Visual Loop Closing using multi-resolution SIFT Grids in Metric-Topological SLAM

Vivek Pradeep, Gerard Medioni, James Weiland
University of Southern California, Los Angeles, CA, USA
{vivekpra, medioni, jweiland}@usc.edu

Abstract

We present an image-based Simultaneous Localization and Mapping (SLAM) framework with online, appearance-only loop closing. We adopt a layered approach with metric maps over small areas at the local level and a global, graph-based abstract topological framework to build consistent maps over large distances. Rao-Blackwellised particle filtering and sparse bundle adjustment are efficiently coupled with a stereo-vision based odometry module to construct conditionally independent 'submaps' using SIFT features. By extracting keyframes from these submaps, a multi-resolution dictionary of distinct features is built online to learn a generative model of appearance and perform loop-closure. Creating such a dictionary also enables the system to distinguish between similar regions during loop closure without requiring any offline training, as has been described in other approaches. Furthermore, instead of occupancy or grid maps, we build 3D reconstructions of the world - a model we plan to use as input to a scene interpretation module for providing navigational cues to the visually impaired. We demonstrate the robustness of our SLAM system with indoor and outdoor experiments for full 6 degrees of freedom motion using only a stereo-camera in-hand, running at 1 Hz on a standard PC.

1. Introduction

An efficient, scalable and optimal solution to the Simultaneous Localization and Mapping (SLAM) problem has been the focus of much research in the robotics community since the seminal works of Smith and Cheeseman [22] and Durrant-Whyte [10]. The availability of inexpensive, light-weight cameras together with greater processing power and development of robust solutions to the feature correspondence problem (such as SIFT, [12]) and visual odometry [19], [23] has also attracted the computer vision community to this area of research. This framework has also found applications in augmented reality [9] and wearable devices. We are particularly interested in employing visual SLAM on a head-mounted stereo-camera for developing a mobility aid for the visually impaired[1]. Way-finding cues based on the generated 3D maps and current location can assist in both indoor and outdoor navigation. The work presented in [20] produces excellent results by collecting video streams along with GPS and inertia measurements. However, our system does not employ any additional hardware to maintain wearability. While monocular SLAM approaches such as MonoSLAM [6] and Hierarchial Visual SLAM [4] have been shown to be feasible in different kinds of environments, we pursue a stereo-based approach because it offers scale observability. It should be noted, however, that the proposed application does not restrict the applicability of our system in any way. Here, we only present key ideas and results relevant to the generic 6DOF visual SLAM problem–implementations of which can be be seamlessly ported onto mobile robots with stereo heads, for instance.

We adopt a metric-topological approach for medium to large scale SLAM, building upon [21] as the backbone for our submap construction. In particular, we extract SIFT features and use corresponding stereo data to implement a Rao-Blackwellised particle filter (RBPF) [16] for consistent local maps. Our contribution to this approach is the introduction of sparse bundle adjustment [8] over selected key-frames in the current submap node before initializing a new node in the topological map. Furthermore, we introduce a novel visual loop closure detection scheme by leveraging the previously extracted keyframes to build a generative model of appearance over submaps. In contrast to earlier approaches, our method eliminates ambiguities due to repetitive features in different regions while keeping the time complexity linear in the number of submaps (a similarity matrix approach discussed in [17] was cubic in the number of features). In this regard, the work of Cummins and Newman [5] is the most similar to our idea, where they build a dictionary of feature words in a different environment offline and then use a Chow-Liu tree to ap-

[1]This work was supported in part by the National Science Foundation under Grant No. EEC-0310723

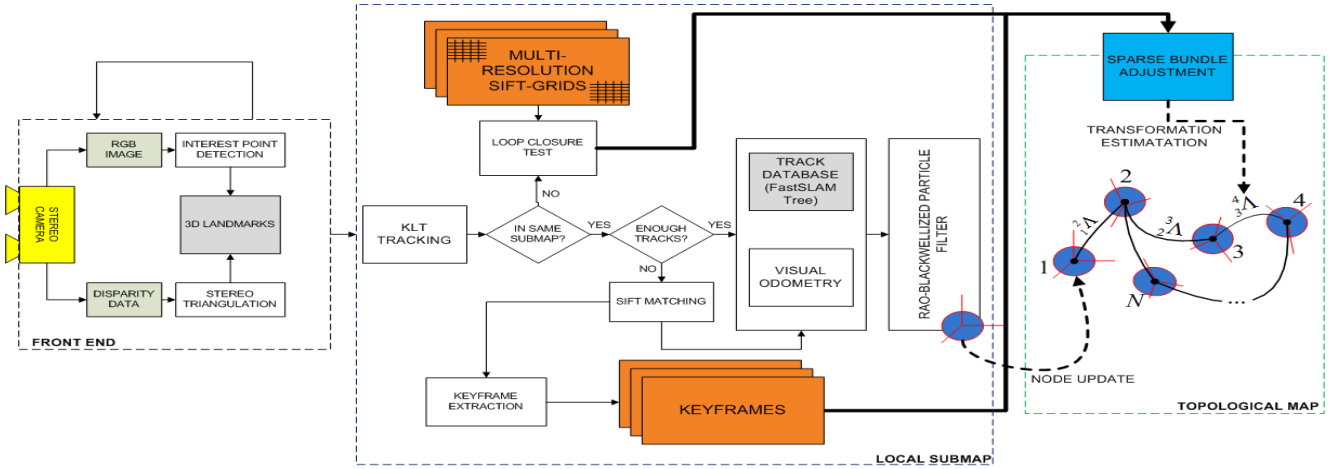

Figure 1. Overview of our landmark-based, visual SLAM framework. The basic structure at the local submap level is a Rao-Blackwellised particle filter . At the global level, we maintain a topological representation of the environment, with nodes corresponding to submaps and edges indicating the transformations between the local coordinate systems. Sparse bundle adjustment is used to smooth the camera trajectories within each submap and our proposed SIFT grids framework detects loop closures.

proximate joint densities over co-occuring words for loop-closure. However, our approach requires no such offline training and the words we consider come from the same environment that is being traversed. The rest of this paper is organized as follows. In the next section, we describe our basic metric-topological SLAM framework with sparse bundle adjustment. Our novel multi-resolution, SIFT grids based visual loop closure scheme is presented in Section 3 with experimental results in indoor and outdoor sequences in Section 4. We conclude with directions for future work in Section 5.

2. Metric–Topological SLAM

The SIFT landmark-based SLAM approach proposed here has two levels of environment representation: local, metric and global, topological. Different variations of such a setup can be found in [25], [7] and [3]. This scheme has two main advantages: by computing metric maps over small areas, the map size can be kept bounded and secondly, loop closure can be efficiently handled by simply reusing a previously computed submap. An overview of the system is presented in Figure 1.

2.1. Local Submap Level

At time $t-1$, let s_{t-1} represent the six dimensional camera pose (three coordinates for Euclidean position and three for the Euler angles, yaw, pitch and roll) and, m_{t-1} represent the associated map of 3D landmarks with reference to the local coordinate system $\mathcal{L}$. In the next time-step, the camera undergoes motion u_t given by a rotation matrix $\mathcal{R}$ and translation vector $\mathcal{T}$ to assume a new pose s_t, simultaneously making new observations z_t. This process can be modeled as a dynamic Bayes network with $x_t = \{s_t, m_t\}$ representing the system state and conditioned on all the observations and odometry gathered until now, i.e. z^t and u^t. In the standard RBPF formulation, an approximate but highly efficient solution can be obtained by the following factorization:

$$p(s^t, m_t \mid z^t, u^t) \approx p(s^t \mid z^t, u^t) \prod_i p(m_t(i) \mid s^t, z^t, u^t) \tag{1}$$

Here, s^t denotes the camera trajectory until time t and $m_t(i)$ is the i–th landmark in the map, represented by a normal distribution $\sim N(\mu_t^i, \Sigma_t^i)$. In our implementation, observations z_t are given by a collection of 'interest points' extracted from the left image of the stereo head coupled with their 3D position relative to the camera using stereo triangulation. The covariances of these landmarks are initialized from knowledge of the camera parameters and error modeling [14]. To estimate the motion, u_t from the previous time step, we compute the rigid transformation between Ncorresponding points in the current and preceding point-clouds C_t and C_{t-1} by minimizing the following objective function

$$\min_{\mathcal{R},\mathcal{T}} E(\mathcal{R}, \mathcal{T}) = \sum_{i=1}^{N} v_i^T S_i^{-1} v_i \tag{2}$$

with $v_i = \mathcal{C}_t^i - \mathcal{R}\mathcal{C}_{t-1}^i - \mathcal{T}$ and $S_i = \mathcal{R}\Sigma_{t-1}^i\mathcal{R}^T + \Sigma_t^i$. This function takes the errors in depth reconstruction into account and has a closed form solution using quaternions as illustrated in [24]. A preemptive RANSAC [18] scheme for minimizing this function gives robust results quickly. Given an estimate of the camera motion and observations, the RBPF in Equation 1 is implemented by propagating

camera pose hypotheses, performing EKF updates on landmarks and resampling.

Data association or establishing correspondences is accomplished by a combination of fast kd-tree [1] SIFT matching and KLT [13] tracking. We initially extract features with 128-dimensional descriptors but only perform KLT tracking (each track inherits the SIFT descriptor from the parent) from hereon until the number of tracks diminishes below a certain threshold. At this stage, we once again extract SIFT features for the current frame and continue this process. Under prolonged smooth motion, KLT tracks last longer and extracting SIFT features in this scenario would be a waste of computational resources. On the other hand, more rapid motion leads to frequent track failures and new SIFT matches provide more information about the same landmark for it to be identified from different viewpoints. This simple observation is exploited in building keyframes for a particular submap, which are used for sparse bundle adjustment and visual loop closing. We maintain a hash-table database of landmarks indexed by unique integer IDs for implementing a reference counted binary tree [15], [21] for quickly updating the per-particle maps during the resampling stage. A single index can point to similar looking but spatially distinct landmarks and allows multiple-data hypothesis during particle filtering. The map for each particle is simply an array of these indices.

2.2. Global, topological representation

A local submap i with coordinate frame $\mathcal{L}_i$, built using the above methodology in the time interval $t-\tau$ to t, encapsulates M samples of the camera pose trajectory ${}^{\mathcal{L}_i}s_k^{[t-\tau:t]}, k = \{1, 2, ...M\}$ and per-particle maps ${}^{\mathcal{L}_i}m_{s_k}$ learnt in the interval τ. Suppose, at this instance, a new submap j having coordinate frame $\mathcal{L}_j$ is to be initialized. A transformation ${}^j_i\Lambda$ between the two coordinate systems is computed from ${}^{\mathcal{L}_i}s_t^*$, which is the most likely pose (the sample with the largest weight) at this time in submap i. In fact, this transformation is smoothed using sparse bundle adjustment, as discussed in the next section. An initial map for this new region is constructed by simply copying the landmarks in ${}^{\mathcal{L}_i}m_{s^*}$ corresponding to only the current observations z_t and transforming them into the new coordinate frame. It can be easily observed that adjacent submaps are conditionally independent due to these shared landmarks and coordinate transformation. A formal proof for this can be found in [7]. In general, this structure can be represented by an annotated graph

$$G = \langle \{{}^i\mathcal{M}\}_{i\in\Omega_t}, \{{}^b_a\Lambda\}_{a,b\in\Omega_t} \rangle \tag{3}$$

${}^i\mathcal{M}$ are the metric submaps , Ω_t is the set of computed submaps until time t and ${}^b_a\Lambda$ are the coordinate transformations between adjacent maps. This is very similar to [3], but unlike them, we do not define multiple-hypotheses on the space of topological maps.

2.3. Keyframes extraction and Sparse Bundle Adjustment

A keyframe is defined as a collection of feature observations $\{\mathrm{F}_i\}_{i=\{1,2,...,N\}} \in z_t$, extracted from an image frame, each accompanied by the following attribute list:$\langle (\ r, c), d[128], h, m, f, id \rangle$.

(r, c) is the two-tuple row and column number of the feature location in the image, $d[128]$ is the SIFT descriptor, h and m are the hit and miss counters while f is the corresponding frame number. h is incremented during tracking (KLT or SIFT matching) every time a match is found while m stores the number of instances the feature was not observed when it was expected to be in the field of view (this can be trivially determined after computing camera motion along with knowledge of the previous feature location). The id integer holds the hash key for indexing into the corresponding 3D landmark from the landmark database described in Section 2.1. To reiterate, keyframes are extracted and stored for each submap every time new SIFT features are extracted. Note that this process is not independent of the relevant submap as the referenced landmarks are obtained from the per-particle maps in the active RBPF.

Frame to frame visual odometry and tracking consider adjacent frames in isolation and do not take advantage of the fact that shared features across several frames can provide a smoother motion estimate. Ignoring such information can lead to incremental errors that adversely impact the global transformations ${}^b_a\Lambda$ between submaps. Bundle adjustment [8], popular in the Structure from Motion paradigm is able to cope with multi-view motion estimation. Most implementations of this method are carried out offline, are extremely costly and in an online SLAM application, have no practical utility. However, our metric-topological architecture together with keyframe extraction allows us to easily integrate this critical component.

Bundle adjustment is performed over all the K keyframes in the current submap ${}^i\mathcal{M}$ before exiting (i.e., just before creating a new submap node or loop closure). Specifically, the most likely camera trajectory until now, ${}^{\mathcal{L}_i}s^{t*}$ is selected and K camera poses corresponding to each keyframe are extracted out. Together with knowledge of the camera parameters, we form projection matrices $\{P_i\}_{i=\{1,2,...K\}}$. We also aggregate all the landmarks seen in more than one keyframe using the id indices to give us N 3D points X_j from the associated most likely map and their corresponding projections in each frame, x_{ij}. The 3D points were actually estimated by running independent EKFs during the RBPF stage and therefore, are accompanied by covariance matrices which can be projected into the image planes to give weights w_{ij}. We proceed to minimize

the reprojection errors using a weighted least-squares formulation

$$\min_{P_i, X_j} \sum_{ij} w_{ij} d(P_i X_j, x_{ij})^2 \tag{4}$$

with $d(x, y)$ representing the Euclidean distance between points x and y in an image. An example of how the sparsity involved in this non-linear minimization can be exploited for an efficient sparse bundle adjustment (SBA) is presented in [24] and [11].

Once SBA is completed, the most likely particle trajectory and the corresponding map are refined and used to compute the transformation ${}^j_i\Lambda$ and map for the new node.

3. Visual Loop Closing

We define beliefs in the form of a probability distribution over the space of abstract, topological maps. This distribution determines the active node in the annotated graph, i.e. it gives the probability of being in the same submap, an older one (loop closure) or whether a new node must be initialized. Hence, this framework also drives creation of edges between the graph nodes. As recommended in [17], we avoid reasoning about the topological position based on the nearest nodes to the current pose as this quantity can be in gross error. It is also intractable to perform SIFT matching against every frame until the current time. Instead, we present an appearance-based method that extracts discriminative 'words' for each submap and makes loop-closure decisions based on current observations. The complexity of this approach is linear in the number of submaps as we employ a generative model over each node. While this approach is similar to the 'bag of words' used in object recognition and recently, visual SLAM [5], our algorithm requires no offline training and therefore has the advantage of learning only those words that are relevant to the environment being traversed.

3.1. Bayesian Framework

Given a set of feature observations z^t until time t, our goal is to estimate the density over each submap ${}^i\mathcal{M}$ in Ω_t. This can be formulated as the following recursive Bayes estimation problem

$$p({}^i\mathcal{M}|z^t) = \frac{p(z_t|{}^i\mathcal{M})p({}^i\mathcal{M}|z^{t-1})}{p(z_t|z^{t-1})} \tag{5}$$

The denominator is simply a normalizing term and $p({}^i\mathcal{M}|z^{t-1})$ is the prior belief over location that can be manually initialized. Hence, our algorithm focuses on estimating the posterior over the current observations z_t for each submap. We check for loop-closure or new submap initialization only when SIFT features are available, i.e. during the stage when a keyframe is being extracted. Therefore, the observation z_t includes a set of M SIFT features $\{\mathcal{F}^1_j\}$ available from the image. If we have a precomputed list of N 'discriminative' features $\{\mathcal{F}^0_j\}$ which have *not* been detected, then these are also added to the observation set. The intuition behind this is that the absence of stable features which predominantly occur in some submaps also provides information about the locations not being currently visited–it helps in reinforcing the belief that those particular submaps are not responsible for the observations. This is necessary for distinguishing between similar looking regions with repetitive features. While the common features will assign equal beliefs over the regions, the absence of characteristic features will penalize submaps that are associated with them. This is illustrated in Figure 2.

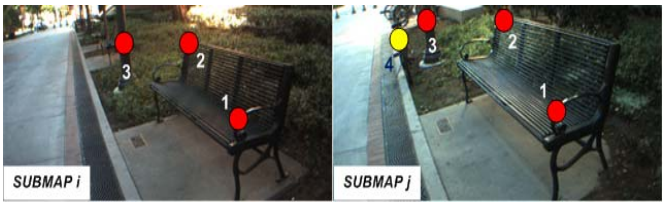

Figure 2. Feature discriminative power. Consider an observation consisting of only feature-words 1, 2 and 3. These words occur with equal probabilities in submaps i and j. Hence, both regions will be considered equal candidates for loop-closure (priors being the same). However, feature word 4 occurs consistently and exclusively in submap j. Since this highly discriminative and stable feature associated with j has not been observed, the confidence over submap j is reduced.

Let us define an indicator function $I(\mathcal{F}^p_j)$ which is 1 when the corresponding feature is present in the image and 0 otherwise. Thus, our observation set z_t consists of $M+N$ features given by

$$z_t = \{\mathcal{F}^p_j : I(\mathcal{F}^p_j) = p; p \in [0, 1]\} \tag{6}$$

Expanding the posterior,

$$p(z_t|{}^i\mathcal{M}) = p(\mathcal{F}^1_1, F^1_2, ...F^1_M, F^0_1, F^0_2, ..., F^0_N|{}^i\mathcal{M}) \tag{7}$$

Let us assume for now that each feature in the image is actually generated with some probability $\Upsilon_{\mathcal{F}^p_j}$ from a feature-word $\mathcal{W}_{\mathcal{F}^p_j}$. We show in the next section how such a dictionary of feature words can be constructed online. This description implies the following likelihoods

$$p(\mathcal{F}^p_j|\mathcal{W}_{\mathcal{F}^p_j} = 0) = 0 \ if \ I(\mathcal{F}^p_j) = 1 \tag{8}$$

$$p(\mathcal{F}^p_j|\mathcal{W}_{\mathcal{F}^p_j} = 1) = \Upsilon_{\mathcal{F}^p_j} \ if \ I(\mathcal{F}^p_j) = 1 \tag{9}$$

$$p(\mathcal{F}^p_j|\mathcal{W}_{\mathcal{F}^p_j} = 0) = 1 \ if \ I(\mathcal{F}^p_j) = 0 \tag{10}$$

$$p(\mathcal{F}^p_j|\mathcal{W}_{\mathcal{F}^p_j} = 1) = \delta_{\mathcal{F}^p_j} \ if \ I(\mathcal{F}^p_j) = 0 \tag{11}$$

Equation 8 and 10 state that in the absence of the generating word, the corresponding feature cannot be detected (no false positives). Equation 9 defines the likelihood of observing the feature if the word is present while equation 11 specifies the miss probability of the feature extractor. Equation 7 can be now rewritten as follows

$$p(z_t|^i\mathcal{M}) = \sum_{\omega} p(\{\mathcal{F}_j^1\},\{\mathcal{F}_j^0\}|\{\mathcal{W}_{\mathcal{F}_j^1}\},\{\mathcal{W}_{\mathcal{F}_j^0}\}) \\ p(\{\mathcal{W}_{\mathcal{F}_j^1}\},\{\mathcal{W}_{\mathcal{F}_j^0}\}|^i\mathcal{M}) \quad (12)$$

The curly braces have been applied for compactness and encapsulate the two sets of features and their generating words. The summation is to be evaluated over the entire space ω of possible word combinations, with each word taking two possible values –present or absent. Obviously, the size of this space grows exponentially with the number of words and can quickly become intractable to evaluate. It is reasonable to state that each word is conditionally independent of all other words, given the submap under consideration. This simplifies Equation 12 to

$$p(z_t|^i\mathcal{M}) = \sum_{\omega} p(\{\mathcal{F}_j^1\},\{\mathcal{F}_j^0\}|\{\mathcal{W}_{\mathcal{F}_j^1}\},\{\mathcal{W}_{\mathcal{F}_j^0}\}) \\ \prod_{j=1}^{M} p(\mathcal{W}_{\mathcal{F}_j^1}|^i\mathcal{M}) \prod_{j=1}^{N} p(\mathcal{W}_{\mathcal{F}_j^0}|^i\mathcal{M}) \quad (13)$$

Making a further assumption that feature detections are conditionally independent of all other words given the generating word and applying Equations 8 through 11,

$$p(z_t|^i\mathcal{M}) \approx \prod_{j=1}^{M} \Upsilon_{\mathcal{F}_j^1} p(\mathcal{W}_{\mathcal{F}_j^1}=1|^i\mathcal{M}) \\ [\prod_{j=1}^{N} p(\mathcal{W}_{\mathcal{F}_j^0}=0|^i\mathcal{M}) + \prod_{j=1}^{N} \delta_{\mathcal{F}_j^0} p(\mathcal{W}_{\mathcal{F}_j^0}=1|^i\mathcal{M})] \quad (14)$$

Note that the above factorization is only an approximation as we have not considered all the product terms while expanding the summation. Intuitively, the first term in Equation 14 favors those submaps in which the words corresponding to the observed features, $\mathcal{W}_{\mathcal{F}_j^1}$ co-occur. As a second level of support, the remaining terms take into account those discriminatory words whose features were not observed. Submaps with high probabilities for the absence of these words explain better why the corresponding features were not detected and get awarded. The $\delta_{\mathcal{F}_j^0}$ term takes care of unstable features and if equal to 1 (i.e, the miss probability is high), offers no advantage in this regard. In the following sections we describe how to construct the dictionary and compute discriminatory words from it.

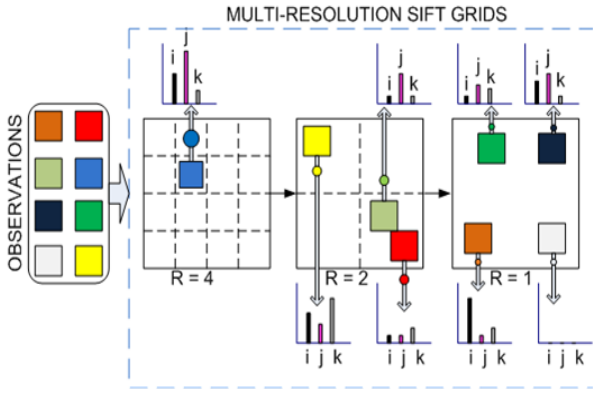

Figure 3. Schematic overview of the loop closure framework. Consider a set of 8 two-dimensional features (in different colors) and three grids of varying resolution. Associated probabilities for word occurrences in three submaps i, j and k are indicated with the corresponding word found in the grid. The radius of the circle along with arrow indicates the weight $\Upsilon_{\mathcal{F}_p^j}$ assigned to the observation.

3.2. Mutli-resolution SIFT grids

In the standard bag-of-words approach, clusters of features are assigned a specific word after an offline training step. For a SLAM algorithm that is supposed to run in varying environments, however, such training on an unrelated image set might not build a relevant vocabulary– for instance, a dictionary built using an outdoor image sequence will not necessarily encode sufficient semantics to distinguish between features collected in an indoor setting. Our approach eliminates this kind of subjectiveness by using an online approach dependent only on the environment being traversed.

We start by building quantized, 128-dimensional 'SIFT grids' that store the number of hits and misses in each bin for each submap. These grids are global and shared by all the submaps. A SIFT grid is updated when a keyframe has been extracted, using the h and m counters for the features to carry out the update. A SIFT grid is characterized by its resolution, $\mathbb{R}$ that determines a fixed quantization along each dimension. Since SIFT descriptors are essentially 128 numbers, each in the range $[0-255]$, setting $\mathbb{R}=32$, for instance, would give 8 bins along each dimension. In other words, the total number of bins can be $(\frac{256}{\mathbb{R}})^{128}$. However, not all bins are expected to be present and we use a hash-table to implement the grids. The key is simply generated by quantizing the descriptor according to $\mathbb{R}$ and updating the relevant grid element. These grid bins are our words $\mathcal{W}_{\mathcal{F}_j^p}$ for each feature. Note that depending upon the quantization level, different features can have the same word but different words cannot include the same feature. A schematic depiction of this procedure is provided in Figure 3.

The value of $\mathbb{R}$ is difficult to arrive at. For loop-closing, it is necessary to be able to establish correspondences between features and words. If $\mathbb{R}$ is set to a small value, minor variations along any one dimension of the descriptor will result in querying for a completely different word than what was intended–the query word might not have even been instantiated in the grid. On the other hand, a large value will collapse all the features into just a few bins and hamper the task of discrimination.

Our solution to this problem is the creation of several SIFT grids with values of $\mathbb{R}$ ranging from small to big and assigning detector probabilities $\Upsilon_{\mathcal{F}_j^p}$ based on the respective resolutions. Each time a keyframe is extracted, all the grids are updated by hashing the descriptors accordingly. Since each grid is a hash-table, this operation can be done very efficiently. In each grid, all possible features that can be generated by a word are considered to be equally likely in the space bounded by the resolution of the bin, i.e.

$$\Upsilon_{\mathcal{F}_j^p} \propto \frac{1}{\mathbb{R}} \tag{15}$$

When checking for loop-closure with a feature $\mathcal{F}_j^p$, a search is made by looking for the existence of the corresponding hash-entry starting with the finest resolution grid to the coarsest. Search is stopped once an entry is found and $\Upsilon^i_{\mathcal{F}_j^p}$ assigned according to equation 15. We then extract the hit and miss counters stored for each submap in this bin to obtain the probabilities for the word given a submap (normalized so that the the results are not biased on the duration of each submap). In a loop-closure scenario, we would expect several high-resolution words to be found (created by their first occurrence) and some low resolution words (corresponding to new features encountered this time). The high resolution words would get higher weights and the probability over each submap can be computed from the counters. In the case of no-loop closure (i.e., a new region), all the new feature observations would correspond to low resolution grids and therefore, the probability over any previous submap will be low. In our implementation, we employ five grids with $\mathbb{R}$ values $16, 32, 64, 128, 256$.

3.3. Learning Discriminative Words

As outlined in Section 3.1., we also consider words that are predominant in only some submaps. These are discriminative words and if the corresponding features are not detected, they provide strong cues about which submaps are not being visited. We only consider the finest resolution grid, $\mathbb{R} = 16$ to extract such words. For any given word $\mathcal{W}_{\mathcal{F}_j^p}$, we have the probability of its occurrence in each submap. Let us assume that we have $\mathbb{N}$ total submaps available at time t in $\Omega_{t.}$ Given the submap occurrence probabilities, we can define a random event X as the word being present in one of the these $\mathbb{N}$ possible locations, i.e. $p(X =^i \mathcal{M})$. If the word could occur with equal likelihood in all submaps, the associated entropy or uncertainty would be maximum, i.e,

$$H(X)^{max} = -\log(\frac{1}{\mathbb{N}}) \tag{16}$$

However, the actual entropy is,

$$H(X) = -\sum_{i=1}^{\mathbb{N}} p(X =^i \mathcal{M}) \log p(X =^i \mathcal{M}) \tag{17}$$

The entropy is zero if the word occurs with a probability of 1 in any submap (i.e., there is no uncertainty regarding which submap the word occurs in). Such a word is guaranteed to be highly discriminative. However, the maximum possible entropy grows with the number of submaps and to what is really relevant is the decrease in entropy. We compute a normalized version of this quantity called 'distinctiveness' for each word

$$D(\mathcal{W}_{\mathcal{F}_j^p}) = \frac{H(X)^{max} - H(X)}{H(X)^{max}} \tag{18}$$

Thus, when checking for loop closure, we uncover all the words with $D(...)$ above a certain threshold (0.95 in our implementation) and filter out those whose corresponding features have been observed. The left overs are used to compile the set $\{\mathcal{F}_j^0\}$. The stability $\delta_{\mathcal{F}_j^0}$ is computed from the miss counter value.

3.4. When to check for loop-closure

Finally, we outline a simple procedure for performing a check for loop-closure or new submap creation. We simply measure the number of KLT tracks that are terminated during keyframe extraction. If several KLT tracks are propagated further after SIFT matching, then the current submap is carried forward. If not, we apply our loop-closure test as presented in the previous subsections. The probability of a new location or an old submap is easily ascertained by the resolution of the grid returning the most words. This is a heuristic approach and requires a threshold on the number of tracks that will be allowed to terminate before a test is performed. Other techniques such as spectral bisection [2] or graph-cuts could also be employed.

4. Experiments

We tested our system for full 6DOF motion using a handheld stereo camera (Bumblebee manufactured by Point Grey Research) in indoor and outdoor environments. All the processing was done offline on a 3.39 GHz, Pentium dual core PC equipped with 3.00 GB of RAM. We assumed no knowledge about the environment (for instance, the ground plane). Furthermore, we have not applied any batch-based

Figure 4. Indoor experiment result. The non-textured 3D map is projected onto the XY plane. Lower-right shows the floor plan of the two rooms with the camera trajectory (red) exhibiting a loop. The highlighted section on the map corresponds to the images (top-right) captured during the beginning and end of loop execution. Loop closure was detected with a probability of 0.9984.

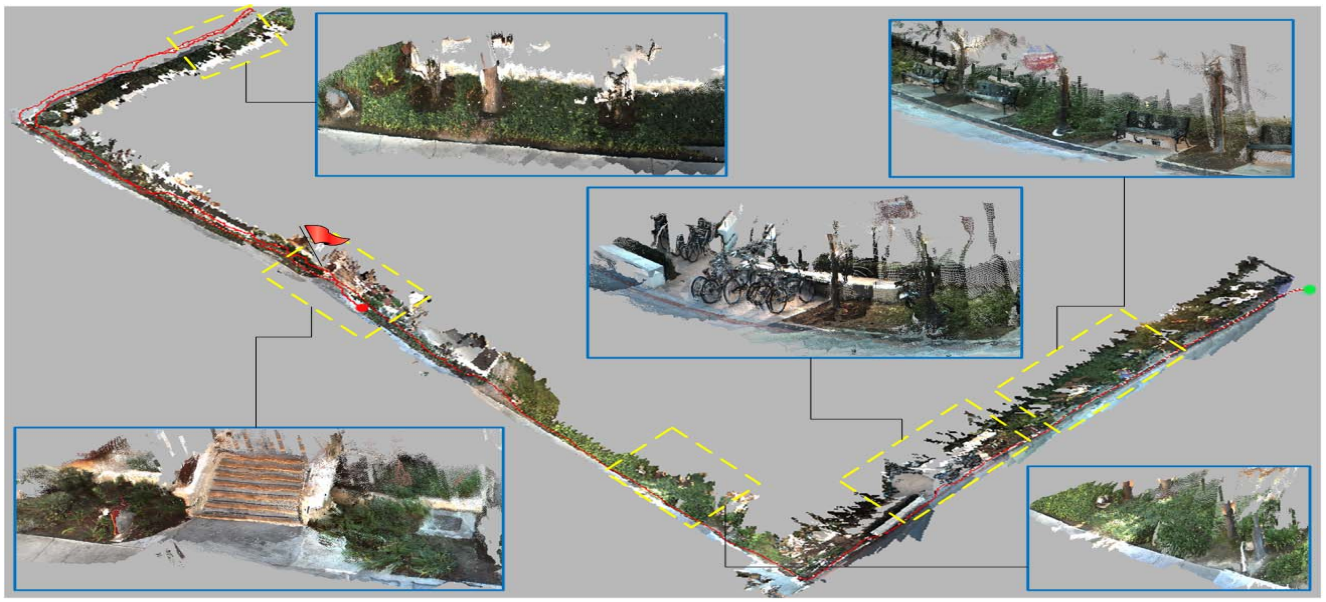

Figure 5. Top view of the reconstructed 3D map with camera trajectory (red line) for the outdoor test environment. Selected sections have been magnified for visualization. A video walk through for this experiment is part of the supplementary material submitted along with this paper. The flag indicates where loop closure was found and corresponding images are displayed in Figure 6. The total trajectory length was 232 meters.

smoothing algorithms to the results presented here, except for sparse bundle adjustment which runs online during execution. The average processing time was 1 frame/sec.

Figure 4 displays an indoor experiment result, with a camera trajectory length of approximately 30 meters (2206 frames). Figure 5 shows the result for a much longer outdoor sequence with a path length of about 232 meters (8000 frames). We typically had between 30 to 150 feature matches between adjacent frames. Both these experiments demonstrate loop closure using our SIFT grids-based approach. The outdoor data-set is characterized by several similar looking regions and in Figure 6, we indicate the probabilities estimated by the algorithm for region similarity. This demonstrates our proposed method's ability to

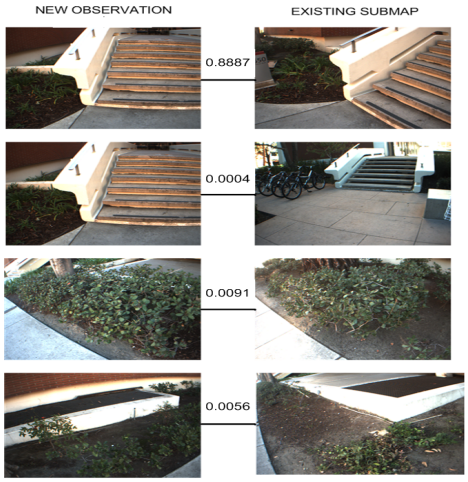

Figure 6. Similarity probabilities between images in different submaps. Top row corresponds to the detected loop in Figure 5.

discriminate between submaps with repetitive features.

5. Conclusion and Future Work

We have presented a vision-based, metric topological SLAM algorithm with impressive performance over large distances and in the presence of loops. Keyframes extraction using SIFT features enables to perform sparse bundle adjustment and therefore, compute smooth transformations between submaps. Furthermore, our visual loop closure algorithm is able to distinguish between similar regions with linear complexity in the number of submaps. This method requires no offline training. For improved performance, we are working on GPU implementations for some of the components of our system. We are also interested in implementing multiple hypotheses over submap topologies. The final objective is to provide way-finding cues to the visually impaired using the output of our system. We are designing initial human experiments.

References

[1] J. S. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. *CVPR*, pages 1000–1006, 1997.

[2] J. L. Blanco, J. Gonzalez, and J. A. Fernandez-Madrigal. Consistent Observation Grouping for Generating Metric-Topological Maps that Improves Robot Localization. *ICRA*, pages 818–823, 2006.

[3] J.-L. Blanco, J.-A. F. Madrigal, and J. Gonzalez. Toward a Unified Bayesian Approach to Hybrid Metric-Topological SLAM. *IEEE Trans. on Robotics*, 2008.

[4] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardos. Mapping Large Loops with a Single Hand-held Camera. *Robotics:Science and Systems*, 2007.

[5] M. Cummins and P. Newman. Probabilistic Appearance Based Navigation and Loop Closing. *ICRA*, pages 2042–2048, 2007.

[6] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam:Real-time single camera SLAM. *IEEE PAMI*, 29(6):1052–1067, 2007.

[7] C. Estrada, J. Neira, and J. D. Tardos. Hierarchial SLAM: Real-Time Accurate Mapping of Large Environments. *IEEE Trans. on Robotics*, 21(4):588–596, 2005.

[8] R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. *Cambridge University Press*.

[9] G. Klein and D. W. Murray. Improving the Agility of Keyframe-Based SLAM. *ECCV*, pages 802–815, 2008.

[10] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. *IROS*, 3:1442–1447, 1991.

[11] M. Lourakis and A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the Levenberg-Marquardt algorithm. *Institute of Computer Science - FORTH, Heraklion, Crete, Greece.*

[12] D. Lowe. Object recognition from local scale-invariant features. *ICCV*, pages 1150–1157, 1999.

[13] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. *IJCAI*, pages 674–679, 1981.

[14] L. Matthies and S. Shafer. Error modeling in stereo navigation. *IEEE JRA*, 3(3):239–248, 1987.

[15] M. Montemerlo, S. Thrun, D. Koller, and B. Weigbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *IJCAI*, pages 1151–1156, 2003.

[16] K. Murphy. Bayesian map learning in dynamic environments. *NIPS*, pages 1015–1021, 1999.

[17] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using Visual Appearance and Laser Ranging. *ICRA*, pages 1180–1187, 2006.

[18] D. Nister. Preemptive RANSAC for Live Structure and Motion Estimation. *ICCV*, 1:199–206, 2003.

[19] D. Nister, O. Naroditsky, and J. Bergen. Visual Odometry for Ground Vehicle Applications. *J. Field Robotics*, 23(1), 2006.

[20] M. Pollefeys, D. Nister, et al. Detailed Real-Time Urban 3D Reconstruction from Video. *IJCV*, 2007.

[21] R. Sim, P. Elinas, and J. J. Little. A study of Rao-Blackwellised Particle Filter for Efficient and Accurate Vision-Based SLAM. *IJCV*, 74(3):303–318, 2007.

[22] R. C. Smith, M. Self, and P. Cheeseman. On the Representation and Estimation of Spatial Uncertainty. *Proc. of the Annual Conf. on Uncertainty in AI*, pages 435–461, 1986.

[23] N. Sunderhauf and P. Protzel. Towards Using Sparse Bundle Adjustment for Robust Stereo Odometry in Outdoor Terrain. *TAROS06*, pages 206–213, 2006.

[24] P. Zhang, J. Gu, and E. E. Milios. Registration uncertainty for robot self-localization in 3D. *The second Canadian Conf. on Computer and Robot Vision*, pages 490–497, 2005.

[25] Z. Zivkovic, B. Bakker, and B. Krose. Hierarchial Map Building using Visual Landmarks and Geometric Constraints. *IROS*, pages 2480–2485, 2005.