# Robust Place Recognition with Stereo Cameras

César Cadena, Dorian Gálvez-López, Fabio Ramos, Juan D. Tardós and José Neira

*Abstract*— **Place recognition is a challenging task in any SLAM system. Algorithms based on visual appearance are becoming popular to detect locations already visited, also known as loop closures, because cameras are easily available and provide rich scene detail. These algorithms typically result in pairs of images considered depicting the same location. To avoid mismatches, most of them rely on epipolar geometry to check spatial consistency. In this paper we present an alternative system that makes use of stereo vision and combines two complementary techniques: bag-of-words to detect loop closing candidate images, and conditional random fields to discard those which are not geometrically consistent. We evaluate this system in public indoor and outdoor datasets from the Rawseeds project, with hundred-metre long trajectories. Our system achieves more robust results than using spatial consistency based on epipolar geometry.**

## I. Introduction

In this paper, we consider the problem of recognising locations based on scene geometry and appearance. This problem is particularly relevant in the context of large-scale global localisation and loop closure detection in mobile robotics. We propose to solve this problem by using two complementary techniques. The first one is based on the bag-of-words method (BoW) [1], which reduces images to sparse numerical vectors by quantising their local features. This enables quick comparisons among a set of images to find those which are similar. Some techniques derived from BoW have been successfully applied to loop closing-related problems ([2], [3]) but exhibit false positives. Obtaining an incorrect loop closure can result in a critical failure for the SLAM algorithms. We use a hierarchical BoW [4] improved with adaptive thresholding to detect scenes that are similar and to enforce temporal consistency. These similar scenes are loop closure candidates, and can be verified by CRF-Matching, the second technique considered. CRF-Matching is an algorithm based on Conditional Random Fields (CRFs) [5], recently proposed for matching 2D laser scans [6] and matching image features [7]. CRF-Matching is a probabilistic model able to jointly reason about the association of features. Here we extend CRF-Matching to reason in the 3D space about the association of data provided by a stereo camera system. We also propose the use of the Minimum Spanning Tree (MST) as graph structure for the CRF-Matching. This allows exact inference with no loss of accuracy [8], as compared, for instance, with loopy belief propagation for cyclic graphs, which is approximate. So far the learning stage in CRFs was done with full or partial manual labelling [9]. Our CRF-Matching algorithm carries out automatic labelling during the learning stage.

The basic idea is to exploit the efficiency of BoW for detecting revisited places in real-time. In order to keep the real time execution, only the result of BoW will be the input for CRF-Matching. CRF-Matching is a computationally demanding data association algorithm because it uses much more information than the BoW. Successful results of BoW filtered by the CRF-Matching will be the system output.

This paper is organised as follows: We begin with a discussion of the related work in Section II. We then provide a description on the loop detection with bag-of-words in Section III. We provide an overview of Conditional Random Fields and how to apply CRFs to our case in Section IV. Finally, we present in Section V experimental results on real data that demonstrate the improvement in robustness of our approach.

## II. Related work

There are different kinds of algorithms to solve the loop closing problem in SLAM, including those based on map or image features [10] and robot poses [11]. Appearance-based methods are becoming popular since cameras provide rich scene information and have become a common sensor in robotics. These methods focus on place recognition, and mainly use the bag-of-words representation [1], supported by some probabilistic framework [3]. On the issue of recognition of places perhaps the state of the art is the FAB-MAP [2], since it has proved very successful with a low proportion of false positives. We propose applying adaptive thresholds to the similarity between two scenes to improve the results yielded by the bag-of-words algorithm.

To avoid mismatches in these appearance-based approaches, some geometrical constraint is generally added in a next step. The epipolar geometry is the most common technique used to find consistent matches [12]. Here we present an algorithm based on CRF-Matching which achieves more robust results than the epipolar constraint.

CRF-Matching was introduced in [6] for loop closure detection by combining information from a 2D laser scanner with the information of texture from a monocular camera. The same framework is proposed in [7] to associate image features, using the 2D Delaunay triangulation as a graph

César Cadena, Dorian Gálvez-López, Juan D. Tardós and José Neira are with the Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza 50018, Spain. {ccadena, dorian, tardos, jneira}@unizar.es.

Fabio Ramos is with the ARC Centre of Excellence for Autonomous Systems, Australian Centre for Field Robotics, The University of Sydney, Sydney, NSW 2006, Australia. f.ramos@acfr.usyd.edu.au.

This research has been funded by the European Union under project RoboEarth FP7-ICT-248942, the Dirección General de Investigación of Spain under projects DPI2009-13710, DPI2009-07130 and the Ministerio de Educación (scholarship FPU-AP2008-02272).

structure. We extend this idea to associate 3D features, using the minimum spanning tree as the graph structure, combining appearance information with the metric information from a stereo camera system. In [8], it was shown that this graph structure properly encodes connections between the hidden variables and ensures global consistency in the object recognition task. Moreover, by using MST we can use exact inference algorithms.

There are other works that join image and geometrical data, such as [13] where an actuated laser scanner and a monocular camera are used. However, this system is not able to combine data from the two sensors, only camera information is used to detect loop closure events. In our system any type of sensor data can be smoothly combined for probabilistic inference without assumptions of independence. In the context of feature-based SLAM, in addition to loop closure, our system provides the data association of features to the observations and the probability of each individual association.

## III. Bag of Words

### A. Image representation

A visual bag-of-words [1] is a technique that represents an image as a numerical vector by quantising its salient local features. For this purpose, we use SURF features [14]. This technique entails an off-line stage that consists in clustering the image descriptor space (the 64-dimensional SURF space, in our case) into a fixed number $N$ of clusters. The centres of the resulting clusters are named *visual words*; after the clustering, a *visual vocabulary* is obtained. Now, a set of image features can be represented in the visual vocabulary by means of a vector $v$ of length $N$. For that, each feature is associated to its approximately closest visual word; then, each component $v_i$ is set to a value in accordance with the relevance of the $i$-th word in the vocabulary and the given set, or 0 if that word is not associated to any of the image descriptors. There are several approaches to measure the relevance of a word in a corpus [15]; in general, the more a word appears in the data used to create the visual vocabulary, the lower its relevance is. We use the term frequency – inverse document frequency (tf-idf) as proposed by [1]. The vector $v$ is the bag-of-words representation of the given set of image features.

This method is suitable for managing big amounts of images; moreover, [4] presents a hierarchical version which improves efficiency. In this version, the descriptor space clustering is done hierarchically, obtaining a visual vocabulary arranged in a tree structure, with a branch factor $k$ and $L$ depth levels. In this way, the comparisons for converting an image descriptor into a visual word only need to be done in a branch and not in the whole discretized space, reducing the search complexity to logarithmic. An own implementation of this data structure is used in this paper, with $k = 9$, $L = 6$ and the *kmeans*++ algorithm [16] as clustering function. This configuration yielded the best performance in both indoor and outdoor, and dynamic, datasets.

### B. Image similarity

Representing images as numerical vectors is very convenient since it allows performing really quick comparisons between images. There are several metrics to calculate the similarity between two image vectors. We use a modified version of the one proposed by [4]. Given two vectors $v$ and $w$, their similarity is measured as the score $s(v, w)$:

$$s(v,w) = 1 - \frac{1}{2}\left\| \frac{v}{||v||} - \frac{w}{||w||} \right\| \quad (1)$$

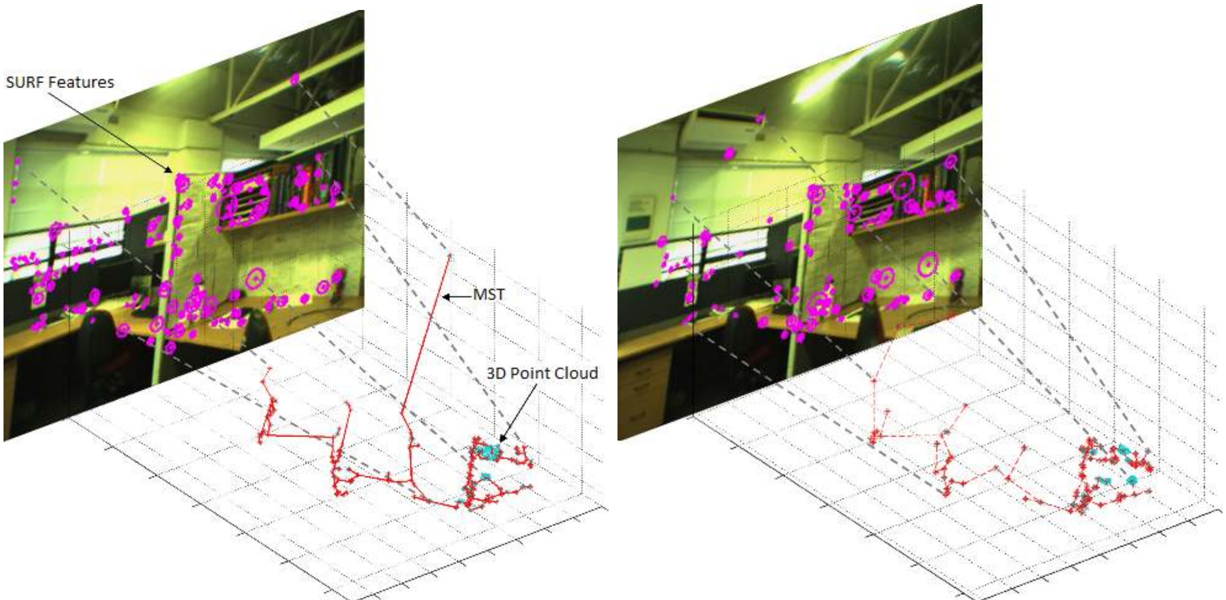
where $||.||$ stands for the $L_1$-norm. Note that this score is 0 when there is no similarity at all, and 1 when both vectors are the same.

This score is the only value used to set the similarity between two images at this stage. If the score between two images is not high enough to be considered the same scene, additional geometrical consistency is needed to make the decision, as in [1].

### C. Loop candidate detection

Our system takes an image at time $t$ from the stereo pair at one frame per second. The image is converted into a bag-of-words vector $v_t$, which is stored in a set $W$. At the same time, a structure (named *inverted file* [4]) is maintained to save in which images each visual word is present. The current image vector $v_t$ is compared against all the ones stored before in $W$. The complexity of this operation is linear in the number of stored vectors, but the inverted file makes it be very quick. The result is a list of matches $< v_t,\ w_{t'} >$, associated to their scores $s(v_t, w_{t'})$, where $w_{t'}$ are the vectors matched from $W$. Low score matches are discarded from this list, together with those matches which are too close in time. The score range depends on the number of features each image contains. For this reason, a score is considered low by comparing it to the maximum expected score for a certain image (denoted $\lambda_t$). Since our images are taken from a video sequence, we approximate $\lambda_t$ with $s(v_t, v_{t-1})$. If images separated by 1 second are not similar (e.g. if the robot is turning), this approximation is not reliable and $\lambda_t$ is small. Therefore, we remove the matches $< v_t,\ w_{t'} >$ with $\lambda_t < 0.1$ or whose score $s(v_t, w_{t'})$ does not achieve $\alpha^- \lambda_t$, where $\alpha^-$ is the minimum confidence expected for a loop closure candidate.

To detect loops, we impose a temporal constraint. A loop candidate between images at time $t$ and $t_0$ is detected if there are matches $< v_t,\ w_{t_0} >$, $< v_{t-1},\ w_{t_1} >$, ..., for a short time interval (set to 4 seconds), and the timestamps $t_0$, $t_1$, ..., are close (within 2 seconds). These temporal values are selected according to the frequency of our image sequences, and the expected reliability of BoW. Finally, the match $< v_t,\ w_{t_0} >$, with score $s(v_t, w_{t_0})$, is accepted as a loop candidate. If this score is high enough, the match is very likely to be correct, so that the candidate is accepted as a loop. However, mismatches can occur. CRF-Matching is used in the cases where the score alone is not sufficient to ensure loop closure. Scores greater than $\alpha^+ \lambda_t$ are accepted as loops, and those between $\alpha^- \lambda_t$ and $\alpha^+ \lambda_t$ are checked by the CRF-Matching stage, where $\alpha^+$ denotes the minimum confidence to trust



(a) Scene A

(b) Scene B

Fig. 1. On the left, for scene A, we show the right image from the stereo pair with the features obtained by the SURF extractor. From these image features, and the 3D information, we get the Minimum Spanning Tree (red) for building the graph used by the CRF. Also, we show the 3D point cloud (light blue) of each vertex in the tree. On the right, the same for the scene B. Here the minimum spanning tree is used to define the neighbourhood of each feature. The MST gives us an idea of the dependencies between features in one scene, and helps the consistency of the features association between scenes A and B.

candidates from BoW without further checking. We use $\alpha^-$ and $\alpha^+$ in order to keep efficiency. Since CRF-Matching is a more time-consuming algorithm, these thresholds allow us to skip that stage for those cases with little chance to match, or with high likelihood to be correct.

## IV. CRF-Matching

### A. Model definition

CRF-Matching is based on Conditional Random Fields, undirected graphical models developed for labelling sequence data [5]. Instead of relying on Bayes rule to estimate the distribution over hidden states $\mathbf{x}$ from observations $\mathbf{z}$, CRFs directly model $p(\mathbf{x}|\mathbf{z})$, the *conditional* distribution over the hidden variables given observations. Due to this structure, CRFs can handle arbitrary dependencies between the observations, which gives them substantial flexibility in using complex and overlapped attributes or observations.

The nodes in a CRF represent hidden states, denoted $\mathbf{x} = \langle \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n \rangle$, and observations, denoted $\mathbf{z}$. In our framework the hidden states correspond to all the possible associations between the $n$ features in scene A and the $m$ features in the scene B, i.e. $\mathbf{x}_i \in \{1, 2, \ldots, m+1\}$, where the additional state is the outlier state. Observations are provided by the sensors (e.g., 3D point cloud, appearance descriptors, or any combination of them). The nodes $\mathbf{x}_i$ along with the connectivity structure represented by the undirected graph define the conditional distribution $p(\mathbf{x}|\mathbf{z})$ over the hidden states $\mathbf{x}$. Let $\mathcal{C}$ be the set of cliques (fully connected subsets) in the graph of a CRF. Then, a CRF factorizes the conditional distribution into a product of *clique potentials* $\phi_c(\mathbf{z}, \mathbf{x}_c)$, where every $c \in \mathcal{C}$ is a clique in the graph, and $\mathbf{z}$ and $\mathbf{x}_c$ are the observed data and the hidden nodes in such clique. Clique potentials are functions that map variable configurations to non-negative numbers. Intuitively, a potential captures the “compatibility” among the variables in the clique: the larger a potential value, the more likely the configuration. Using the clique potential, the conditional distribution over hidden states is written as:

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{z}, \mathbf{x}_c) \qquad (2)$$

where $Z(\mathbf{z}) = \sum_{\mathbf{x}} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{z}, \mathbf{x}_c)$ is the normalizing partition function. The computation of this function can be exponential in the size of $\mathbf{x}$. Hence, exact inference is possible for a limited class of CRF models only, e.g. in tree-structured graphs.

Potentials $\phi_c(\mathbf{z}, \mathbf{x}_c)$ are described by log-linear combinations of *feature functions* $\mathbf{f}_c$, i.e., the conditional distribution (2) can be rewritten as:

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})} \exp\left\{ \sum_{c \in \mathcal{C}} \mathbf{w}_c^T \cdot \mathbf{f}_c(\mathbf{z}, \mathbf{x}_c) \right\} \qquad (3)$$

where $\mathbf{w}_c^T$ is the transpose of a weight vector, which represents the importance of different features for correctly identifying the hidden states. Weights can be learned from labeled training data.

### B. Inference

Inference in a CRF estimates the marginal distribution of each hidden variable $\mathbf{x}_i$, and can thus determine the

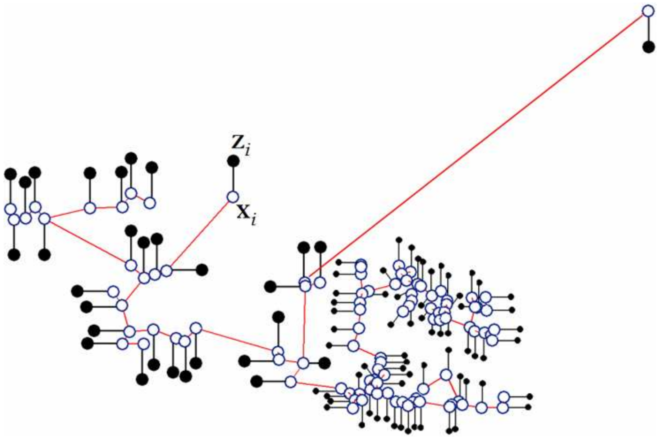


Fig. 2. The corresponding graphical representation of the CRF-Matching model from Fig. 1(a). The hidden state $\mathbf{x}_i$ corresponds to all the possible associations between the feature $i$ in the scene A and all the features in the scene B. The observations $\mathbf{z}_i$ correspond to shape or visual appearance information extracted from the scenes for the feature $i$.

most likely configuration of the hidden variables $\mathbf{x}$ (i.e., the maximum a posteriori, or MAP, estimation). Both tasks can be solved using *belief propagation* (BP) [17], which works by transmitting messages containing beliefs through the graph structure of the model. Each node sends messages to its neighbours based on messages it receives and the clique potentials. BP generates exact results in graphs with no loops, such as trees or polytrees. To create the graph structure we use a minimum spanning tree over the 3D coordinates of the SURF features extracted from the right image in the stereo pair (see Fig. 2).

### C. Parameter learning

The goal of parameter learning is to determine the weights of the feature functions used in the conditional likelihood (3). CRFs learn these weights discriminatively by maximising the conditional likelihood of labeled training data. We resort to maximising the *pseudo-likelihood* of the training data, which is given by the product of all local likelihoods $p(\mathbf{x}_i|\mathrm{MB}(\mathbf{x}_i))$; $\mathrm{MB}(\mathbf{x}_i)$ is the Markov Blanket of variable $\mathbf{x}_i$, which contains the immediate neighbours of $\mathbf{x}_i$ in the CRF graph. Optimisation of this pseudo-likelihood is performed by minimising the negative of its log, resulting in the following objective function:

$$L(\mathbf{w}) = -\sum_{i=1}^{n} \log p(\mathbf{x}_i|\mathrm{MB}(\mathbf{x}_i),\mathbf{w}) + \frac{\mathbf{w}^T\mathbf{w}}{2\sigma_{\mathbf{w}}^2} \tag{4}$$

The rightmost term in (4) serves as a zero-mean Gaussian prior, with variance $\sigma_{\mathbf{w}}^2$, on each component of the weight vector.

The training data is labeled from the best rigid-body transformation using RANSAC after a SURF matching [18] of two consecutive scenes (see Section V). In this way we avoid the burden of manually labelling data.

### D. Feature description

CRF-matching can employ arbitrary local features to describe shape, images properties, or any particular aspect of the data. Since our focus is on associating features from two 3D scenes, our features describe *differences* between shape and appearance of the features. The local features we use are the following:

**Shape difference**: These features capture how much the local shape of dense stereo data differs for each possible association. We use the geodesic, PCA and curvature distance.

The *geodesic distance*, defined as the sum of Euclidean distances between points in the minimum spanning tree, provides shape information of a scene. It can be calculated for different neighbourhoods representing local or long-term shape information. Given points $z_{A,i}$, $z_{B,j}$ and a neighbourhood $k$, the geodesic distance feature is computed as:

$$\mathbf{f}_{geo}(i,j,k,z_A,z_B) = \frac{\left\| \sum_{l=i}^{i+k-1} \|z_{A,l+1} - z_{A,l}\| - \sum_{l=j}^{j+k-1} \|z_{B,l+1} - z_{B,l}\| \right\|}{\sigma} \tag{5}$$

where $i$ and $j$ correspond to the hidden state $\mathbf{x}_i$ that associate the feature $i$ of the scene A with the feature $j$ of the scene B. The neighbourhood $k$ of $\mathbf{x}_i$ in the graph corresponds to all the nodes separated $k$ nodes from $\mathbf{x}_i$. In our implementation, this feature is computed for $k \in \{1,2,3\}$. A similar feature is used to match 3D laser scans in [19]. The parameter $\sigma$ controls the scale of the corresponding distance; the same in the subsequent equations.

We also use Principal Component Analysis over the dense 3D point cloud that is contained within a radius given by the scale obtained by the SURF extractor for each node in the graph, light blue points in Fig. 1. Then *PCA distance* is computed as the absolute difference between principal components of a dense point cloud $z_{A,i}^{pca}$ in scene $A$ and $z_{B,j}^{pca}$ in scene $B$:

$$\mathbf{f}_{PCA}(i,j,z_A^{pca},z_B^{pca}) = \frac{\left| z_{A,i}^{pca} - z_{B,j}^{pca} \right|}{\sigma} \tag{6}$$

Another way to consider local shape is by computing the difference between the curvatures of the dense point clouds. This feature is computed as:

$$\mathbf{f}_{curv}(i,j,z_A^c,z_B^c) = \frac{\left\| z_{A,i}^c - z_{B,j}^c \right\|}{\sigma} \tag{7}$$

where $z^c = \frac{3s_3}{s_1+s_2+s_3}$, and $s_1 \geq s_2 \geq s_3$ are the *singular values* of the point cloud of each node.

**Visual appearance**: These features capture how much the local appearance from the points in the image differs for each possible association. We use the *SURF distance*. This feature calculates the Euclidean distance between the descriptor vectors for each possible association:

$$\mathbf{f}_{SURF}(i,j,z_A^{descr},z_B^{descr}) = \frac{\left\| z_{A,i}^{descr} - z_{B,j}^{descr} \right\|}{\sigma} \tag{8}$$

All previous features described are unary, in that they only depend on a single hidden state $i$ in scene $A$ (indices $j$ and $k$

TABLE I

INDOOR DATASET, $\alpha^- = 0$, $\alpha^+ = \infty$

| | BoW | BoW + EC | BoW + CRF-Matching |
|---|---|---|---|
| True positives | 232 | 216 | 102 |
| False positives | 284 | 165 | 4 |
| True negatives | 1232 | 1351 | 1512 |
| False negatives | 8 | 24 | 138 |
| Precision | 44.96% | 56.69% | 96.23% |
| Recall | 96.67% | 90.00% | 42.50% |

TABLE II

OUTDOOR DATASET, $\alpha^- = 0$, $\alpha^+ = \infty$

| | BoW | BoW + EC | BoW + CRF-Matching |
|---|---|---|---|
| True positives | 133 | 118 | 9 |
| False positives | 106 | 40 | 0 |
| True negatives | 1811 | 1877 | 1917 |
| False negatives | 229 | 244 | 353 |
| Precision | 55.65% | 74.68% | 100% |
| Recall | 36.74% | 32.60% | 2.49% |

TABLE III

MIXED DATASET, $\alpha^- = 0$, $\alpha^+ = \infty$

| | BoW | BoW + EC | BoW + CRF-Matching |
|---|---|---|---|
| True positives | 211 | 200 | 60 |
| False positives | 57 | 28 | 1 |
| True negatives | 1767 | 1796 | 1823 |
| False negatives | 115 | 126 | 266 |
| Precision | 78.73% | 87.72% | 95.08% |
| Recall | 64.72% | 61.35% | 17.79% |

TABLE IV

INDOOR DATASET, $\alpha^- = 15\%$, $\alpha^+ = 60\%$

| | BoW | BoW + EC | BoW + CRF-Matching |
|---|---|---|---|
| True positives | 224 | 208 | 158 |
| False positives | 221 | 161 | 4 |
| True negatives | 1295 | 1355 | 1512 |
| False negatives | 16 | 32 | 82 |
| Precision | 50.34% | 56.37% | 97.53% |
| Recall | 93.33% | 86.67% | 65.83% |

TABLE V

OUTDOOR DATASET, $\alpha^- = 15\%$, $\alpha^+ = 60\%$

| | BoW | BoW + EC | BoW + CRF-Matching |
|---|---|---|---|
| True positives | 114 | 84 | 18 |
| False positives | 43 | 6 | 0 |
| True negatives | 1874 | 1911 | 1917 |
| False negatives | 248 | 278 | 344 |
| Precision | 72.61% | 93.33% | 100% |
| Recall | 31.49% | 23.20% | 4.97% |

TABLE VI

MIXED DATASET, $\alpha^- = 15\%$, $\alpha^+ = 60\%$

| | BoW | BoW + EC | BoW + CRF-Matching |
|---|---|---|---|
| True positives | 208 | 165 | 113 |
| False positives | 25 | 4 | 0 |
| True negatives | 1799 | 1820 | 1824 |
| False negatives | 118 | 161 | 213 |
| Precision | 89.27% | 97.63% | 100% |
| Recall | 63.80% | 50.61% | 34.66% |

in the features denote nodes in scene B and neighbourhood size). In order to generate mutually *consistent* associations it is necessary to define features, over the cliques, that relate the hidden states in the CRF to each other.

**Pairwise distance**: This feature measures the consistency between the associations of *two* hidden states $\mathbf{x}_i$ and $\mathbf{x}_j$ and observations $z_{A,i}$, $z_{A,j}$ from scene $A$ and multiple observations $z_{B,k}$ and $z_{B,l}$ in scene $B$:

$$\mathbf{f}_{pair}(i,j,k,l,z_A,z_B) = \frac{\left|\|z_{A,i} - z_{A,j}\| - \|z_{B,k} - z_{B,l}\|\right|}{\sigma} \qquad (9)$$

## E. Loop closure acceptance

We use the CRF-Matching stage over the loop closing candidates provided by the BoW stage. Then, we compute the negative log-likelihood ($\Lambda$) from the MAP associations between the scene in time $t$, against the loop closing candidate in time $t'$, $\Lambda_{t,t'}$, and the scene in $t-1$, $\Lambda_{t,t-1}$. We accept the loop closing only if $\Lambda_{t,t'} \leq \Lambda_{t,t-1}$, where $\Lambda_{t,:}$ is normalised by the number $n$ of graph's nodes of the scene in time $t$.

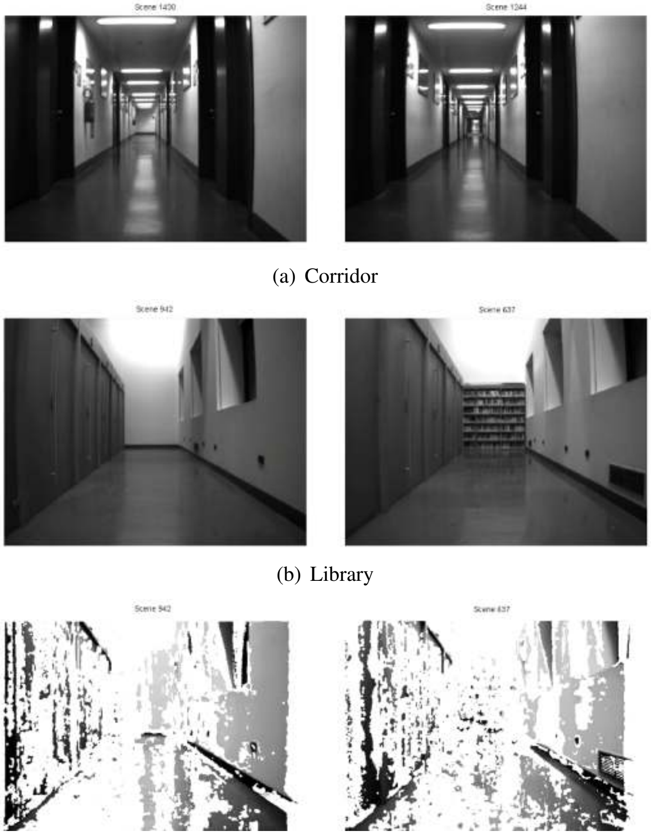
# V. EXPERIMENTS

We have evaluated our system with the public datasets from the RAWSEEDS Project[1]. The data were collected by a robotic platform in static and dynamic indoor, outdoor and mixed environments. We have used the data corresponding to the Stereo Vision System with 18cm of baseline. These are b/w images (640x480 px) taken at 15 fps. We used 200 images uniformly distributed in time, from a static mixed dataset, for training the vocabulary for BoW and for learning the weights for CRF-Matching. Afterwards, we tested the whole system in three datasets: static indoor, static outdoor and dynamic mixed. The four datasets were collected on different dates and in two different campus. Refer to the RAWSEEDS Project for more details.

In order to learn the weights for the CRF-Matching, we obtained the SURF features from the right image in the stereo system and computed their 3D coordinates. Then, we ran a RANSAC algorithm over the rigid-body transformation between the scene at time $t$ and the scene at time $t-\delta_t$. The results from RANSAC were our labels. Since the stereo system has high noise in the dense 3D information, we selected $\delta_t = 1/15s$. Thus, we obtained a reliable enough labelling for the training. Although this automatic labelling can return some outliers, the learning algorithm has demonstrated being robust in their presence. The weights obtained suggest that the most relevant features in CRF-Matching are $\mathbf{f}_{SURF}$ and $\mathbf{f}_{pair}$. The smallest weights are given to the third value of $\mathbf{f}_{PCA}$ and to $\mathbf{f}_{curv}$, but even so, these features play a valuable role in the effectiveness of the algorithm.

The online system can run at 1 fps. Extracting SURF features is usually done in $0.21s$ per image, whereas running the BoW algorithm and maintaining the inverted file takes $28ms$ on average. Per each candidate evaluation, the CRF

[1] RAWSEEDS is an European FP6 Project, `http://www.rawseeds.org`



(a) Corridor

(b) Library

(c) Library, zones with 3D information

Fig. 3. Two of the challenging cases that both the epipolar constraint and the CRF-Matching mismatch in the indoor dataset.

stage takes, on average, $0.3s$ computing the features and $0.15s$ in the inference process.

### A. CRF-Matching vs Epipolar constraint

We have compared the CRF-Matching algorithm we propose against a common approach for rejecting outliers, based on epipolar geometry. This epipolar constraint consists in calculating the fundamental matrix (by using the 8-point algorithm [20]) between the images matched by a loop closure candidate. This checking is passed if a well conditioned fundamental matrix can be obtained.

In order to show the improvements of our CRF stage over the epipolar constraint, we have first computed the loop closing candidates from the BoW with $\alpha^- = 0$ and $\alpha^+ = \infty$ (i.e. BoW does not filter out any candidate). Then, we have computed the candidates accepted by the epipolar constraint and the CRF-Matching. The results are shown in Tables I, II and III. In every case, we can see that the precision of the CRF-Matching is better than the one of the epipolar constraint. On the other hand, the recall is punished by the strictness of the CRF-Matching.

In the static indoor dataset, Table I, the false positives that result after the CRF-Matching are due to perceptual aliasing, as we show in Fig. 3. Note that the scene in Fig. 3(b) could be solved with another baseline in the stereo system. In Fig. 3(c) we show the zones where we have 3D information for the scene in Fig. 3(b) with the baseline given. With a greater baseline there would be reliable 3D information of the book shelf at the end of the hall.

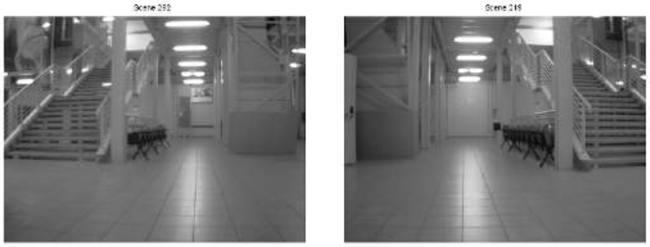


Fig. 4. The mismatch that both the CRF-Matching and the epipolar constraint accept in the dynamic mixed dataset. Note that these scenes seem specular images.

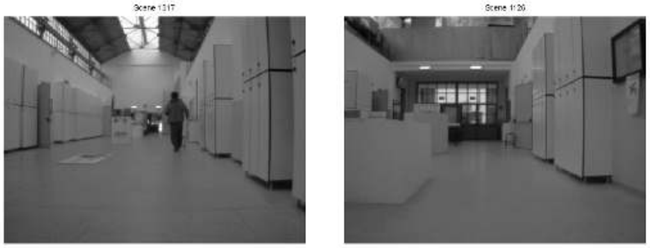


Fig. 5. The epipolar constraint mismatches these scenes in the dynamic mixed dataset. CRF-Matching, on the contrary, correctly rejects it.

The static outdoor dataset, Table II, does not present false positives after the CRF-Matching, but it has a low recall, $2.49\%$. This also occurs because the baseline of the stereo system used is not able to capture enough 3D information. Therefore, the SURF features of distant objects are hardly included in the graph.

The only false positive obtained after the CRF-Matching in the dynamic mixed dataset, Table III, is a case of specular symmetry (Fig. 4). In contrast to Fig. 3(c), these scenes have enough 3D information but the CRF-Matching fails because all the geometric information used is relative, not able to discriminate between specular scenes. In this case, both CRF-Matching and the epipolar constraint are not able to detect the mismatch without some additional checking. However, CRF-Matching succeeds in other cases where the epipolar constraint does not, such as the scenes in Fig. 5.

### B. Our system

In the previous section we compared the effectiveness of our method against an epipolar constraint. We have also checked how these methods work in our whole system. For that, we selected the working values $\alpha^- = 15\%$ and $\alpha^+ = 60\%$. We have set these parameters by observing their effect on the precision-recall curves achieved by the BoW algorithm on its own in the datasets tested (see Fig. 6). Since these datasets are fairly heterogeneous, we think these values can work well in many situations. It might depend on the datasets and the vocabulary size, though.

Tables IV, V, VI show the results when applying $\alpha^-$ and $\alpha^+$. If we compare the *BoW* column in Tables I, II and III

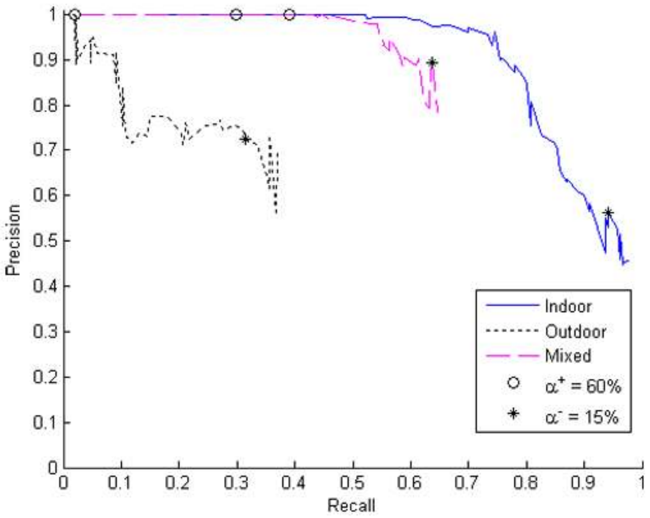


Fig. 6. Precision and Recall of BoW in each dataset, along with the working values of $\alpha^+$ and $\alpha^-$.

with that in Tables IV, V and VI, we see that the increase in precision is always greater than the decrease in recall. This indicates that thresholds $\alpha^-$ and $\alpha^+$ improve efficacy besides efficiency. In addition, if we look at false positives in Tables III and VI, we see that the mismatch shown in Fig. 4 disappears. This is due to $\alpha^-$ along with the temporal consistency imposed by the BoW algorithm. We can also check how well our proposed combination of BoW and CRF-Matching works: both precision and recall improve in the *BoW + CRF-Matching* column in all the cases when using $\alpha^-$ and $\alpha^+$.

The precision of our system is $100\%$ in the outdoor and mixed datasets, and $97.53\%$ in the indoor dataset due to those cases shown in the Fig. 3. If we check the chart in Fig. 6, we see than by adjusting $\alpha^-$ to a certain value, we would eliminate the false positives in the indoor dataset. This is generally true for any dataset; however, this would involve calculating a specific threshold for every dataset. In addition, increasing $\alpha^-$ always involves a trade off with recall. Our system exhibits a good recall in static indoor and dynamic mixed datasets, $65.83\%$ and $34.66\%$ respectively. And, although the recall is low in the static outdoor dataset ($4.97\%$), our system detects the biggest loop closures (see Fig.7(b)).

In Fig. 7 we can see that our system with CRF-Matching removes most of the mismatches made by BoW. The indoor dataset (Fig. 7(a)) is especially challenging, since it presents several similar-looking corridors, so there is perceptual aliasing. In this case, the epipolar constraint can only reject a few cases, whereas the CRF-Matching algorithm discards most of them, only those from Fig. 3 pass. In total, our system with CRF-Matching yields 4 mismatches in three datasets with hundreds of meters, as compared the use of epipolar constraint, that result in 171.

Since our system improves the precision, with a lot of the loop closings detected, we can say that this outperforms the epipolar constraint.

## VI. Conclusions and Future work

We have presented a system that uses together a bag-of-words algorithm and conditional random fields to robustly solve the place recognition problem. Our results have shown that the CRF-Matching algorithm outperforms the classical epipolar constraint to verify loop candidates, especially under perceptual aliasing conditions. CRF-Matching is more robust since it uses 3D information (either provided by stereo vision, range scanners, etc), whereas epipolar geometry can be applied on a single image. However, CRF-Matching is also able to fuse any other kind of information, such as image colour, with ease. In addition, our whole system has proved very successful on several indoor, outdoor and dynamic environments.

Immediate future work consists in extending our system for robust cooperative multi-robot SLAM. In the longer run, we believe that CRF-Matching alone can be very robust for place recognition, and furhermore it is not specific of visual sensors like the bag-of-words technique. Reducing its computational cost, currently linear, will allow the technique to be used in the relocalisation problem, and also to tackle other field applications of interest where 3D sensor information can be obtained, such as underwater applications using Synthetic Aperture Sonar.

## References

[1] 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*, vol. 2, Oct. 2003, pp. 1470–1477.

[2] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[3] A. Angeli, D. Filliat, S. Doncieux, and J. 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, pp. 1027–1037, 2008.

[4] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, 2006, pp. 2161–2168.

[5] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001, pp. 282–289. [Online]. Available: citeseer.ist.psu.edu/lafferty01conditional.html

[6] F. Ramos, D. Fox, and H. Durrant-Whyte, "CRF-Matching: Conditional Random Fields for Feature-Based Scan Matching," in *Robotics: Science and Systems (RSS)*, 2007.

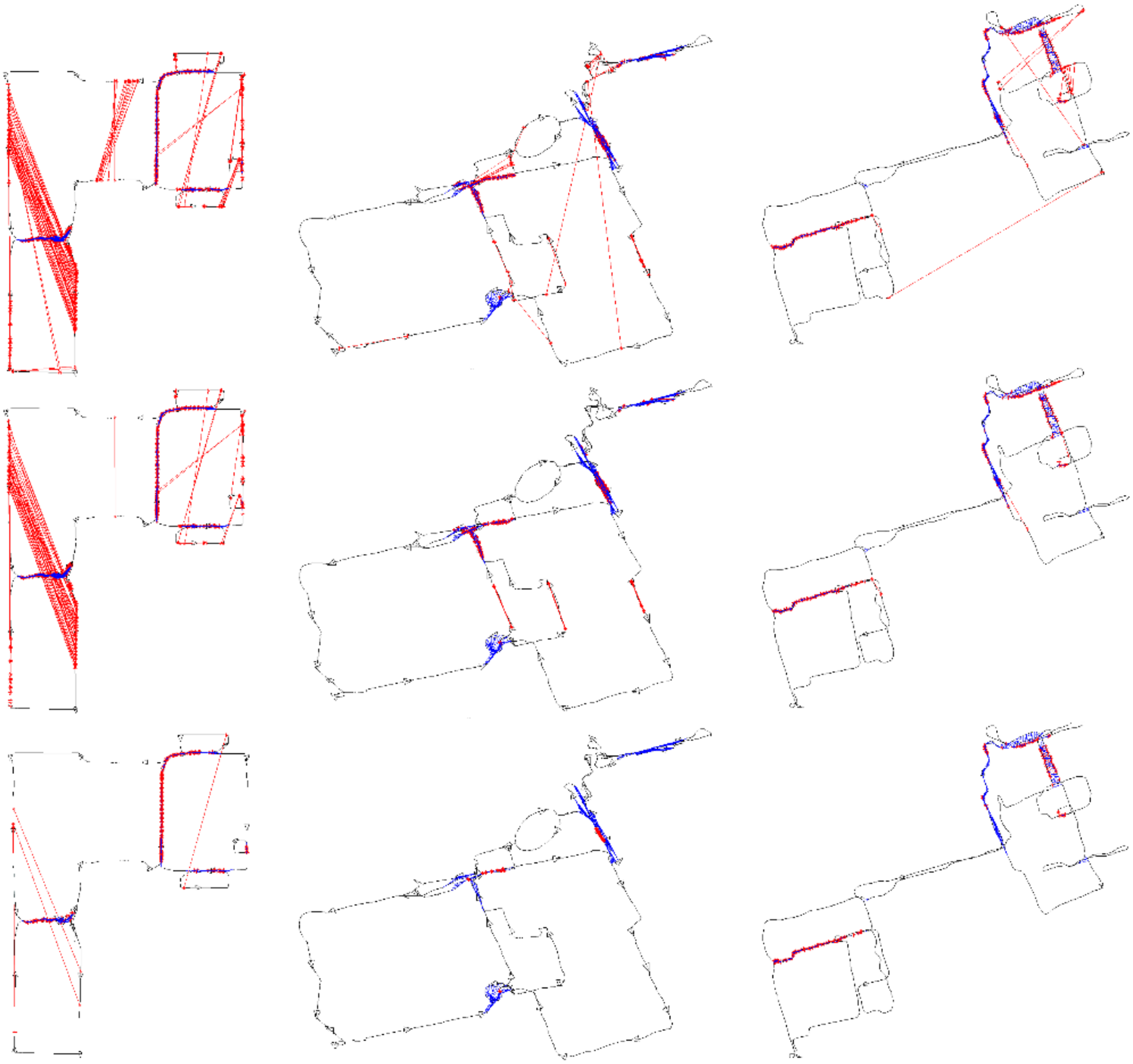
[7] F. Ramos, M. W. Kadous, and D. Fox, "Learning to associate image features with CRF-Matching," in *ISER*, 2008, pp. 505–514.

[8] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell, "Hidden conditional random fields," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 10, pp. 1848–1852, Oct. 2007.

[9] B. Douillard, D. Fox, and F. Ramos, "Laser and vision based outdoor object mapping," in *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland, June 2008. [Online]. Available: www.roboticsproceedings.org/rss04/p2.pdf

[10] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A comparison of loop closing techniques in monocular slam," *Robotics and Autonomous Systems*, 2009.

[11] E. Olson, "Recognizing places using spectrally clustered local matches," *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1157–1172, December 2009.



(a) Indoor

(b) Outdoor

(c) Mixed

Fig. 7. Loops detected by each of the methods in each dataset, with $\alpha^{-} = 15\%$, $\alpha^{+} = 60\%$. From top to bottom: BoW, BoW + epipolar constraint, BoW + CRF-Matching. Black lines and triangles denote the trajectory of the robot; deep blue lines, actual loops, and light red lines, loops detected. Note that CRF-Matching does not have false positives except for 4 cases in the indoor dataset, due to perceptual aliasing (Fig. 3). Even in this dataset, results are better than when using the epipolar constraint.

[12] D. Nister, “An efficient solution to the five-point relative pose problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 756–777, 2004.

[13] P. Newman, D. Cole, and K. L. Ho, “Outdoor SLAM using visual appearance and laser ranging,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando Florida USA, May 2006.

[14] T. T. Herbert Bay and L. V. Gool, “SURF: Speeded up robust features,” in *Proceedings of the 9th European Conference on Computer Vision*, vol. 3951, no. 1. Springer LNCS, 2006, pp. 404–417.

[15] J. Yang, Y. Jiang, A. Hauptmann, and C. Ngo, “Evaluating bag-of-visual-words representations in scene classification,” in *Proceedings of the international workshop on Workshop on multimedia information retrieval*. ACM, 2007, p. 206.

[16] D. Arthur and S. Vassilvitskii, “k-means++: the advantages of careful seeding,” in *SODA ’07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[17] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[18] E. Olson, “Robust and efficient robotic mapping,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.

[19] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, “SCAPE: shape completion and animation of people,” *ACM Trans. Graph.*, vol. 24, no. 3, pp. 408–416, 2005.

[20] R. Hartley, “In defense of the eight-point algorithm,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 6, pp. 580–593, 1997.