Robust Loop Closing Over Time

Yasir Latif, Cesar Cadena and José Neira
Instituto de Investigación en Ingeniería de Aragón, I3A
Universidad de Zaragoza
Zaragoza, Spain
Email: {ylatif, ccadena, jneira} @unizar.es

***Abstract*—Long term autonomy in robots requires the ability to reconsider previously taken decisions when new evidence becomes available. Loop closing links generated by a place recognition system may become inconsistent as additional evidence arrives. This paper is concerned with the detection and exclusion of such contradictory information from the map being built, in order to recover the correct map estimate. We propose a novel consistency based method to extract the loop closure regions that agree both among themselves and with the robot trajectory over time. We also assume that the contradictory loop closures are inconsistent among themselves and with the robot trajectory. We support our proposal, the RRR algorithm, on well-known odometry systems, e.g. visual or laser, using the very efficient graph optimization framework g2o as back-end. We back our claims with several experiments carried out on real data.**

I. Introduction

The ability to detect that past place recognition decisions (also known as the loop closing problem) were incorrect, and to recover the accurate state estimate once they are removed, is crucial for lifelong mobile robotic operations. No matter how robust a place recognition system might be, there always exists the possibility of getting just one false positive loop closing hypothesis. This can be catastrophic for any estimation algorithm. Only by collecting more evidence over time we can detect and correct these mistakes in place recognition. Reliability increases with redundancy during the verification process.

Consider long term navigation, see Fig. 4(b) (best viewed in color), where information is collected session by session and place recognition is used to improve the estimation after every session. Failures may happen, but With the arrival of new evidence and the reconsideration of decisions taken earlier, the estimation can be corrected.

To achieve this, we need to realize that the place recognition system has generated wrong constraints, remove them if necessary, and recompute the estimation. We propose a novel consistency method based on consensus: wrong loop closure constraints usually are in contradiction among themselves, with the correct ones, and with the sequential movement constraints (odometry) of the robot. At the same time, the correct ones form consensus among themselves and with the odometry.

Our method, the Realizing, Reversing, Recovering (RRR) algorithm, works with the pose graph formulation. It is part of the back-end of a SLAM system and is therefore independent of the type of sensor used for odometry or place recognition. All our method needs is a system that is able to generate a pose graph for the sequential poses and a place recognition system that can provide constraints for loop closure.

In the next section, we discuss relevant work and highlight the need for a robust loop closing method. In section III we detail our proposal, the RRR algorithm, and carry out real experiments in section IV. Finally, in section V and VI discussion and conclusions about the work are presented, along with possible future work.

II. Related Work

Several approaches to persistent mapping have appeared in the robotics literature in the recent past. Konolige and Bowman [8] presented a stereo visual odometry system that works together with a bag of words place recognition system to build multiple representations of dynamic environments over time. Multiple map stitching, recovery from odometry failures, loop closing, and global localization, all rely on the robustness of the place recognition system. "Weak links" are removed when place recognition is able to close loops, making it prone to errors when the place recognition closes loops wrongly. Similarly, McDonald et al. [11] presented a multi-session 6 DOF Visual SLAM system using "anchor nodes". In their approach place recognition is assumed to be perfect and its output is trusted every time. Sibley et al. [17] presented a relative bundle adjustment approach for large scale topological mapping. They show an example of mapping from London to Oxford, over a trajectory of about $121km$. They also use appearance based place recognition and are therefore in danger of making wrong decisions in case of incorrect place recognition. All of these approaches to large scale and persistent mapping rely on a place recognition system with zero false positives.

Appearance based loop closing approaches that work in real time usually follow the bag of words approach [18]. FabMap, from Cummins and Newman [3], uses a bag of words approach with probabilistic reasoning that has been shown to work robustly over a long trajectory. The BoW-CRF system [1] generates loop closings using a bag-of-words and carries out verifications using conditional random fields. This improves the robustness of the system. In any case, neither approach can guarantee 100% accuracy. Olson [12] proposed a hypothesis verification method for loop closure constraints using graph partitioning based on spectral clustering. Another approach is to delay decision making and maintain multiple topologies of the map with an associated belief for each one. Ranganathan and Dellaert [14] follow this approach using Rao-Blackwellized particle filter. However, they do not explicitly show how their system is affected by and recovers from wrong loop closures. Their method is also unique in the sense that it uses the estimation process itself to reason about possible loop closures. A similar estimation based reasoning approach using pose graph formulation was presented by Sünderhauf and Protzel [20] in which a robust SLAM back end using "switch factors" was proposed. The central idea is to penalize those loop closure links during graph optimization that deviate from the constraints they suggest between two nodes. Similar to our approach, they change the topological structure of the graph based on identification and rejection of wrong loop closures. In contrast, however, they assess the validity of every loop closure on its own, without forming a general consensus using all the available information. In cases where there are a number of hypotheses that suggest the same but wrong loop closings (for example due to perceptual aliasing in long corridors), overall consensus helps us in rejection of such outliers. Their method suggest a continuous function governing the state of "switch factors" which does not make sense in all the cases.

III. Our proposal: the RRR algorithm

Since place recognition is a very important step in map estimation, there is a need for a robust mechanism that can identify incorrect place recognition links. We propose a robust consistency-based loop closure verification method using the pose graph formulation based on the observation that correct loop closure in conjunction with odometry can help in the detection of wrong loop closures. Our method follows the line of work in which the estimation process itself is used in making the distinction between correct and false loop closures. Thanks to the maturity of the Simultaneous Localization and Mapping (SLAM) problem, in the last years several very efficient estimation methods have been published as for instance iSAM by Kaess et al. [7], HOG-Man by Grisetti et al. [6], and g2o by Kümmerle et al. [10].

The graph based formulation for SLAM, the so-called "graph-SLAM", models robot poses as nodes in a graph where links from odometry or loop closures form edges or "constraints". Let $\mathbf{x} = (x_1 \ldots x_n)^T$ be a vector of parameters that describe the configuration of the nodes. Let ω_{ij} and Ω_{ij} be the mean and the information of the observation of node j from node i. Given the state $\mathbf{x}$, let the function $f_{ij}(\mathbf{x})$ be a function that calculates the perfect observation according to the current state. The residual r_{ij} can then be calculated as:

$$r_{ij}(\mathbf{x}) = \omega_{ij} - f_{ij}(\mathbf{x}) \tag{1}$$

Constraints can either be introduced by odometry which are sequential constraints ($j = i + 1$), or from place recognition system which are non-sequential. The amount of error introduced by each constraint weighed by its information can be calculated as:

$$d_{ij}(\mathbf{x})^2 = r_{ij}(\mathbf{x})^T \Omega_{ij} r_{ij}(\mathbf{x}) \tag{2}$$

and therefore the overall error, assuming all the constraints to be independent, is given by:

$$D^2(\mathbf{x}) = \sum d_{ij}(\mathbf{x})^2 = \sum r_{ij}(\mathbf{x})^T \Omega_{ij} r_{ij}(\mathbf{x}) \tag{3}$$

The solution to graph-SLAM problem is to find a state $\mathbf{x}^*$ that minimizes the overall error.

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum r_{ij}(\mathbf{x})^T \Omega_{ij} r_{ij}(\mathbf{x}) \tag{4}$$

Iterative approaches such as Gauss-Newton or Levenberg Marquadt can be used to compute the optimal state estimate [10].

We divide the constraints into two sets; S contains sequential links and R contains links from place recognition. Since all constraints are mutually independent, the error in (3) be written as:

$$D^2(\mathbf{x}) = \sum_{(i,j)\in S} d_{ij}(\mathbf{x})^2 + \sum_{(i,j)\in R} d_{ij}(\mathbf{x})^2 \tag{5}$$

We further divide the set R into n disjoint subsets R_k, where each subset only contains topologically related constraints (sequences of links that relate similar portions of the robot trajectory) such that $R = \cup_{k=1}^{n} R_k$ and $\forall (i \neq j) R_i \cap R_j = \emptyset$. We term each of theses subsets as "clusters".

Then the error for set R can be written as:

$$\sum_{(i,j)\in R} d_{ij}(\mathbf{x})^2 = \sum_{c=1}^{n} \sum_{(i,j)\in R_c} d_{ij}(\mathbf{x})^2 = \sum_{c=1}^{n} d_{R_c}(\mathbf{x})^2 \tag{6}$$

where $d_{R_c}(\mathbf{x})^2$ is the error contributed by the cth subset. This simply means that the overall error introduced due to place recognition constraints is the sum of the individual errors of each cluster.

Assuming that we do not have any outliers in odometry, the error in (3) is caused practically only by the loop closing links. Once we iterate to find the optimal state, the error in the odometry is no longer zero. This increase in odometry error gives us a measure of the metric change that must take place so that the graph conforms to place recognition constraints. This error will be smaller when the corresponding place recognition constraints are correct. It is because comparatively smaller metric change is needed as opposed to the change required by wrong constraints. Moreover, clusters that suggest the same change would cause smaller errors among them as compared to conflicting clusters. By measuring how much errors the clusters introduce, we can detect which clusters agree with each other. A point to note here is that even though odometry drifts with time, it is still a useful measure of the underlying topology of the graph.

Consider the case, when each of these clusters is the only place recognition constraint in the graph. In absence of other clusters, this cluster can make the odometry converge to an acceptable solution, where acceptable means that the overall errors are within some statistical threshold. If this cluster alone deforms the graph enough to make the solution unacceptable, it is highly likely that this cluster is suggesting a wrong place recognition decision. This forms the basis of our intra-cluster test. Mathematically, for any cluster R_i to be individually consistent, the following two conditions must hold:

$$D_G^2(\mathbf{x}) = \sum_{(i,j)\in R_i} r_{ij}(\mathbf{x})^T \Omega_{ij} r_{ij}(\mathbf{x}) + \sum_{(i,j)\in S} d_{ij}(\mathbf{x})^2 < \chi^2_{\alpha,d_G} \tag{7}$$

where d_G are the degrees of freedom of the whole graph. Moreover,

$$D_l^2(\mathbf{x}) = r_{ij}(\mathbf{x})^T \Omega_{ij} r_{ij}(\mathbf{x}) < \chi^2_{\alpha,d_l},\ (i,j) \in R_i \tag{8}$$

ensures that if there are any outliers within the cluster they are omitted. d_l are the degrees of freedom of each constraint. If the criterion in (7) is not met, the whole cluster is rejected. Constraints not meeting the criterion in (8) are removed from the cluster and the rest of the links are accepted as being individually consistent.

Similarly, for a selected subsets of clusters C, we term the clusters in C to be jointly consistent if:

$$D_C^2(\mathbf{x}) = \sum_{c=1}^{|C|} \sum_{(i,j)\in R_c} r_{ij}(\mathbf{x})^T \Omega_{ij} r_{ij}(\mathbf{x}) < \chi^2_{\alpha,d_C} \tag{9}$$

and

$$D_G^2(\mathbf{x}) = D_C^2(\mathbf{x}) + \sum_{(i,j)\in S} r_{ij}(\mathbf{x})^T \Omega_{ij} r_{ij}(\mathbf{x}) < \chi^2_{\alpha,d_G} \tag{10}$$

This first criteria ensures that the present clusters are consistent with each other while the second one ensures the consistency of the clusters with the odometry links.

We also advocate that rather than testing a single hypothesis on its own, clusters are more robust and identify regions of loop closures rather than single pose-to-pose correspondence. This is the case when appearance based place recognition systems generate hypotheses. Having said that, our method does not pose any restriction on the minimum number of links in a cluster and can handle clusters composed of just single links.

An overview of our method is provided in the following section.

A. Method

1) Clustering: Our method starts by collecting topologically related loop closing hypotheses into clusters. Clusters represent sequences of loop closures that relate similar portions of trajectory. We use a simple incremental way to group them using timestamps. We proceed as follow: with the first loop closure that arrives, we initialize the first cluster R_1. Then, we decide according to (11) if the next loop closure arriving belongs to same cluster or a new cluster needs to be initialized.

$$\begin{aligned} &\omega_{i,j} \in R_k \iff \\ &\quad \exists \omega_{p,q} \in R_k \ \mid \ \|t_i - t_p\| \le t_g \land \|t_j - t_q\| \le t_g \end{aligned} \tag{11}$$

where t_i means the timestamp related to the node i, and t_g can be selected according to the rate at which the place recognition system runs. This threshold defines the cluster neighbourhood. In our experiments, we consider loop closing hypotheses less than $t_g = 10s$ apart to be a part of the same cluster.

A cluster is considered closed if for t_g time no new links are added to it.

2) Intra-Cluster Consistency: After clustering hypotheses together, the next step is to compute the internal consistency for each cluster. This involves optimizing the pose graph with respect to just this single cluster and checking which links satisfy the χ^2_{α,d_l} bound. The links inside the cluster that do not pass this test are removed from the cluster and are no longer considered in the optimization. Algorithm 1 describes the intra-cluster consistency. This procedure is carried out for each cluster and it can be performed as soon as a cluster is closed.

Algorithm 1 Intra_Cluster_Consistency

```
Input: poses, slinks, cluster of rlinks
Output: cluster
  add poses, slinks to PoseGraph
  PoseGraphIC ← PoseGraph
  add cluster to PoseGraphIC
  optimize PoseGraphIC
  if D_G^2 < χ^2_{α,d_G} then
    for each rlink_l ∈ cluster do
      if D_l^2 < χ^2_{α,d_l} then
        Accept rlink_l
      else
        Reject rlink_l
      end if
    end for
  else
    Reject cluster
  end if
```

3) The RRR algorithm: Having established intra-cluster consistency, we now look for clusters that are mutually consistent. We initially assume that all the clusters are consistent and carry out optimization by including all of them in the optimization problem. Once optimized, we check for any links whose residual error satisfies the χ^2 test. The clusters to which these links belong are then selected and added to the candidate set to be evaluated for joint consistency according to eqs. (9) and (10).

We accept the clusters that are jointly consistent and term them $goodSet$. At this point, we remove the $goodSet$ from the optimization and try to re-optimize with the remaining clusters. The idea behind doing so is that in the absence of the good clusters, other correct clusters will be able to pass the threshold tests. A good analogy would be that of athletes running in a special kind of race in which the observer is only able to see who crosses the finish line first. In order to

Algorithm 2 Inter_Cluster_Consistency

Input: $goodSet$, $candidateSet$, PoseGraph
Output: $goodSet$, $rejectSet$
  PoseGraphJC $\leftarrow$ PoseGraph
  add $(goodSet, candidateSet)$ to PoseGraphJC
  $rejectSet \leftarrow \{\}$
  optimize PoseGraphJC
  if $D_C^2 < \chi^2_{\alpha,d_C} \wedge D_G^2 < \chi^2_{\alpha,d_G}$ **then**
    $goodSet \leftarrow \{goodSet, candidateSet\}$
  else
    find the $cluster_i \in candidateSet$ with largest CI
    remove $cluster_i$ from $candidateSet$
    $rejectSet \leftarrow cluster_i$
    if $\neg$isempty $candidateSet$ **then**
      $(goodSet, rSet) \leftarrow$
      $Inter\_Cluster\_Consistency(goodSet, candidateSet)$
      $rejectSet \leftarrow \{rejectSet, rSet\}$
    end if
  end if

Algorithm 3 RRR

Input: $poses$, $slinks$, $\mathcal{R}$ set of clusters containing $rlinks$
Output: $goodSet$ of $rlinks$
  add $poses, slinks$ to PoseGraph
  $goodSet \leftarrow \{\}$
  $rejectSet \leftarrow \{\}$
  loop
    PoseGraphPR $\leftarrow$ PoseGraph
    $currentSet \leftarrow R \backslash \{goodSet \cup rejectSet\}$
    $candidateSet \leftarrow \{\}$
    add $currentSet$ to PoseGraphPR
    optimize PoseGraphPR
    for each $cluster_i \in currentSet$ **do**
      if $\exists D_l^2 < \chi^2_{\alpha,d_l} \mid rlink_j \in cluster_i$ **then**
        $candidateSet \leftarrow \{candidateSet, cluster_i\}$
      end if
    end for
    if isempty$(candidateSet)$ **then**
      STOP
    else
      $s = goodSet.size$
      $(goodSet, rSet) \leftarrow$
      $Inter\_Cluster\_Consistency(goodSet, candidateSet)$
      if $goodSet.size > s$ **then**
        $rejectSet \leftarrow \{\}$
      else
        $rejectSet \leftarrow \{rejectSet, rSet\}$
      end if
    end if
  end loop

establish which runner would end up in the second position, we ask the winner to not participate in the race and conduct the race again. We continue asking the winner each time to join the previous winners, to see who will take the next spot. The winners in our case are the member of the $goodSet$. This is show in Algorithm 2. As long as we keep finding clusters that are jointly consistent with the $goodSet$, it will grow.

An important point to mention here is the use of the $rejectSet$. The reject set contains all the clusters that we checked in the last iteration but found them to be inconsistent with the $goodSet$. We omit them from the optimization process until something is added to the $goodSet$. The main algorithm is given in the Algorithm 3. The algorithm terminates when we are no longer able to find any clusters to add to the $candidateSet$.

Long term operation: The algorithm described above works on data from a single session. In order to integrate multiple sessions, we run the algorithm on the first session, select the $goodSet$ and optimize the graph using these correct loop closing hypothesis. The optimized graph is then used in conjunction with the second session. The loop closing hypothesis from the first are carried over to the second session, and we now look for a $goodSet$ that is correct for both the sessions. In this way, we can extend the algorithm for n sessions, at the end of which we have a $goodSet$ consisting of hypothesis that have support from all the n sessions.

IV. Experiments

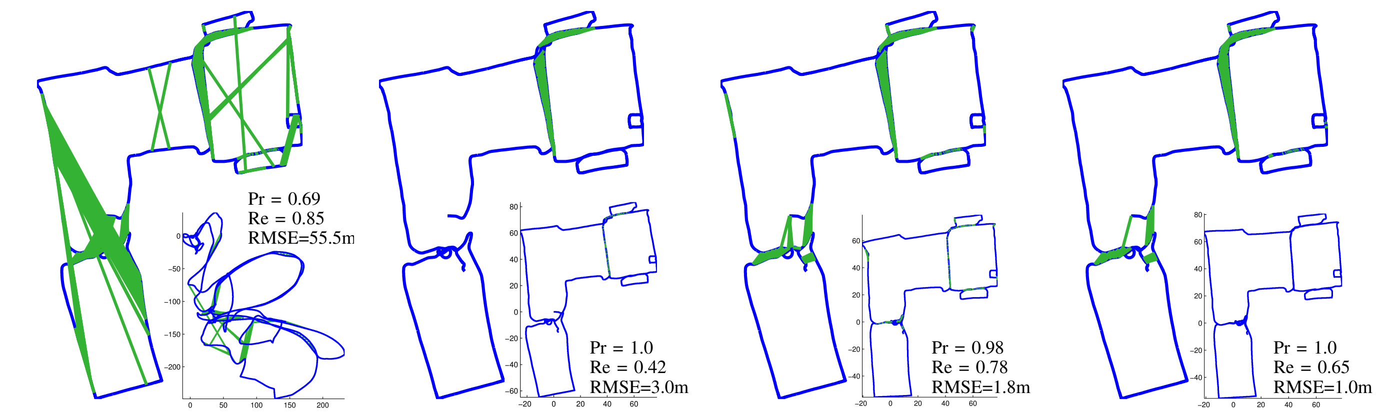
We have evaluated the RRR algorithm on an indoor environment (Bicocca campus) from the RAWSEEDS project [15], an outdoor environment (Bovisa campus) from the same source, and on the NewCollege dataset [19]. Also, we show two multi-session experiments, 5 large scale sessions on Bicocca, and 36 sessions on a changing environment of an office floor at the University of Lincoln [4]. As front-end we use the BoW algorithm of [1], and as back-end we use the g2o framework, configured with the Gauss-Newton method and four iterations.

A. Comparisons

In the experiments, the RRR algorithm is used in combination with a weak place recognition algorithm (BoW), that is known to exhibit many false positives. In this way, we test the robustness of RRR to incorrect loop closing links. We compare our method with a branch and bound search (BBS) traversal of the whole solution space, computationally very demanding but guaranteed to find the largest subset of hypotheses that is jointly consistent (according to Algorithm 2). We also compare with Olson's method [12] and a modified version (see below). This is for verification within a single session.

Olson's method is mainly used for outlier rejection and finding inliers that are in some sense consistent with each other. Without any modifications and calculating the consistency just within the "groups" (that we call clusters), the method has very low recall. We believe this happens because a few very well connected prevent other *correct* but comparatively weakly connected clusters for entering the final solution. In the modified version, we calculate the consistency between all the given loop closure links. This makes the algorithm unreliable because Dijkstra links are derived from long trajectory links. At the same time, the adjacency criterion takes a long time to compute. The only advantage is that it dilutes the effect of a few very well connected clusters. The results for this method are shown in Fig. 1. The method accepts some false positive loop closures mainly because of the aforementioned reason.

BBS traverses the solution space to compute the largest set of clusters of loop closing hypotheses that are jointly consistent. As can be seen in Fig. 2, BBS selects the same



(a) Loop closures from a BoW system. (b) Output of the Olson's method. (c) Modified Olson's method's output. (d) Output of our proposal.

Fig. 1. Experiment used for comparisons against other possible approaches. This is one of the sessions from Bicocca campus shown in Fig. 4(b). **Inset**: result of optimization carried out using accepted links. We show the precision (Pr), recall (Re) and root mean square error (RMSE) for each method.

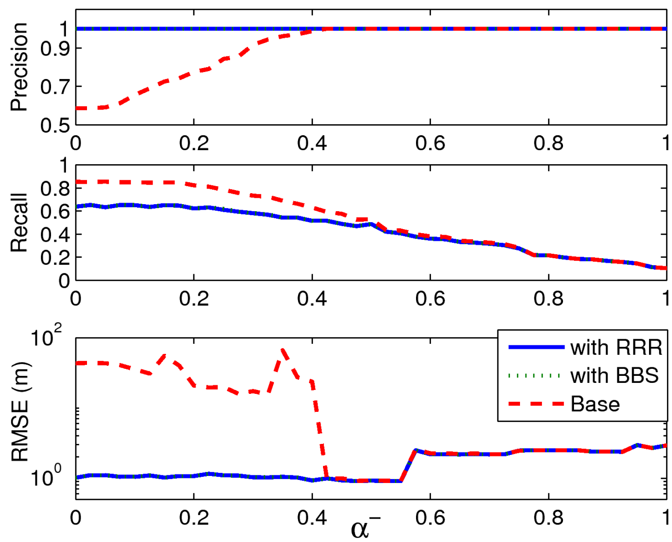


Fig. 2. Effect of the RRR algorithm over the precision (top), recall (middle), and root mean square error (bottom) compared with the base approach (BoW+GC) as we sweep the parameter α^- of the BoW system. This is over the same Bicocca session used in Fig. 1. The curves for BBS in each case are superimposed with those of RRR

clusters as our method but at the expense of much higher computational complexity.

To further evaluate the performance of the RRR algorithm, we sweep the minimum confidence level of acceptance α^- of the BoW system, and compute the precision, recall, and the root mean square error of the optimized graph for the constraints for each value of α^-. In Fig. 2, we show the result without our method (base) and with our RRR algorithm. RRR obtains full precision in the whole range and better root mean square error than the base approach. It is important to note that although sacrificing recall, we obtain better metric accuracy. This shows the disastrous effect of just a single false positive on the optimization process. When the base approach reaches full precision our method accepts all the constraints, obtaining the same recall and errors.

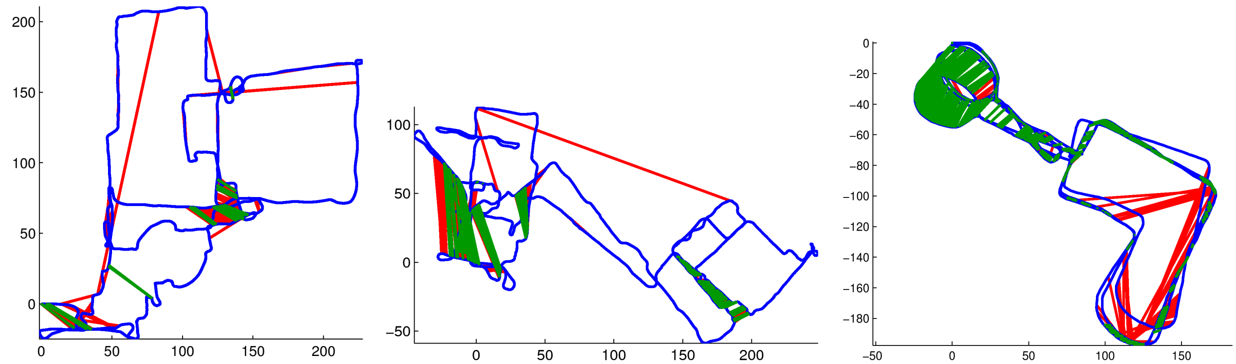
We test the RRR algorithm in several real datasets. The odometry information comes from visual+wheel, laser, and stereo odometry, for outdoors-Bovisa (Fig 3(a)), mixed-Bovisa (Fig. 3(b)), and NewCollege (Fig. 3(c)) respectively. We use BoW as place recognition with geometrical checking (GC) based on finding the rigid-body transformation with stereo correspondences. In Fig. 3 we show the results. It can be seen that our method successfully rejects all the wrong loop closing links.

B. Long term operation

The task of long term navigation involves robustly joining multiple sessions into a single map. For this purpose, we use multiple sessions from RAWSEEDS [15]. The dataset contains five overlapping runs in indoor environment and provides wheel odometry, laser scans, and images from a camera mounted on the robot.

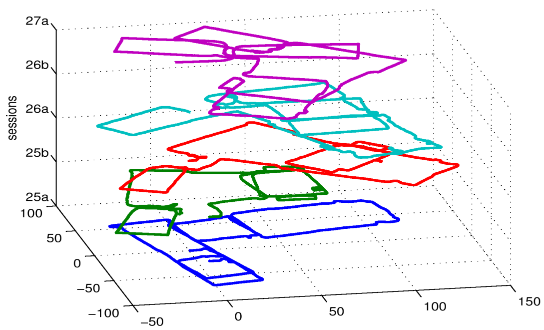
In order to use the RRR algorithm, we need some form of odometry as well as the loop closing hypotheses. We use laser scans from the dataset to compute the laser odometry using a simple scan matcher, see Fig. 4(a), and use BoW plus GC, see Fig. 4(b), to compute appearance based place recognition links. These are the inputs to our method.

We run our method incrementally on the five sessions. The first iteration of the algorithm works with data from the first session and computes the set of consistent clusters. The optimized first session is shown at the top-most position in Fig. 5. In the inset, the input to this step can be seen. This optimized session is used in conjunction with the second session and combined reasoning is done on both. During the second session, due to lack of evidence and odometry error we accept some false positives, the effect of which can be seen in the 3D plot. During the next session, as more evidence becomes available, we are able to recover from the previous mistake. As can be seen from the successive improvement in the overall estimate, our method is able to take into account more evidence as it become available and incorporate it into the overall map. The last (bottom most) in Fig. 5 is the final map estimate based on evidence and combined reasoning on

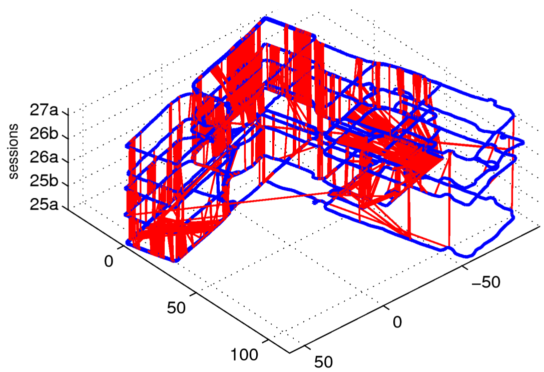


(a) Outdoors (Bovisa campus) (b) Mix outdoors-indoors (Bovisa campus) (c) Outdoors (New College)

Fig. 3. Results of the RRR algorithm over different environments. The loop closures are obtained from an appearance based place recognition algorithm. Odometry is in blue, accepted links in green and rejected ones in red.



(a) Laser odometry given by a simple scan matching.



(b) Loop closing constraints from BoW+GC , $\alpha^{-} = 0.15$

Fig. 4. Bicocca multi-session experiment in an indoor environment

all the sessions.

In the same way, we run our method on the 36 sessions of Lincoln dataset. This dataset provides omnidirectional images and laser scans. We compute the odometry by simple scan matching of the laser returns, see Fig. 6(a), and the loop closure constraints taking the best match comparing the GIST descriptors of the Omnidirectional images as described in [16], see Fig. 6(b). The final result after our proposal is shown in Fig. 7. The algorithm takes $40.5s$ to complete, with no false positives.

V. Discussion

The method proposed in this work is applicable for metric pose graph formulation. Any SLAM system that constructs

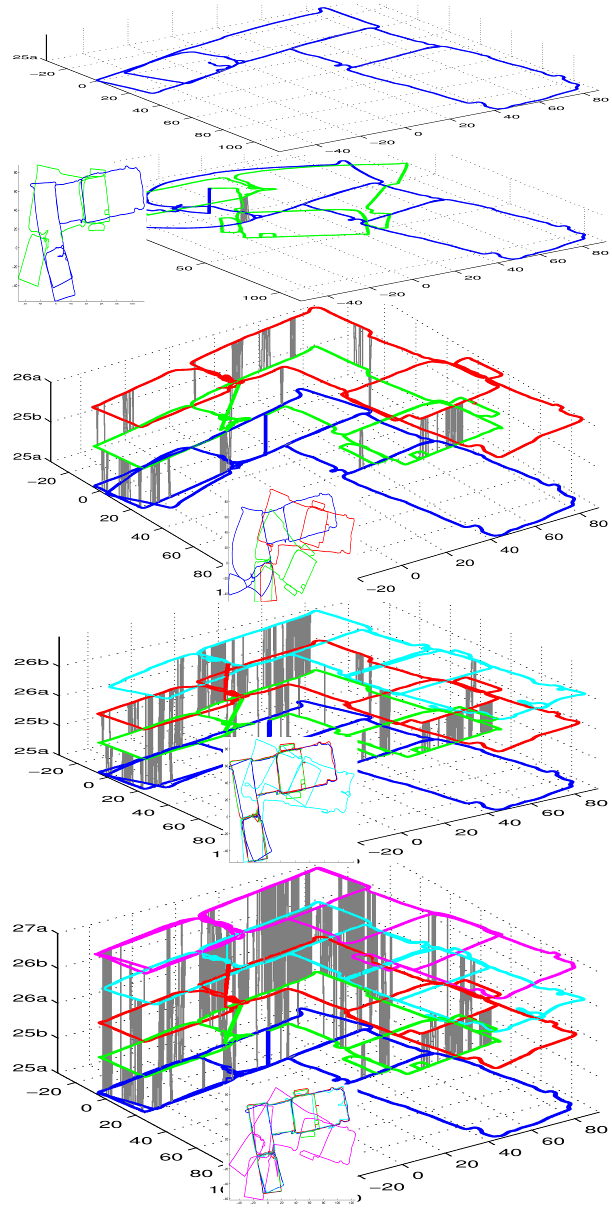


Fig. 5. Results for long term operation using Bicocca Feb-25a(blue), Feb-25b(green), Feb-26a(red), Feb-26b(cyan) and Feb-27a(magenta) Inset: Input to current step. 3D plot : Optimized graph as output of our method

a metric map can be used in conjunction with our system for loop closure verification over time. The method is not restricted to any type of sensor. The pose graph, for example can be constructed from either wheel odometry, laser odometry or visual odometry or any other sensor. Similarly, the method is independent of the mechanism used for place recognition. This gives our method the flexibility to be used together with any SLAM front end that can generate a pose graph and the corresponding place recognition constraints.

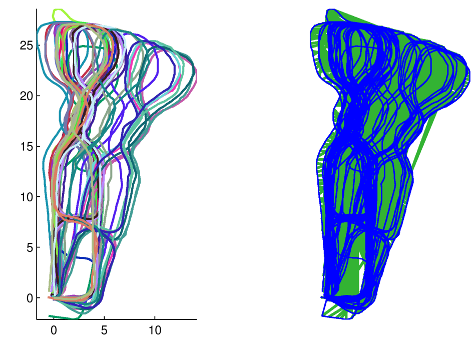
For timing information, we compare our method again with BBS and Olson's method. BBS explores an interpretation tree which, in the case of our extension to clusters, is a binary tree. The worse case complexity of BBS is therefore $O(2^n)$ where n is the number of clusters. In our experiment, after clustering and intra-cluster consistency ($6s$), the algorithm took $12s$ for a binary tree of depth 20. There are randomized versions to check joint consistency such as the RJC algorithm proposed by Paz et al. [13] or the 1-point RANSAC algorithm proposed by Civera et al. [2]. They are based on the random sample consensus paradigm [5], and need a minimum number of data points to fit a model. In the case addressed here, with arbitrary trajectories and without prior knowledge of the number of loop closure areas, there is no clear way to determine the minimum number of data points needed to fit the model.

For Olson's method, the most expensive operation is the computation of the consistency matrix. For N hypotheses the complexity of calculating the consistency matrix is $O(N^2)$. It should be noted that N, the number of hypothesis, is much larger than the number of clusters n. In our experiment, the running time for this method was $8s$.

For n clusters, our method needs to carry out n intra-cluster consistency tests. More over, in the worst case if all the clusters are considered for joint consistency after the first optimization and are not jointly consistent, we need n joint consistency checks. This makes our method linear $O(n)$ in the number of clusters. The actual cost of carrying out each step, however, depends on the underlying graph and the optimization method used. For the experiment in Fig. 1 , our method takes $8s$ to calculate the solution, spending 6 of them in clustering and intra-cluster consistency checks. As we mentioned above, these two operations can be carried out in real time.

As we mentioned earlier, currently we consider that links less than $t_g = 10s$ apart belong to the same cluster, in accordance with the usual frequency of the place recognition algorithm, $1fps$. A smaller value for t_g results in more clusters, and a corresponding increase in computational cost, with precision-recall not being affected. A large value can result in low recall because large clusters containing valid loop closing links could be rejected more frequently.

One of the main advantages of our method is that while finding a solution similar to BBS, the algorithm works in $O(n)$ as opposed to $O(2^n)$. For the first two sessions of the long term experiment in Fig. 5, BBS takes $1537.6s$ compared to our method which runs in $34.8s$. For the complete experiment, BBS takes over two days to come up with a solution, while our method takes $314.5s$.



(a) Laser odometry. (b) Loop closure constraints.

Fig. 6. Lincoln dataset.

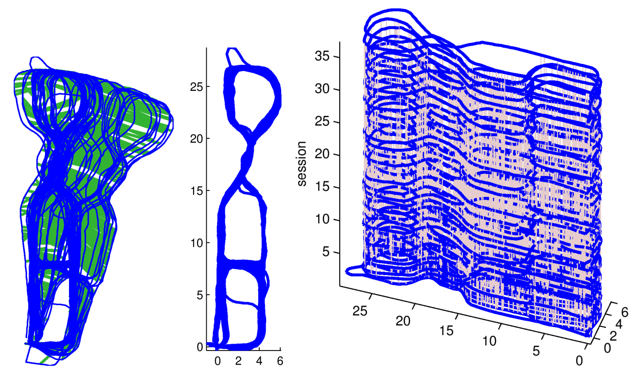


Fig. 7. Results of our RRR algorithm for long term operation using the 36 sessions of Lincoln dataset. Left: Loop closure constraints accepted. Center and right: Optimized graph as output of our method.

The long term experiments demonstrate the importance of incorporating new evidence session by session to realize possible previous failures. It is important to mention here that as the number of sessions continue increasing, further criteria need to be taken into account in order to join and/or forget past evidence. For instance, if we track the performance history of the clusters we can decide to forget some of them if they are always rejected (as would be the case for perceptual aliasing). Also we can fuse verified clusters, from different times, if they are metrically close after the optimization is carried out.

Here, we assumed that the errors in odometry links are small. We think that it is a plausible assumption given the higher frame rate of odometry w.r.t place recognition rate and the state of the art in odometry techniques. Nevertheless, if a failure occurs the approaches of "weak links" [9] or "anchor nodes" [11] can be used to separate the experiment into different sessions.

VI. Conclusion

The task of long term operation requires that at any given time, the map be updated using all the available information. Therefore, a method is required that can reason based all on the available evidence and as a result produce the best map estimate in light of that evidence. We therefore use the term "loop closing over time" for methods that are able to reason about the consistency of loop closing hypotheses not only in space but also along the time dimension. In this work, we have presented a novel consistency based approach to this problem. We support with evidence our claim that the estimation process itself can be used to establish the validity of loop closing hypotheses. We back up our claims with experiments showing the long term operation abilities of our system as well as its robustness when compared to state of the art data association methods.

As future work, we will explore extensions of our RRR algorithm in order to consider the possibility of incorrect odometry links.

Acknowledgements

This research has been funded by the Dirección General de Investigación of Spain under projects DPI2009-13710 and DPI2009-07130, and by DGA-FSE(group T04).

References

[1] C. Cadena, D. Gálvez-López, J.D. Tardós, and J. Neira. Robust place recognition with stereo sequences. *IEEE Transaction on RObotics*, 28(4), 2012. doi: DOI:10.1109/TRO.2012.2189497. to appear.

[2] J. Civera, Oscar G. Grasa, A. J. Davison, and J. M. M. Montiel. 1-Point RANSAC for extended kalman filtering: Application to real-time structure from motion and visual odometry. *J. Field Robot.*, 27:609–631, September 2010. ISSN 1556-4959. doi: http://dx.doi.org/10.1002/rob.v27:5. URL http://dx.doi.org/10.1002/rob.v27:5.

[3] M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, 2010. doi: 10.1177/0278364910385483. URL http://ijr.sagepub.com/content/early/2010/11/11/0278364910385483.abstract.

[4] F. Dayoub, G. Cielniak, and T. Duckett. Long-term experiments with an adaptive spherical view representation for navigation in changing environments. *Robotics and Autonomous Systems*, 59(5):285–295, 2011.

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[6] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2d and 3d mapping. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 273 –278, may 2010. doi: 10.1109/ROBOT.2010.5509407.

[7] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Shanghai, China, May 2011.

[8] K. Konolige and J. Bowman. Towards lifelong visual maps. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1156–1163. IEEE, 2009.

[9] K. Konolige, J. Bowman, J.D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based Maps. *The International Journal of Robotics Research*, 29(8): 941–957, 2010. doi: 10.1177/0278364910370376. URL http://ijr.sagepub.com/content/29/8/941.abstract.

[10] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[11] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J.J. Leonard. 6-DOF Multi-session Visual SLAM using Anchor Nodes. In *European Conference on Mobile Robotics, ECMR*, 2011.

[12] E. Olson. Recognizing places using spectrally clustered local matches. *Robotics and Autonomous Systems*, 57 (12):1157–1172, December 2009.

[13] L. M. Paz, J. Guivant, J. D. Tardós, and J. Neira. Data association in *O(n)* for divide and conquer SLAM. In *Proc. Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[14] A. Ranganathan and F. Dellaert. Online probabilistic topological mapping. *The International Journal of Robotics Research*, 30(6):755–771, May 2011. doi: 10.1177/0278364910393287. URL http://ijr.sagepub.com/content/early/2011/01/23/0278364910393287.abstract.

[15] RAWSEEDS. Robotics advancement through Webpublishing of sensorial and elaborated extensive data sets (project FP6-IST-045144), 2009. http://www.rawseeds.org/rs/datasets.

[16] A. Rituerto, A. C. Murillo, and J. J. Guerrero. Semantic labeling for indoor topological mapping using a wearable catadioptric system. *to appear in Robotics and Autonomous Systems, special issue Semantic Perception, Mapping and Exploration*, 2012.

[17] G. Sibley, C. Mei, I. Reid, and P. Newman. Vast-scale outdoor navigation using adaptive relative bundle adjustment. *The International Journal of Robotics Research*, 29(8):958–980, 2010.

[18] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, October 2003.

[19] 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, May 2009. ISSN 0278-3649. doi: http://dx.doi.org/10.1177/0278364909103911. URL http://www.robots.ox.ac.uk/NewCollegeData/.

[20] N. Sünderhauf and P. Protzel. Brief-gist - closing the loop by simple means. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1234 –1241, sept. 2011. doi: 10.1109/IROS.2011.6094921.