (5)]. When the first image encoded by $\boldsymbol{i}_1$ arrives, $\boldsymbol{b}_1 = \boldsymbol{f}(\boldsymbol{i}_1)$ is added to the dictionary. In general, updating the dictionary at the i-th time step is simply appending $\boldsymbol{b}_i = \boldsymbol{f}(\boldsymbol{i}_i)$ to the end of the current dictionary[1]. However, before augmenting the dictionary, we need to determine whether or not there are some previous images that explain the current one, i.e., we need to detect any loop that can be closed based on the current image.

B. Solving ℓ_1-Minimization

Once the dictionary is available and when we obtain an image at every time step, we are now ready to solve the sparse ℓ_1-minimization problem (5) in order to find (if any) loop closures. Since this is a convex optimization problem, various approaches such as the primal-dual method are available for solving it [5], while in this work, we focus on the homotopy approach primarily due to its efficiency [22, 16].

The homotopy method is specifically designed to take advantage of the properties of ℓ_1-minimization. In particular, by relaxing the equality constraint in (5), we have the following *constrained* minimization problem:

$$\min_{\boldsymbol{\alpha}} \quad \|\boldsymbol{\alpha}\|_1 \quad \text{subject to} \quad \|\mathbf{D}\boldsymbol{\alpha} - \boldsymbol{b}\|_2 \leq \epsilon \tag{6}$$

where $\epsilon > 0$ is a pre-determined noise level. This is termed the basis pursuit denoising problem in compressive sensing [15]. A variant of (6) is the following *unconstrained* minimization that is actually solved by the homotopy approach:

$$\min_{\boldsymbol{\alpha}} \quad \lambda \|\boldsymbol{\alpha}\|_1 + \frac{1}{2} \|\mathbf{D}\boldsymbol{\alpha} - \boldsymbol{b}\|_2 \tag{7}$$

where λ is a scalar weighting parameter. To solve (7), the homotopy method uses the fact that the objective function undergoes a homotopy continuation from the ℓ_2 constraint to the ℓ_1 cost as λ increases [16]. The computational complexity

[1]Although this simple augmentation would make the dictionary grow unbounded, more sophisticated update policy (e.g., replacing or merging basis vectors for the same locations) can be designed in order to control the size of the dictionary when a robot repeatedly operates in the same environment.

Algorithm 1 Closing Loops via ℓ_1-Minimization

Input: Dictionary $\mathbf{D}_{i-1}$, Current image $\boldsymbol{i}_i$, Threshold τ, Weight λ, Ignoring-time window t_g
Output: Loop-closure hypotheses $\mathbf{H}$, Updated dictionary $\mathbf{D}_i$

$\boldsymbol{b}_i := \boldsymbol{f}(\boldsymbol{i_i})$
Hypothesis generation:
Solve $\min_{\boldsymbol{\alpha}_i} \ \lambda \|\boldsymbol{\alpha}_i\|_1 + \frac{1}{2}\|\mathbf{D}_{i-1}\boldsymbol{\alpha}_i - \boldsymbol{b}_i\|_2$ using the homotopy approach (see Section IV-B)
Normalize $\hat{\boldsymbol{\alpha}}_{\boldsymbol{i}} := \frac{\boldsymbol{\alpha_i}}{\|\boldsymbol{\alpha_i}\|_2}$
Find hypotheses $\mathbf{H} := \{j \mid \hat{\alpha}_{i,j} > \tau, \ \|i-j\|_1 > t_g\}$
Dictionary update:
$\mathbf{D}_i := [\ \mathbf{D}_{i-1} \quad \boldsymbol{b}_i\ ]$

of this approach is $O(dn^2 + dnm)$ for recovering a d-sparse signal in d steps, although in the worst case when recovering a non-sparse solution in a high-dimensional observational space and large number of basis vectors, it can perform as worse as $O(m^3)$, which fortunately is not the case in this work.

The above homotopy solver is employed to determine loop closure for the i-th image represented by $\boldsymbol{i_i}$, by solving (7) for $\boldsymbol{f(i_i)}$ using the update-to-date dictionary $\mathbf{D}_{i-1} = [\ \mathbf{I}_n \quad \boldsymbol{f(i_1)} \quad \dots \quad \boldsymbol{f(i_{i-1})}\ ]$.[2] The solution $\boldsymbol{\alpha_i} = \begin{bmatrix}\alpha_{i,1} & \cdots & \alpha_{i,n+i-1}\end{bmatrix}^T$ at the i-th time step contains the contribution of each previous basis in constructing the current image. To find a unique image to close a loop with, we are interested in which basis has the greatest relative contribution, which can be found by calculating the unit vector $\hat{\boldsymbol{\alpha}}_{\boldsymbol{i}} = \boldsymbol{\alpha_i}/\|\boldsymbol{\alpha_i}\|_2$. Any entry greater than a predefined threshold, τ, is considered a loop-closure candidate. In addition, due to the fact that in a visual navigation scenario, the neighbouring images are typically overlapped with the current image and thus have great "spurious" contributions, we explicitly ignore a time window, t_g, around the current image, during which loop-closure decisions are not taken. This is a design parameter and can be chosen based on the camera frequency (fps) and robot motion. Once the decision is made, the dictionary is updated by appending $\boldsymbol{f(i_i)}$ to it, i.e., $\mathbf{D}_i = [\ \mathbf{D}_{i-1} \quad \boldsymbol{f(i_i)}\ ]$. The main steps of this process are summarized in Algorithm 1.

C. Remarks

1) Global uniqueness: It is important to note that the solution to (7), by construction, is guaranteed to be sparse. In the case of no perceptual aliasing, the solution is expected to be 1-sparse, because only one image in the dictionary should match the current image when a revisit occurs. In the case of exploration where there is no actual loop-closure match in the dictionary, the current image is best explained by the last observed and the solution hence is still 1-sparse.

In a general case where $k > 1$ images that have been previously observed and that are visually similar to the current image, a naive thresholding method – which simply compares the current image to each of the previous ones based on some similarity measure – would likely produce k loop-closure hypotheses corresponding to the k images in the dictionary. However, this thresholding calculates the contribution of each previous image *without* taking into account the effects of other images or data noise. This can be considered as *decoupled* computation of contributions – despite the fact that due to noise they may be correlated – and thus becomes *suboptimal*, while an erroneous loop closure may be catastrophic for the navigation (estimation) algorithm. In contrast, the proposed ℓ_1-minimization-based approach guarantees the *unique* hypothesis, by selecting the j-th image with the greatest $\hat{\alpha}_{i,j}$, where $\hat{\alpha}_i$ is the *global* optimal solution of the convex problem (7), at step i that *simultaneously* considers all the contributions from all the previous images and noise. In the case of multiple revisits to the same location, the proposed approach, as presented here, is *conservative*. Including the corresponding images from earlier visits in the dictionary would lead to a non-unique solution, when the same location is revisited again. However, the proposed method can be easily extended to detect loops on multiple revisits. Instead of considering the contribution of all the previous basis separately, if a loop exists between previous locations k and l, we consider their joint contribution $(\hat{\alpha}_{i,k} + \hat{\alpha}_{i,l})$ when making the decision. This ensures that even though these places are not individually unique enough to explain the current image, together (and since they are visually similar as we already have a loop closure between them), they best explain the current observation, allowing us to detect loop closures in case of multiple revisits.

2) Flexible basis representation: We stress that the dictionary representation used by the proposed approach is general and flexible. Although we have focused on the simplest basis representation using the down-sampled whole images, this does not restrict our method only to work with this representation. In fact, any discriminative feature that can be extracted from the image (such as GIST, HOG, and so on) can be used as dictionary basis for finding loops, thus permitting the desired properties such as view and illumination invariance. To show that, one particular experiment has been performed in Section V, where the basis vectors are the HOG descriptors computed over the whole image. Moreover, it is not limited to a single representation at a time. If we have k descriptors $\boldsymbol{f}_i : \mathcal{R}^{(r,c)} \rightarrow \mathcal{R}^{k_i}$, a "unified" descriptor can be easily formed by stacking them up in a vector in $\mathcal{R}^K$ ($K = \sum_{i=1}^{k} k_i$). Therefore, our proposed method can be considered as a *generalized* loop-closing approach that can use any basis vector as long as a metric exists to provide the distance between them.

3) Robustness: It is interesting to point out that sparse ℓ_1-minimization inherently is robust to *data noise*, which is widely appreciated in computer vision (e.g., see [11]). In particular, the sparse noise (error) term in (5) can account for the presence of dynamic changes or motion blurs. For example, in Fig. 1 the dominant background basis explains most of the image, while the dynamic elements (which have not been observed before) can be represented by the sparse noise, and Fig. 2 shows that the proposed approach robustly finds these loops. Such robust performance becomes necessary particularly for long-term mapping where the environment often gradually changes over time and thus reliable loop closure in presence of such changes is essential.

[2]The subscript $i-1$ is hereafter used to denote the time index and thus to reveal the online incremental process of the proposed approach.

Fig. 1: Sample images from the New College dataset: Query images (top) and the corresponding match images(bottom). The images are down-sampled to 8×6 pixels. Note that in spite of dynamic changes and motion blurs occurring in these images which deteriorate the loop-closure problem, the proposed approach still provides reliable results.

As a final remark, the proposed ℓ_1-minimization-based loop-closure algorithm is also robust to *information loss*, which is closely related to the question raised by Milford [23]: How much information is needed to successfully close loops? In this work, we have empirically investigated this problem by down-sampling the raw images (which are used as the basis of the dictionary) without any undistortion and then evaluating the performance of the proposed approach under such an adverse circumstance. As shown in Section V, truly small raw images, even with size as low as 48 pixels, can be used to reliably identify loops, which agrees with [23].

V. Experimental results

In this section, we perform a set of real-world experiments on the publicly-available datasets to validate our proposed ℓ_1-minimization-based loop-closure algorithm. In particular, a qualitative test is conducted on the New College dataset [29], where we examine the different types of basis (raw images and descriptors) in order to show the flexibility of basis representation of our approach as well as the robustness to dynamics in the scene. Subsequently, we evaluate the proposed method on the RAWSEEDS dataset [27] and focus on the effects of the design parameters used in the algorithm.

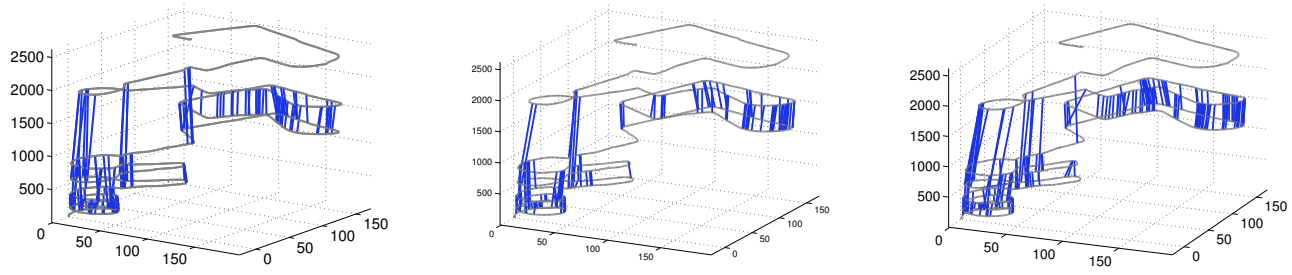
A. New College: Different Types of Basis

The New College dataset [29] provides stereo images at 20 Hz along a 2.2 km trajectory, while in this test we only use every 20 frames with an effective frame rate of 1 Hz and in total 2624 images. Each image originally has a resolution of 512×384, but here is down-sampled to either 64×48 or 8×6 pixels. We show below that even under such adverse circumstance, the proposed approach can reliably find the loops. The image is scaled so that its gray levels are between zero and one, and then is vectorized and normalized as a unit column vector. For the results presented in this test, we use the threshold $\tau = 0.99$ and the weighting parameter $\lambda = 0.5$. Due to the fact that neighbouring images typically are similar to the current one and thus generate false loop closures, we ignore the hypotheses within in a certain time window from the current image and set $t_g = 10$ sec, which effectively excludes the spurious loops when reasoning about possible closures. Note that t_g can be chosen according to speed of the robot as well as the frame-rate at which the loop closing algorithm is working. We also eliminate random matches by enforcing a temporal consistency check, requiring at least one more loop closure within a time window from the current match. We ran all the experiments in Matlab on a Laptop with Core-i5 CPU of 2.5GHz and 16 GB RAM, and use the homotopy-based method [2] for solving the optimization problem (7).

The qualitative results are shown in Fig. 2 where we have used *three* different basis, i.e, down-sampled 64×48 and 8×6 raw images,[3] and GIST descriptors. In these plots, the odometry-based trajectory provided by the dataset is superimposed by the loop closures detected by the proposed approach, which are shown as vertical lines connecting two locations where a loop is found. All the lines parallel to the z-axis represent loop closures that connect the same places at different times. Any false loops would appear as non-vertical lines, and clearly do not appear in Fig. 2, which validates the effectiveness of the proposed method in finding correct loops. These results clearly show the flexibility of basis representation of the proposed method. In particular, instead of using the different down-sampled raw images as basis, our approach can use the GIST descriptors, $\mathbf{GIST}(\mathbf{b}) \in \mathcal{R}^{256}$, which are computed over the whole image, and is able to detect the same loop closures as with the raw images.

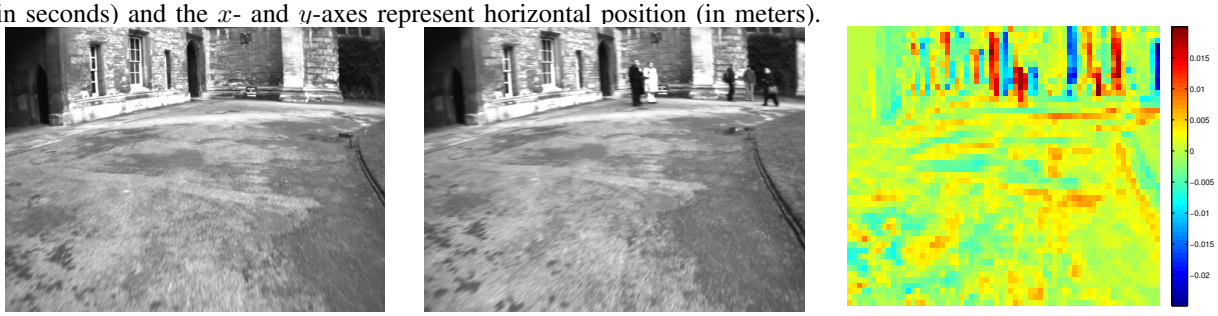
An interesting way of visualizing the locations where loop closures occur is to examine the sparsity pattern of the solution matrix, which is obtained by stacking all the solutions, $\hat{\boldsymbol{\alpha}}_i$, for all the queried images in a matrix. Fig. 4 shows such matrix that contains non-zero values in each column corresponding to the elements greater than the threshold τ. In the case of no loop closure, each image can be best explained by its immediate neighbour in the past, which gives rise to non-zeros along the main diagonal. Most importantly, the off-diagonal non-zeros

[3] In part due to the unoptimized implementation in Matlab, it is too costly to use the vectorized full-sized raw images that would have dimension of $512 \times 384 = 196608$ as the basis for our proposed approach. Thus, we have employed down-sampled raw images of different sizes as the simplest possible basis to show the effectiveness as well as the robustness of our method.



(a) Basis used as raw 64×48 images (b) Basis used as raw 8×6 images (c) Basis used as GIST descriptors

Fig. 2: Loop closures detected by the proposed approach using two different bases for the New College dataset. In these plots, visual odometry provided with the dataset is shown in gray while the loop closures are shown in blue. The z-axis represents time (in seconds) and the x- and y-axes represent horizontal position (in meters).



(a) Query image (b) Match image (c) Noise contributions

Fig. 3: A typical dynamic scenario in the New College dataset: When querying a current image to the dictionary that uses 64×48 raw images as its basis, the proposed approach robustly finds the correct match – which however is contaminated with moving people – by modelling the dynamics as noise.

indicate the locations where loops are closed. It is interesting to see that there are a few sequences of loop closures appearing as off-diagonal lines in Fig. 4. This is due to the fact that the first three runs in the circular area at the beginning of the dataset, correspond to the three off-diagonal lines in the top-left of the matrix; while a sequence of loop closures detected in the lower part of New College, correspond to the longest line parallel to the main diagonal.

It is important to note that although both dynamic changes and motion blurs occur in the images, the proposed approach is able to reliably identify the loops (e.g., see Fig. 1), which is attributed to the sparse error used in the ℓ_1-minimization [see (5)]. To further validate this robustness to dynamics, Fig. 3 shows a typical scenario in the New College where we query a current image with no dynamics to the dictionary that uses 64×48 down-sampled raw images as its basis, and the correct match is robustly found, which however contains moving people. Interestingly, the dominant noise contributions (blue) as shown in Fig. 3(c), mainly correspond to the locations where the people appear in the match image. This implies that the sparse error in (5) correctly models the dynamic changes.

B. RAWSEEDS: Effects of Design Parameters

To further test the proposed algorithm, we use the Bicocca 25b dataset from the RAWSEEDS project [27]. The dataset provides the laser and stereo images for a trajectory of 774 m. We use the left image from the stereo pair sampled at 5 Hz, resulting in a total of 8358 images. Note that we do *not* perform any undistortion and work directly with the raw images coming from the camera. In this test, we focus on studying the effects of the most important parameters used in the proposed approach, and evaluate the performance based on precision and recall. Specifically, precision is the ratio of correctly detected loop closures over all the detections. Thus, ideally we would like our algorithm to work at full precision. On the other hand, recall is the percentage of correct loop closures that have been detected over all possible correct detections. A high recall implies that we are able to recover most of the loop closures.

1) Threshold τ and weight λ: We first examine the acceptance threshold τ, whose valid values range from 0.5 to 1. This parameter can be thought of as the the similarity measure between the current image and the matched image in the dictionary. In order to study the effect of this parameter on the precision and recall, we vary the parameter for a fixed image of 20×15 pixels. Moreover, we are also interested in if and how the weighting parameter λ impacts the performance and thus vary this parameter as well.

The results are shown in Fig. 5. As expected, the general trend is that a stricter threshold (closer to 1) leads to higher precision, and as a side effect, a lower recall. This is because as the threshold increases, we get fewer loop closing hypotheses but a larger proportion of them is correct. Note that this dataset is challenging due to the perceptual aliasing in many parts of the trajectory; the matched images are visually similar but considered as false positives since the robot is physically not in the same place. Note also that as seen from Fig. 5, smaller λ leads to higher precision (and lower recall). This is because

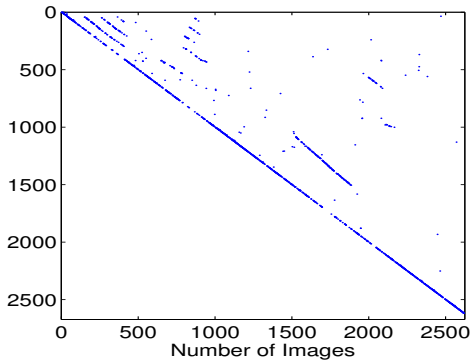


Fig. 4: Sparsity pattern induced by solving (7) for *all* the images in the New College dataset. The i-th column corresponds to the solution for the i-th image, and the non-zeros are the values in each column that are greater than $\tau = 0.99$. Note that the main diagonal occurs due to the current image being best explained by its neighboring image, while the off-diagonal non-zero elements indicate the loop closures.

a smaller value of this parameter results in a better data-fitting (less sparse) solution of (5), thus requiring the images to be as visually similar as possible but at the same time, lowering the contribution of the greatest basis vector.

2) Image size: Inspired by the recent work [23], we also examine the performance difference by varying image sizes and see if we can obtain meaningful results using small-size images. The original image size from the Bicocca dataset is 320×240, and the first image size we consider is 80×60 which is a reduction of a quarter in each dimension. For each new experiment, we half the size in each dimension, which results in images of size 40×30, 20×15, 10×8, and finally 5×4. The weighting parameter λ is fixed to be 0.5. Precision and recall curves are generated by varying the acceptance threshold τ are shown in Fig. 6.

It is clear from Fig. 6 that the curves are tightly coupled and undergo the same behaviour for each image size. Precision curves for the three largest image sizes overlap each other, showing that we can generate the same quality of loop closure hypotheses using any of the image sizes. These plots show a graceful degradation as the image size decreases. Considering that the image of size 10×8 is a factor of 960 times smaller than the original image, our method is able to distinguish places based on very little information, which agrees with [23].

3) Execution time: Since the proposed method solves an optimization problem in a high-dimensional space, it is important to see how long the method takes to come up with the loop-closing hypotheses. Despite that each image is an $r \times c$ vector for an image with r rows and c columns, and at the end of the experiment we have nearly 8500 images, the computation is very efficient thanks to the sparsity induced by the novel formulation. Most of our solutions are expected to be 1-sparse (i.e., we expect only one non-zero if the current image matches perfectly one of the basis vectors in the dictionary), and thus the homotopy-based solver performs efficiently as shown in Table I. For the largest image size, the mean time is 390 ms with a maximum of just over a second. The proposed

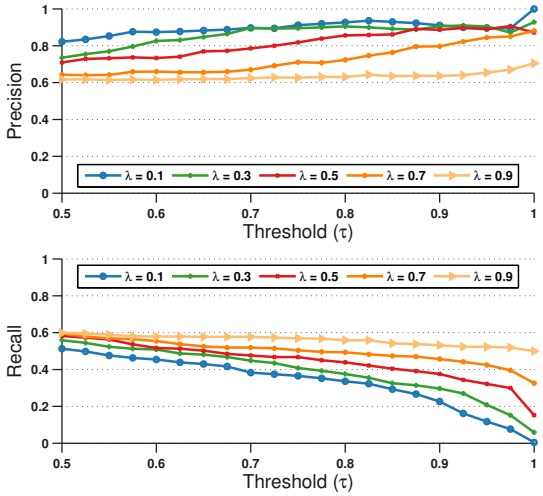


Fig. 5: Precision and recall curves for the Bicocca dataset while using the 20×15 raw images.

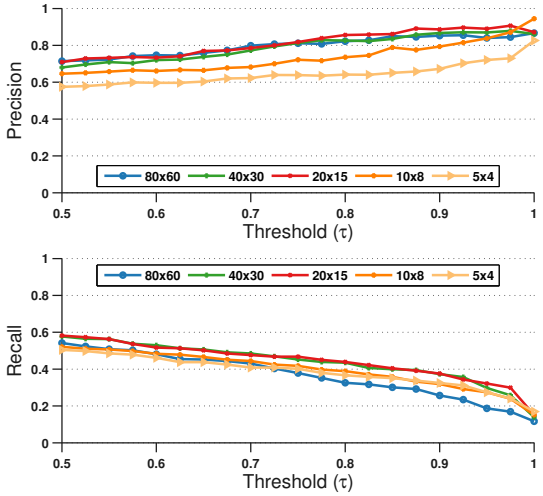


Fig. 6: Precision and recall curves for the Bicocca dataset while fixing the weighting parameter $\lambda = 0.5$.

method works well on small images such as 20×15, which take on average 17 ms. The runtime gradually grows as the number of basis vectors increases. In general, even with the current unoptimized Matlab implementation, the execution is fast enough to be used in real-time operations with runtime well below a second in almost all cases.

Interestingly, we found $\lambda = 0.5$ is a good trade-off between precision/recall and computational cost. In general, a higher threshold τ would lead to fewer high-quality loop closures. This parameter can be designed based on the the application in question. Similarly, images of size larger than 20×15 do not provide great improvement in terms of precision/recall. Thus, the choice of image size should take into account the complexity of the environment being modelled. In an environment (e.g., outdoors) where there is rich textural information, smaller images may be used. If the environment itself does not contain a lot of distinguishable features, larger images can be used in order to be able to differentiate between them.

TABLE I: Execution time for different image sizes

size	min (sec)	mean (sec)	max (sec)	std (sec)
80×60	0.1777	0.3928	1.0495	0.1245
40×30	0.0133	0.0671	0.1768	0.0309
20×15	0.0014	0.0173	0.2544	0.0097
10×8	0.0008	0.0057	0.0502	0.0033
5×4	0.0008	0.0030	0.0922	0.0023

C. Comparison to DBoW

Finally, in this section, we compare the performance of the proposed method against the state-of-the-art DBoW algorithm [19]. For the DBoW, we operate on the full-sized 320×240 images, using different temporal constraints ($k = 0, 1, 2$) along with geometric checks enabled. Its performance is controlled by a so-called confidence parameter $\alpha \in [0, 1]$. We sweep over the values of this parameter and compute the precision and recall for each α, which are shown in Fig. 7.

For purposes of comparison, we carry out the same geometric verification step in DBoW (Fig. 7) and in the proposed method (Fig. 8): feature extraction, matching and fitting a fundamental matrix between the matched features. If sufficient support is not found for the fundamental matrix, the proposed hypothesis is rejected. This geometric verification is carried out on the full resolution images.

As seen from Figs. 8 and 7, the best precision-recall curve of our method competes with that of the DBoW in terms of precision; moreover, the proposed algorithm is more conservative and operates with *lower* recalls. This low recall is a consequence of requiring the match to be globally unique in order to be considered a loop closure. Overall, the proposed approach achieves competitive trade-off between precision and recall. We consider the results promising for the proposed methodology to be applied to other problems such as place categorization and the selection of visually distinct images for life-long topological mapping.

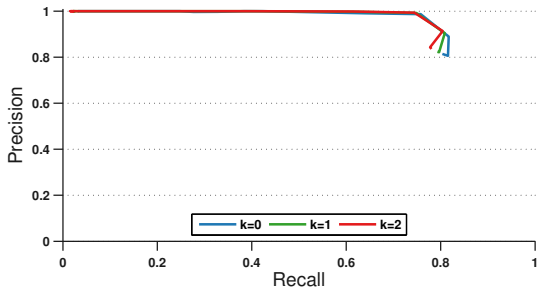


Fig. 7: Precision and recall curves of DBoW [19] for the Bicocca dataset using the full-sized 320×240 images. In this plot, k denotes the values used for temporal consistency constraint.

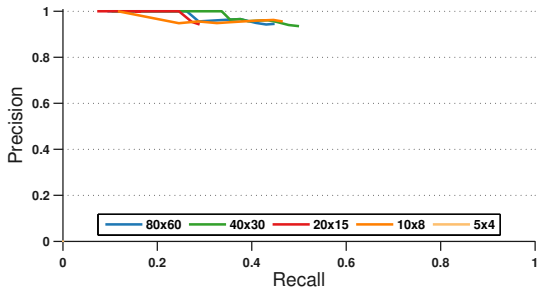


Fig. 8: Precision and recall curves corresponding to Fig.6 with an additional geometric verification step, same as the one used in DBoW.

VI. Conclusions and Future Work

While the problem of loop closure has been well studied in visual navigation, motivated by the sparse nature of the problem (i.e., only a small subset of past images actually close the loop with the current image), in this work, we have for the first time ever posed it as a sparse convex ℓ_1-minimization problem. The *globally optimal* solution to the formulated convex problem, by construction, is *sparse*, thus allowing efficient generation of loop-closing hypotheses. Furthermore, the proposed formulation enjoys a *flexible* representation of the basis used in the dictionary, with *no* restriction on how the images should be represented (e.g., what descriptors to use). Provided any type of image vectors that can be quantified with some metric to measure the similarity, the proposed formulation can be used for loop closing. Extensive experimental results have validated the effectiveness and efficiency of the proposed algorithm, using either the whole raw images as the simplest possible representation or the high-dimensional descriptors extracted from the entire images.

We currently use a single threshold τ to control the loop-closure hypotheses, which guarantees a globally unique hypothesis. However, in the case of multiple revisits to the same location, this hard thresholding would prevent detecting any loop closures and the revisits would be simply considered as perceptual aliasing, which is conservative but loses information. In the future, we will investigate different ways to address this issue. For example, as mentioned earlier, we can sum up the contributions of basis vectors if a loop has already been detected between them and thus ensure that multiple visits lead to more robust detection of loop closures. Nevertheless, this has not been a major issue in our tests; as shown in Fig. 2, the proposed algorithm is capable of detecting loops at different revisits. As briefly mentioned before, the number of basis vectors in the dictionary grows continuously and can prohibit the real-time performance for large-scale problems. To mitigate this issue, one possible way would be to update the dictionary dynamically by checking a novelty factor in terms of how well the current image can be explained by the existing dictionary, which is akin to adding "key frames" in visual SLAM.

Acknowledgments

This work was partially supported by the MINECO-FEDER project DPI2012-36070, by research grant BES-2010-033116, by travel grant EEBB-I-13-07010, by ONR grants N00014-10-1-0936, N00014-11-1-0688 and N00014-13-1-0588, and by NSF award IIS-1318392.

REFERENCES

[1] E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(12): 237–260, 1998.

[2] M. Asif. Primal dual pursuit: A homotopy based algorithm for the Dantzig selector. Master's thesis, Dept. of Electrical and Computer Engineering, Georgia Institute of Technology, 2008.

[3] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, pages 19–53, 2011.

[4] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417. Graz, Austria, May 7–13, 2006.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *European Conference on Computer Vision (ECCV)*, pages 778–792. Crete, Greece, September 5–11, 2010.

[7] C. J. Cannell and D. J. Stilwell. A comparison of two approaches for adaptive sampling of environmental processes using autonomous underwater vehicles. In *MTS/IEEE OCEANS*, pages 1514–1521, Washington, DC, December 19–23, 2005.

[8] F. Capezio, F. Mastrogiovanni, A. Sgorbissa, and R. Zaccaria. Robot-assisted surveillance in large environments. *Journal of Computing and Information Technology*, 17 (1):95–108, 2009.

[9] J. J. Casafranca, L. M. Paz, and P. Pinies. ℓ_1 Factor Graph SLAM: going beyond the ℓ_2 norm. In *Robust and Multimodal Inference in Factor Graphs Workshop,IEEE International Conference on Robots and Automation, (ICRA)*, Karlsruhe, Germany, 2013.

[10] J. J. Casafranca, L. M. Paz, and P. Pinies. A backend ℓ_1 norm based solution for factor graph SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 17–23, Tokyo, Japan, November 3–8, 2013.

[11] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang. Learning with ℓ_1-graph for image analysis. *IEEE Transactions on Image Processing*, 19(4):858–866, 2010.

[12] W. Churchill and P. Newman. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661, 2013.

[13] 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.

[14] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, San Diego, CA, June 20-26, 2005.

[15] D. L. Donoho. Compressed sensing. *IEEE Transactions on Signal Processing*, 52(4):1289–1306, 2006.

[16] D. L. Donoho and Y. Tsaig. Fast solution of ℓ_1-minimization problems when the solution may be sparse. Technical report, Dept. of Statistics, Stanford University, 2006.

[17] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[18] M. Elad, M. Figueiredo, and Y. Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6):972–982, 2010.

[19] D. Galvez-Lopez and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[20] Y. Latif, C. Cadena, and J. Neira. Robust loop closing over time for pose graph SLAM. *The International Journal of Robotics Research*, 32(14):1611–1626, 2013.

[21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[22] D. M. Malioutov, M Cetin, and A. S. Willsky. Homotopy continuation for sparse signal representation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[23] M. Milford. Vision-based place recognition: how low can you go? *The International Journal of Robotics Research*, 32(7):766–789, 2013.

[24] M. Milford and G. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1643–1649, St. Paul, MN, May 14–18, 2012.

[25] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[26] R. Paul and P. Newman. Self-help: Seeking out perplexing images for ever improving topological mapping. *The International Journal of Robotics Research*, 32(14): 1742–1766, 2013.

[27] RAWSEEDS. Robotics advancement through Webpublishing of sensorial and elaborated extensive data sets (project FP6-IST-045144), 2009.

[28] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1508–1515, Beijing, China, October 17-20, 2005.

[29] 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.

[30] H. Sugiyama, T. Tsujioka, and M. Murata. Collaborative movement of rescue robots for reliable and effective networking in disaster area. In *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, San Jose, CA, December 19–21, 2005.