2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

# Hierarchical Visual Mapping with Omnidirectional Images

Hemanth Korrapati, Ferit Uzer
Université Blaise Pascal, Institut Pascal,
Aubière, F-63171.
firstname.lastname@univ-bpclermont.fr

Youcef Mezouar
Institut Française Mécanique Avancée (IFMA),
Aubière, F-63171.
firstname.lastname@ifma.fr

***Abstract*— A topological mapping framework designed for omnidirectional images is presented. Omnidirectional images acquired by the robot are organized as places which are represented as nodes in the topological graph/map. Places are regions in the environment over which the global scene appearance of all acquired images is consistent. A hierarchical loop closure algorithm is proposed which quickly sifts through the places to retrieve the most similar places and another level of thorough similarity analysis is performed over the images belonging to the retrieved places. An Image similarity metric based on spatial shift of local image features across omnidirectional/panoramic image pairs is proposed. Newly proposed VLAD (Vector of Locally Aggregated Descriptors) descriptors have been used for loop closure at place and image levels. Accuracy and efficiency of our system are corroborated with experimental results on three publicly available datasets. It is shown that our approach achieves good loop closure recall rates even without using epi-polar geometry verification common among many other approaches.**

## I. INTRODUCTION

Loop closure is a problem of knowing if the robot is revisiting an already visited area of the environment and plays a pivotal role in accurate map construction. Many powerful vision based approaches have been proposed recently [1], [2], [3], [4] to address this problem efficiently. Although some of these approaches use omnidirectional cameras for experimentation, they do not explicitly take advantage of the rich 360 degree image representation. With a 360 degree field of view, omnidirectional images do not suffer from objects going out of the field of view as the robot moves or rotates. In other words, even during robot motion, omnidirectional image appearance remains constant for a longer time compared to the pinhole camera. This observation motivates us towards the notion of $places$ - regions of an environment over which the acquired images' appearance remains similar. Another advantage of omnidirectional cameras is that one needs to traverse each path only once (irrespective of the direction of motion) in order to map the environment as opposed to the traditional cameras which demand at least two passes through a path each in opposite directions.

Representing the environment as places has many advantages in place categorization, providing stronger constraints for pose-graph SLAM [5], [6], [7], semantic labelling [8], and topo-metric SLAM [9]. However, scope of the present approach is only to use place representation for accurate and efficient map building. Since the maps can be represented by fewer nodes, one might as well refer to them as sparse topological maps.

A hierarchical loop closure algorithm is proposed which when given a query image, firstly the most similar places/nodes in the map are retrieved. Then, an exhaustive similarity analysis is performed on the member images of the retrieved places. The first phase of loop closure constituting most similar nodes retrieval is achieved using VLAD (Vector of Locally Aggregated Descriptors) descriptor which has been proposed in [10] for web-scale image search. This process happens very fast and boils down the whole map to a few important places. The second phase of loop closure that aims to find the most similar images is carried out using a novel spatial similarity measure for omnidirectional images using visual words obtained by quantized local image features (like SIFT, SURF, etc). This metric is obtained by measuring shifts of matched features and can be used as a soft geometric similarity measure which can be applied to hundreds of images per time step and hence a good alternative to RANSAC based epipolar geometry verification. Also this technique does not need camera calibration and can offer in plug-and-play type functionality to any omnidirectional/panoramic camera images.

A secondary contribution lies in introducing the VLAD descriptor to the robotics community which has never been done to the best of our knowledge. Different aspects of our algorithm are experimentally evaluated on the NewCollege [11] public dataset and two of our own multi-sensor datasets.

The remainder of this paper is organized as follows. Section II discusses the related work, section III briefly introduces the VLAD descriptor construction and section IV discusses the map formulation followed by node and image similarity analyses. Finally our experiments are detailed in section V.

## II. RELATED WORK

Several approaches [12], [13], [8], [14], [15], [16] have used place representation of the environment similar to that of ours. Vatani et al. [12] proposed a sparse topological mapping algorithm in which places are recognized by optical flow. Statistical information over convex hulls formed over the image features are used to detect place/scene changes in [13], while bayesian change point detection over spatial pyramid histograms is used in [8]. A normalized graph cuts based space segmentation has been used in [14] for topological mapping in indoor environments. In [17], SIFT feature matching scores have been used in building sparse maps for indoor environments. Incremental spectral clustering has been used by [18] to form nodes of a sparse topological map

which was used to localize the robot in different seasons. A bag of words based sparse topological mapping has been proposed and evaluated in [15]. GIST features [19] have been used in [16] to construct a sparse topological map. However GIST features are known for their low degree of invariance [10].

Most of the existing vision based loop closure techniques make use of the bag-of-words model [3], [1], [2], [20], [4] but differ in the ways they detect loop closures. The power of inverted files has been used in efficient loop closure in [3]. Relations between visual words are modeled using a generative model in [1] and [2]. Loop closures using word histograms is used in [20] and a basically similar but much more efficient approach is presented in [4] using BRIEF descriptors. 3D range information is used in conjunction with a camera in [21] for navigation and map building. The above discussed approaches assume a dense topological map where each image is treated as a node and focus more on the loop closure problem. Most of these approaches do not assume any specific camera model and propose generic approaches and also do not capture the geometric information encoded among the visual words of the images.

## III. VLAD Feature Descriptor

This section provides a brief overview of VLAD (Vector of Locally Aggregated Descriptors) descriptor [10] construction. As the name suggests, VLAD is a global image descriptor constructed from local image descriptors like SIFT [22] or SURF [23]. The basic intuition behind VLAD descriptors is to combine the quantization residues of the local feature descriptors into a single descriptor and use it as a global image descriptor.

Algorithm 1 describes VLAD computation using SURF descriptors as local image descriptors. The inputs for VLAD computation are image $I$, a bag of words quantizer $Q$ (codebook) of $k$ words learned on a training data, SURF descriptor length $l$ and a PCA (Principal Component Analysis) matrix $P$ which is also learned on training data. First, the SURF descriptors are extracted on image $I$ which are then quantized (lines 3-6). Subsequently, quantization residues are computed for each descriptor. Quantization residue is the vector difference between the feature descriptor and the centroid to which it is quantized to in vocabulary $Q$, and hence has the same dimensionality $l$ as the feature descriptor. Vector sum of quantization residues corresponding to all the features quantized to each centroid is computed and then represented as a column vector of matrix $d$. Finally, the $k$ column vectors of $d$ (sum of quantization residues) are augmented to form a full vlad descriptor $\mathbf{D_{vlad}}$ of dimensionality $k * l$ which can be quite huge. For example, in our implementation, a $128-$word vocabulary ($k$) and $64-$dimensional ($l$) SURF descriptors are used and the resulting full VLAD descriptor is $8192-$dimensional. Therefore, descriptor size is reduced using a PCA-projection (line 15). In the present application, PCA-projection has been used to compress the full VLAD descriptor to a $256-$dimensional descriptor. Hereafter in this article, whenever a reference is made to VLAD descriptor it actually means PCA compressed VLAD descriptor.

The quantizer $Q$ and the PCA matrix $P$ are the parameters which are learned on the training data. It has been suggested in [10] that very small vocabulary sizes like $k = 64$ to $k = 256$ are sufficient for attaining a good accuracy. A detailed description of the quantizer and PCA matrix learning is given in section V.

Since VLAD only depends on the continuous quantization residues, it can bypass the effects of hard quantization [24] to some extent.

**Algorithm 1** VLAD Descriptor Computation

1: **procedure** GET_VLAD($I$, $Q$, $l$, $P$)
2: ▷ $I$ - Image, $Q$ - Quantizer, $l$ - SURF descriptor dimension, $P$ - PCA matrix
3: $\mathbf{F_{surf}}$ =Extract_SURF($I$) ▷ Extracts SURF features
4: $n$ =Num($\mathbf{F_{surf}}$) ▷ Number of SURF features extracted.
5: $k$ =Vocabulary_Size($Q$)
6: $\mathbf{F_w}$ =Quantize($\mathbf{F_{surf}}$, $Q$) ▷ Quantize features into words.
7: $d = [O]_{k \times l}$ ▷ Initialize residue matrix with zeros.
8: **for** $i$ = 1 to $n$ **do** ▷ For each SURF feature
9: $c^i$ =Get_Centroid($Q$, ${\mathbf{F_w}}^i$) ▷ get centroid corresponding to the word.
10: ${d(\mathbf{F_w}}^i) = d({\mathbf{F_w}}^i) + {\mathbf{F_{surf}}}^i - c^i$
11: ▷ Accumulate quantization residue as columns of $d$.
12: **end for**
13: $\mathbf{D_{vlad}} = [d(1)^T | d(2)^T | .... | d(k)^T]_{1 \times (k*l)}$
14: ▷ VLAD descriptor computation by augmenting quantization residues.
15: $\mathbf{D_{pca-vlad}} = P \times {\mathbf{D_{vlad}}}^T$ ▷ PCA-projection to compress the descriptor length.
16: **end procedure**

## IV. Map Representation & Loop Closure

An image acquired at a time instant $t$ is represented by $I_t$ and the features extracted on it by $F_t = \{\mathbf{D_t}, \mathbf{Z_t}\}$. Where $\mathbf{D_t}$ is the VLAD descriptor and $\mathbf{Z_t}$ is the vector of bag of words quantized SURF features. The topological map at time $t$ is represented as a set of nodes/places $\mathbf{M}_t = \{N_1, N_2, ...\}$ and a graph $\mathbf{G_t}$ which encodes adjacency relations among nodes. Each node $N_i$ contains a set of member images and their features $\mathbf{I}^{N_i} = \{I_1^{N_i}, I_1^{F_i}, I_2^{N_i}, I_2^{F_i} ...\}$ and a representative feature $R^{N_i}$ which is the centroid of its member images' VLAD descriptors. $R^{N_i}$ is updated on addition of every new image to the node.

For each new image $I_t$, its similarity to all the reference images in the map $\mathbf{M}_t$ is evaluated hierarchically through node and image levels. First, the nodes which produce high similarities with $I_t$ are found and then, a thorough similarity with spatial constraints is evaluated on the images belonging to the highly similar nodes. These image similarity scores are used as likelihoods in a recursive bayesian filter similar to [3], [1], [2], [17]. The transition probabilities are uniformly distributed across the two neighboring images on both sides of the reference image and zero with respect to other images. The following subsections discuss in detail, the node construction and similarity evaluation mechanisms.

### A. Node Similarity

The aim of node similarity analysis is to search the graph for the most similar nodes to the given query image $I_t$. To simplify the notation, let us call the query image $I_q$. To obtain node similarities, we treat the set of nodes in the graph $\mathbf{G_{t-1}}$ as two disjoint parts such that $\mathbf{M}_{t-1} = \{\mathbf{N^R} \cup \{N^c\}\}$. Where $\mathbf{N^R}$ is called the reference nodes set which constitutes all the nodes in the map except the current place node $N^c$.

Given a query image $I_q$, there are three possibilities:

1) It is similar to an existing reference node(s).
2) It is not similar to any of the reference nodes but is similar to the current place/node.
3) It is neither similar to any reference node nor the current node and hence should belong to a new node.

The intuition behind separation of reference nodes and current node is that in an image sequence, a query image $I_q$ can be similar to the current place node $N^c$ in most cases. This can happen due to their temporal proximity which can often mean appearance similarity, leading to a temporally constant possibility of loop closure. Hence, a loop closure possibility with the reference node set is evaluated first.

Given the query VLAD descriptor $\mathbf{D_q}$ we evaluate its similarity to all the reference nodes in the graph using a gaussian kernel as follows:

$$node\_sim(\mathbf{D_q}, N_i) = 1 - \frac{g_\sigma(\mathbf{D_q}, N_i)}{\sum_{j=1}^{|\mathbf{N^R}|} g_\sigma(\mathbf{D_q}, N_j)} \quad (1)$$
$$\text{where : } g_\sigma(\mathbf{D_q}, N_i) = exp\left(\frac{-dist(\mathbf{D_q}, N_i)}{2\sigma^2}\right)$$

In Equation 1, all the computations involving nodes are performed using the node's centroid. For example $dist(\mathbf{D_q}, N_i)$ indicates the euclidean distance between $\mathbf{D_q}$ and the centroid of the node $N_i$. The computed similarity values range between 0 and 1 inclusively. Kernel width $\sigma$ will be discussed in section V.

A set of relevant nodes $\mathbf{N}^w$ whose similarities are greater than a threshold $T_s$ are selected as the best matches. The Relevant nodes' member images are selected for image similarity analysis which yields likelihoods used for loop closure posterior computation. A no-loop-closure event is recognized when none of the similarities rise above the threshold or when the posterior probabilities indicate a no-loop-closure event. In case of a no-loop-closure event, since the image does not belong to any of the places, we verify if it is addable to the current place node.

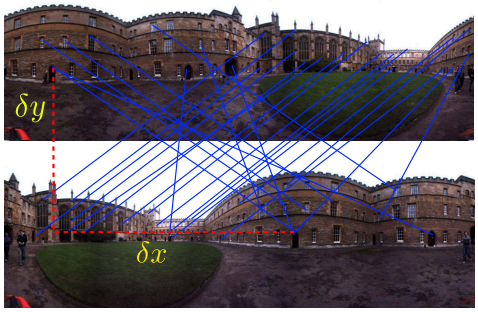
Stricter conditions must be satisfied for an image to be added to the current node. The current node $N_c$ is modeled as a hyper-sphere with center as the representative feature $R^{N_c}$ (centroid of the member images' VLAD features). A new feature can be added to the node only if the updated centroid is less than a distance of $r_n$ from all the member features as well as the new feature. The parameter $r_n$ is called node radius and ensures that all member features are tightly bound within this radius from the centroid. Finally, if the feature cannot be added to the current node a new node is formed with the feature itself being the centroid.

### B. Image Similarity

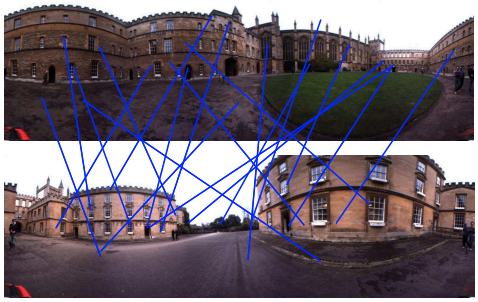
Given a set of relevant nodes $\mathbf{N}^w$ a reference image set $\mathbf{I}^R = \{I_{r1}, I_{r2}, ...\}$ is constructed by the union of relevant nodes' member images. Image similarities are computed by combining two similarity measures namely the VLAD descriptor similarity and the spatial similarity. VLAD descriptor similarity is computed as the euclidean distance between the query image VLAD descriptor $\mathbf{D_q}$ and the reference image descriptor.

The second similarity measure evaluates spatial similarity between two omnidirectional images (assumed to be unwrapped panoramas as the images in Figure 1). Let us consider two omnidirectional images acquired at approximately same location but with different heading directions ; and the robot is assumed to move in locally planar environments. Since the omnidirectional images have a circular field of view, distances between different objects in an image are well preserved even under a change in heading direction and a slight translation. In other words, the spatial structure of the objects(also applies to local image features) does not change with an in-place rotation of the camera. Hence if two images are from the same place, all objects in the first image should be shifted by similar amount to take their positions in the second image. A zero shift in object/feature coordinates indicates that the images are acquired in the same place and same heading direction, while a non-zero shift indicates same place with different heading. In case of a non match different objects will have different shifts. This situation is illustrated in Figure 1 from which, one can infer that the feature shifts of the true loop closure follow a converging pattern and those of the false loop closure look dispersed.

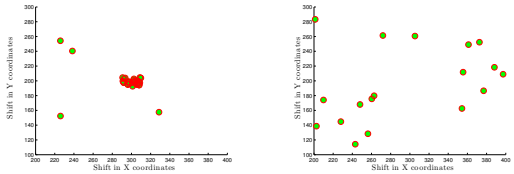
The major hurdle here is to mathematically discriminate true matches from false matches using the shift values while being robust to the outliers. Algorithm 2 details the feature shift analysis. The first procedure shows the structure of image similarity evaluation, which includes a call to the spatial similarity evaluation procedure. To compute spatial similarity (procedure 2), initially the feature shifts are accumulated into histograms (lines 10-17). There can be problems in accommodating features which are shifted beyond the right border of the image start appearing on the left and vice versa. To tackle this problem, image width is used in order to measure shifts in x coordinates only in one direction. Another problem is in case of multiple instances of the same visual word in one or both images. For this we have an efficient clustering based solution. However, to keep things simple, an approximation is made by considering the mean of the keypoints as a representative keypoint of all occurrences of the visual word. While adding shifts corresponding to the mean keypoints to the shift histogram, the corresponding bins are updated by the number of occurrences of the visual word. All the experimental results provided in this paper are generated by using this approximation.




(a) True Match



(b) False Match



(c) True Match Feature Shifts



(d) False Match Feature Shifts

Fig. 1: Feature Shift analysis of a true match and a false match. 1a shows features matched across a pair of images which are acquired in the same place. Matches are shown with blue lines. Shift in X and Y coordinates of a matched feature pair is demonstrated with a red dashed line. 1b shows illustrates a false match of a pair of images acquired at different places. 1c and 1d show the plots of matched feature shifts corresponding to the true and false match cases respectively.

Weighted mean of shifts is computed using the shift histograms (lines 22-31). Weighted mean is preferred in place of regular mean to acquire robustness towards outliers. Then the distances of all shifts to the weighted mean is computed and entered into a distance histogram with $L$ bins (lines 32-36). The final similarity score is computed using the histogram by assigning weights that are inversely proportional to the distance the bin corresponds to (lines 39, 40). This way, the first bin elements which contains all the shifts closest to the mean get multiplied by the highest weight and hence contribute the most to the similarity score. Similarly, the higher distance the bin corresponds to, the lower its contribution to the similarity. Note that the above discussed spatial similarity evaluation is only robust to minor translations and variable heading. However, it perfectly captures the definition of a loop closure and is clearly indicated by the experimental results. The main advantage of this approach which widely discriminates it from others is that it produces strong similarity scores even with a few feature matches between a pair of images acquired at the same location. Consequently, the loop closure recall rate is improved tremendously.

### C. Map Update

When the query image is similar to the current node or when a new node is to be formed, the query image is simply added to the corresponding node and the node centroid is updated.

However in the third case (mentioned in section IV-A), image similarity likelihoods are computed (Likelihood_Evaluation procedure in Algorithm 2), which in turn are used in posterior probability computation of loop closure. A posterior probability greater than $0.9$ is considered to be a degree of belief high enough to indicate a loop closure. In case of a loop closure, the image $I_t$ is added to the corresponding node only if the past $m$ images also formed loop closures with the node $N_i$ or one of its immediate neighbors.

## V. EXPERIMENTS

Experimental results are presented on three data sequences : the NewCollege [11], PAVIN and Cezeaux datasets. NewCollege dataset is a popular laser and vision dataset from Oxford. Panoramas from the LadyBug camera are used for our experiments. The dataset also contains GPS readings but they are only partly stable and hence are not very useful for ground truth construction. Hence loop closure accuracies for precision-recall evaluation were determined by manual inspection wherever necessary. PAVIN and Cezeaux sequences are part of the Institut Pascal multi-sensor datasets (IPDS). The sequences are acquired using an omnidirectional camera mounted on a VIPALAB platform 2 meters above the ground. More details about IPDS can be found on the website[1]. The sequences are available for download here[2]. Complete ground truth information is obtained from an RTK-GPS which was also mounted on the vehicle along with the other sensors. The Cezeaux sequence is very challenging because of the low resolution of images acquired with the omnidirectional camera and huge variation in environmental conditions as well as illumination.

The datasets contain huge number of images (refer to Table I), however, much smaller subsets are used in this paper. More precisely, only around two images per second are considered, resulting $3977$ images for NewCollege sequence, $1144$ for PAVIN sequence and $11571$ for Cezeaux sequence (values in parentheses of table I). Loop closures are considered when the image location is closer than $5$ meters. Our algorithm runs on a laptop computer equipped with an intel core i7 under Linux.

[1]The IP datasets website: http://ipds.univ-bpclermont.fr/

[2]http://hemanthk.me/joomla/index.php/ipdataset The sequences are organized and stored here to facilitate easy use and download.

**Algorithm 2** Image similarity computation.

**procedure 1:** LIKELIHOOD_EVALUATION($\mathbf{I}_R$,$I_q$)
  $likelihoods = []$
  **for** each image $i$ in $\mathbf{I}_R$ **do**
    $d_i = Euclidean\_Distance(\mathbf{D_i}, \mathbf{D_q})$
      ▷ $\mathbf{D_i}, \mathbf{D_q}$ are VLAD descriptors.
    $s_i = Spatial\_Similarity(\mathbf{Z_i}, \mathbf{Z_q}, W, H, b)$
      ▷ $\mathbf{Z_i}, \mathbf{Z_q}$ are lists of quantized words of SURF features.
      ▷ $W$- Width of images, $H$- Height of images
      ▷ $b$- Bin width for shift histograms.
    $likelihoods(i) = d_i * s_i$
  **end forreturn** $likelihoods$
**end procedure**

**procedure 2:** SPATIAL_SIMILARITY($\mathbf{Z_i}$, $\mathbf{Z_q}$, $W$,$H$,$b$)
  $\mathbf{M} = \mathbf{Z_i} \cap \mathbf{Z_q}$ ▷ Accumulate feature matches.
  $n = size(\mathbf{M})$ ▷ Number of matches.
  $num\_x\_bins = W/b$
  $num\_y\_bins = W/b$
  $HX = [num\_x\_bins]$ ▷ Histogram of x-coordinate shifts.
  $HY = [num\_y\_bins]$ ▷ Histogram of y-coordinate shifts.
  $shifts = []$ ▷ Vector holding x & y shifts.
  **for** each matched word $w_j$ in $\mathbf{M}$ **do** ▷ Computes shifts for matched words.
    $k_i = get\_coordinates(I_i, w_j)$
    $k_q = get\_coordinates(I_q, w_j)$
    **if** ($k_q.x \leq k_r.x$) **then**
      $\delta x = k_r.x - k_q.x$
    **else**
      $\delta x = (W - k_q.x) + k_r.x$
      $\delta y = (k_q.y - k_r.y)$;
    **end if**
    $HX.add(\delta x)$
    $HY.add(\delta y)$
    $shifts.append(\delta x, \delta y)$
  **end for**
  $x_{mean} = 0$; $y_{mean} = 0$
  $wtx = []$; $wty = []$
  **for** each bin $j$ in $HX$ **do**
    ▷ Computes weights for mean computation.
    $wtx(j) = \frac{num\_elements(HX(j))}{n}$
  **end for**
  **for** each bin $j$ in $HY$ **do**
    $wty(j) = \frac{num\_elements(HY(j))}{n}$
  **end for**
  $x_{mean} = Weighted\_Mean(HX, wtx)$
  $y_{mean} = Weighted\_Mean(HY, wty)$
  $H_{dists} = [L]$
  **for** each shift $s$ in $shifts$ **do**
    $d = Euclidean\_Distance((x_{mean}, y_{mean}), s)$
    $H_{dists}.add(d)$
  **end for**
  $similarity\_score = 0$
  **for** each bin index $j$ in $H_{dists}$ **do**
    $k = num\_elements(H_{dists}(j)$
    $similarity\_score = similarity\_score + \frac{1}{2^j} * k$
      ▷ Final similarity score..
  **end for**
**return** $similarity\_score$
**end procedure**

| Sequence | Trajectory | Velocity | #(Images) | FPS |
|---|---|---|---|---|
| PAVIN | 1.3 km | 2.3 m/sec | 8002 (1144) | 15 (2) |
| Cezeaux | 15.4 km | 2.5 m/sec | 80913 (11571) | 15 (2) |
| NewCollege | 2.2 km | 1.0 m/sec | 7854 (3977) | 3 (1.5) |

TABLE I: Datasets Description. Values in the parenthesis are the sample quantities from the original datasets that were used for our experiments.

| Parameter Name | Variable | Value |
|---|---|---|
| Bag of words vocabulary size for VLAD | $k$ | 128 |
| SURF descriptor size | $l$ | 64 |
| Full VLAD descriptor size | $k * l$ | 8192 |
| PCA VLAD descriptor size | | 256 |
| kernel width for node similarity | $\sigma$ | $0.84/1.08$ |
| Node radius | $r_n$ | $0.7/0.9$ |
| Node Similarity Threshold | $T_n$ | $0.3/0.4$ |
| Bin width for shift histogram | $b$ | 10 |
| #(bins) for distances histogram | $L$ | 10 |
| Vocabulary size for spatial similarity | | 32768 |
| Minimum supporting loop closures | $m$ | 4 |

TABLE II: Parameters

### A. Parameters and Learning

All the parameters used in our system are shown in table II. $64-$dimensional Upright SURF (USURF-64) are used as local image features. Training data is formed by randomly selecting $20\%$ of images from each sequence; SURF features extracted on all the training images are used in learning the bag of words vocabularies for VLAD computation and spatial similarity evaluation. These two vocabularies are learned using a single vocabulary tree - the first level of the tree contains 128 nodes and each of these nodes are again split with a branching factor of 4 for 4 levels having a total of $128*4^4 = 32768$ leaf nodes. Each SURF feature is quantized at two levels - one at the first level of the tree (forms a 128-word vocabulary) which is used for VLAD and the other at the leaf nodes (forms a $32768-$word vocabulary) which is used for spatial similarity analysis. Full VLAD descriptors computed on all the training images are used to learn the PCA matrix $P$.

### B. Node Similarity Analysis

As mentioned earlier in the paper, node similarity analysis involves filtering the places in the map and selecting the most similar nodes. This process leads to good results only if it produces good recall rates even with bad precision. This means that the retrieved nodes should contain all the loop closure images which leads to high recall rates. However, since the nodes also contain many more images other than the true matches, the precision values can be far from $100\%$. This is evident from figure 2a, which shows the precision-recall values obtained by varying the node similarity threshold $r_n$. For best precision, node radius for PAVIN and Cezeaux sequences was set as $r_n = 0.9$ and that of NewCollege sequence as $r_n = 0.7$. The kernel width for node similarity computation is chosen to be $\sigma = 1.2 * n_r$, so that it can give a slight cushion which accounts for noise in node similarity evaluation.

| Sequence | $r_n$ | #(Nodes) | #(images)/node | (Traj.)/node |
|---|---|---|---|---|
| NewCollege | 0.7 | 126 | 31.5 | 17.5 m |
| PAVIN | 0.9 | 60 | 19.06 | 21.6 m |
| Cezeaux | 0.9 | 572 | 20.22 | 26.2 m |

TABLE III: Node Statistics. #(Nodes) - Number of nodes of the map built on the sequence. #(images)/node - Average number of images represented by each node. (Traj.)/Node - Average trajectory length represented by each node.

Table III shows various node statistics of the maps built on the three sequences. It can be observed that the average number of images represented per node is between 20 and 30. This number is far less than the VLAD descriptor dimensionality (256) which leads to singular matrices while using statistically significant distance measures like Mahalanobis distance for node level loop closure determination.

### C. Accuracy

After the image similarity analysis, loop closure decision is taken. The precision and recall of these loop closures are shown in the Figure 2. Figure 2b illustrates the precision-recall of the loop closure decisions computed without using the spatial similarity measure and only using the VLAD descriptor similarity. The curves are computed by varying the similarity threshold $T_s$ which controls which reference nodes (and therefore reference images) are considered for image similarity analysis. We can see that $100\%$ precision is only possible till $35\%$ recall for the NewCollege sequence, $24\%$ for the PAVIN sequence. Also $100\%$ precision was never reached on the Cezeaux sequence. The reason is the low resolution images and the significant variation in illumination. Figure 2c illustrates the improved precision-recalls obtained by using the spatial constraint, where the recall of NewCollege sequence is extended by $36\%$ to a total of $71\%$ and that of PAVIN has seen an improvement by $58\%$ reaching a total of $82\%$ at $100\%$ precision. Spatial constraints also helped the Cezeaux sequence which achieved a recall of $41\%$ with $100\%$ precision, demonstrating the advantage of spatial constraints in obtaining better recall rates. It should be noted that no geometric verification has been applied to obtain the results. Loop closure maps over PAVIN, Cezeaux and NewCollege sequences are shown in and a loop closure scenario each are showed in Figure 3.

We have used our datasets and training data to compare our accuracies with the FABMAP 2.0 [3] algorithm as the authors recently opened ther code to the public. We have used the default parameters of the algorithm [1]. A vocabulary of size equal to that of ours is used but on USURF-128 features (128 dimensional). Precision and recall were analysed by varying the feature extraction threshold (varying the number of features per image), vocabulary size and prior probability thresholds for presence and absence of words in a location. Full Precision is obtained till $45\%$ recall on PAVIN, $43\%$ recall on NewCollege and $19\%$ recall on Cezeaux datasets without epipolar geometr verification step. Evidently, out technique achieves recall rates almost double that of FABMAP on all datasets.

[3]http://www.robots.ox.ac.uk/ mobile/wikisite/ pmwiki/pmwiki.php?n=Software.FABMAP

### D. Computational Time

There are five major modules in our algorithm. The three modules: local feature extraction ($120ms$), local feature quantization ($5ms$) and VLAD Extraction ($5ms$) take constant time for every image irrespective of the map size. Node similarity analysis and image similarity analysis costs on the other hand increase with the map size. Node similarity analysis takes less than $2ms$ and image similarity analysis takes $15ms$ per frame on average. From the above values, it can be seen that it takes an average of $150ms$ to process each frame, enabling to process $6-7$ frames per second. The computational costs are all observed on the Cezeaux dataset (the largest dataset with $11571$) after all the images are loaded into the map.

## VI. CONCLUSION

A hierarchical mapping model is proposed which organizes images into places and represents them as nodes. A loop closure framework which uses VLAD descriptors for retrieving most similar nodes and a spatial similarity measure on visual words that retrieves the most similar images is described. Experimental results demonstrating the sparsity, accuracy and computational time efficiency achieved by using are presented. The spatial similarity measure particularly improved the recall rates (nearly double) than that of the FAMBAP 2.0. These recall rates were achieved without the geometric verification step. Capability of processing a decent amount of frames per second is demonstrated on maps containing over $11000$ images.


## ACKNOWLEDGEMENTS

The work presented in this paper has been done as a part of the R-Discover (ANR-08-CORD-019) and ARMEN (ANR-09-TECS-020) projects funded by the French National Research Agency (l'ANR).


## REFERENCES


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

[2] ——, "Highly scalable appearance-only slam : Fab-map 2.0," in *Robotics Science and Systems*, Seattle, USA, 2009.

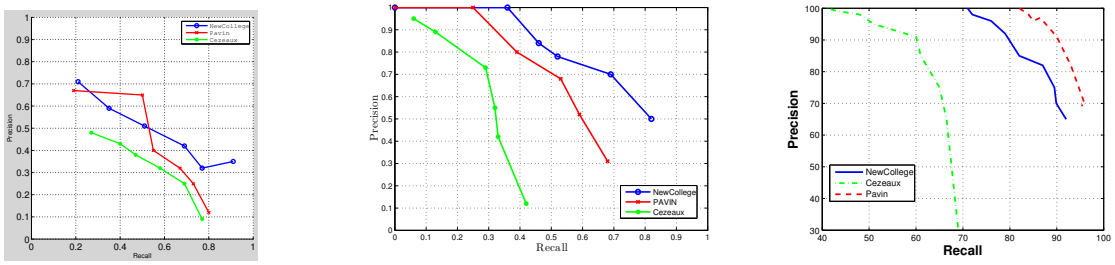
[3] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "A fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions On Robotics, Special Issue on Visual SLAM*, vol. 24(5), pp. 1027–1037, Oct. 2008.

[4] D. Galvez-Lopez and J. Tardos, "Real-time loop detection with bags of binary words," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, Sep. 2011, pp. 51–58.

[5] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," 2006, pp. 2262–2269.

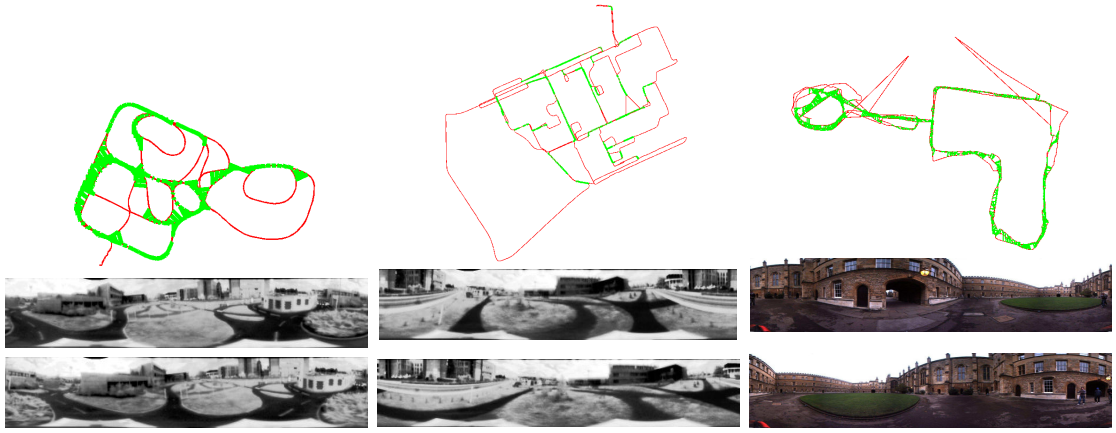
[6] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intell. Transport. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, 2010.

[7] N. Sünderhauf and P. Protzel, "Switchable constraints for robust pose graph slam," in *IROS*, 2012, pp. 1879–1884.




(a) Precision-Recall of node similarity evaluation. (b) Precision-Recall Without Spatial Similarity (c) Precision-Recall With Spatial Similarity

Fig. 2: Precision-Recall graphs at node level(Figure 2a) and image level (Figure 2b and 2c).



(a) PAVIN (b) CEZEAUX (c) NewCollege

Fig. 3: Dataset sequence trajectories are plotted in red with regions were loop closures were detected are shown in green.


[8] A. Ranganathan, "Pliss: Detecting and labeling places using online change-point detection," in *Robotics: Science and Systems*, Zaragoza, Spain, Jun. 2010.

[9] J. Lim, J.-M. Frahm, and M. Pollefeys, "Online environment mapping using metric-topological maps," *I. J. Robotic Res.*, vol. 31, no. 12, pp. 1394–1408, 2012.

[10] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *IEEE Conference on Computer Vision & Pattern Recognition*, San Francisco, USA, 2010, pp. 3304–3311.

[11] 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*, vol. 28(5), pp. 595–599, 2009.

[12] N. Nourani-Vatani and C. Pradalier, "Scene change detection for vision-based topological mapping and localization," in *IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, Taipei, Taiwan, Oct. 2010, pp. 3792–3797.

[13] M.-L. Wang and H.-Y. Lin, "A hull census transform for scene change detection and recognition towards topological map building," in *IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, Taipei, Taiwan, Oct. 2010, pp. 548–553.

[14] Z. Zivkovic, O. Booij, and B. Krose, "From images to rooms," *Robotics and Autonomous Systems*, vol. 55(5), pp. 411–418, 2007.

[15] H. Korrapati, J. Courbon, Y. Mezouar, and P. Martinet, "Image sequence partitioning for outdoor mapping," in *IEEE International Conference on Robotics and Automation, ICRA'12*, St. Paul, MN, USA, 2012, pp. 13–18.

[16] C. Murillo, P. Campos, J. Kosecka, and J. Guerrero, "Gist vocabularies in omnidirectional-images for appearance based mapping and localization," in *10th OMNIVIS*, Zaragoza, Spain, Jun. 2010, pp. 1–9.

[17] J. Kosecka, F. Li, and X. Yang, "Global localization and relative positioning based on scale-invariant keypoints," *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 27–38, 2005.

[18] C. Valgren and A. Lilienthal, "Incremental spectral clustering and seasons: Appearance-based localization in outdoor environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2008, pp. 1856–1861.

[19] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42(3), pp. 145–175, May–June 2001.

[20] F. Fraundorfer, C. Engels, and D. Níster, "Topological mapping, localization and navigation using image collections," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'07*, San Diego, USA, Oct. 2007, pp. 3872–3877.

[21] R. Paul and P. Newman, "Fab-map 3d: Topological mapping with spatial and visual appearance," in *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, USA, May 2010, pp. 2649–2656.

[22] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: http://www.cs.ubc.ca/˜ lowe/keypoints/

[23] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.

[24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2008, pp. 1–8.