

# Fast Hybrid Relocation in Large Scale Metric-Topologic-Semantic Map

Romain Drouilly<sup>1,2</sup>, Patrick Rives<sup>1</sup>, Benoit Morisset<sup>2</sup>

**Abstract**—Navigation in large scale environments is challenging because it requires accurate local map and global relocation ability. We present a new hybrid metric-topological-semantic map structure, called *MTS-map*, that allows a fine metric-based navigation and fast coarse query-based localisation. It consists of local sub-maps connected through two topological layers at metric and semantic levels. Semantic information is used to build concise local graph-based descriptions of sub-maps. We propose a robust and efficient algorithm that relies on *MTS-map* structure and semantic description of sub-maps to relocate very fast. We combine the discriminative power of semantics with the robustness of an interpretation tree to compare the graphs very fast and outperform state-of-the-art-techniques. The proposed approach is tested on a challenging dataset composed of more than 13000 real world images where we demonstrate the ability to relocate within 0.12ms.

## I. INTRODUCTION

Although it has been largely studied in the last decade, autonomous navigation remains a challenging issue, particularly in complex large scale environments. In this paper we address the problem of building navigation-oriented maps capable of dealing with different localization levels, from coarse to fine. Map-based navigation requires the robot to be able to request efficiently the content of the map at large scale to retrieve its position and simultaneously to infer the position of local objects around it. The map needs to be precise locally but lightweight at large scale. However in most of 3D maps, information density is homogeneous in space yielding to a compromise between precision of local model and size of the environment to model. This kind of representation intrinsically limits local quality of the model or reduces its scale-up capability. In this work, we use a more convenient map structure. It consists of a graph whose nodes are local sub-maps built from ego-centred spherical views of the environment, previously introduced in [1].

Navigation is further a product of environment understanding and planning, besides local metrical precision. A map is a cognitive representation of the world and the robot is able to reason only about concepts encoded within it. The more complex are these concepts the more "intelligent" could be its behaviour. Therefore intelligent navigation needs the map to contain abstraction layers to represent higher level concepts than geometry and color. Toward this goal topological mapping has early been considered of interest [2], [3], capturing the environment accessibility properties and allowing navigation in complex large scale environments

[4]. Semantic mapping is only recently receiving significant attention. It provides a powerful way to enrich the cognitive model of the world and thereby being of interest for navigation. However, despite a notable amount of work about outdoor scene parsing, the use of semantics for outdoor navigation has been poorly studied. Many mapping strategies rely on place classification or landmarks like doors to infer robot's position [5]. But localisation is only possible if object classes are strongly related to particular places, which is not the case outdoors. Additionally the place concept is hard to define for most of outdoor environments as these scenes are not structured enough to allow unambiguous delimitations between different areas.

We propose three main contributions to deal with those problems: a new 3D hybrid map structure designed for navigation purposes, a new framework to extract semantic information and an efficient algorithm to request the content of the map in a human-friendly way. All these improvements provide the robot both with precise local representation and fast global content request ability.

The rest of this paper is organized as follows: related works for space modelling and relocation in large databases are discussed in section II, *MTS-map* architecture is presented in section III followed by scene parsing results. Then the content request problem is treated in section IV before wrapping up with experimental results in section V and conclusion in section VI.

## II. RELATED WORK

### A. Hybrid mapping

Semantic mapping is an active research domain for the last years and many semantic representations exist in the literature. A semantic map model has been proposed in [6] where objects are grouped along two dimensions - semantic and spatial. Objects clustering along semantic dimension allows to capture place label where place is defined as group of objects. Grouping objects in clusters of increasing size provides meaning to the global scene. A multi-layers map is proposed in [7]. It is constructed in a semi-supervised way. The first three levels are composed of metric, navigation and topological maps. The last level is the semantic map that integrates acquired, asserted inferred and innate conceptual-ontological knowledge. A 3D extension of the well-known constellation model is presented in [8]. Here again object is the basic unit of representation for semantic labelling of place. Despite their interest these approaches are difficult to adapt to outdoor environments because they rely on the concept of place that is not well defined outdoors. Other methods do not directly rely on this concept. The 3D

\*This work was supported by ECA Robotics

<sup>1</sup>Authors are with INRIA Sophia-Antipolis, France  
romain.drouilly@inria.fr,  
patrick.rives@inria.fr

<sup>2</sup>Authors are with ECA Robotics, bmo@eca.fr

semantic map presented in [9] is defined as a map containing both metric information and labels of recognised classes. Prior knowledge and object models are needed for scene interpretation and object recognition. More recently [10] define a semantic SLAM approach that allows to build a map based on previously known object models. If they perform well indoors these works are not easily transferrable to outdoor environments. These models rely on object recognition and require to have a model of every object which is not easily tractable in large scale environments.

### B. Content Request

Relocation can be formulated as a content request problem: given the current observation, we ask the database to provide the corresponding position. Vision-based localization is studied for a long time and the use of omni-images dates back to early 90's [11]. Most of the modern techniques decompose to three steps: first, interest points are extracted and descriptors computed. Then descriptors between two images are matched. Finally outliers are rejected. In well-known Bag-Of-Words (BoW) methods, tree structure is commonly used to organize the search and speed up the comparison process. Many variations of BoW algorithm exist. We may cite [12] that uses feature context to improve their discriminant power. The idea is to select good features candidates to match in order to avoid perceptual aliasing. Other recent BoW methods offer good performances in image retrieval using different strategies. A tree structured Bayesian network is used in [13] to capture words co-occurrence in images. A compact vocabulary is created in [14] through discretization of a binary descriptor space. Despite undeniable efficiency, these algorithms have the drawback of providing a low-level description of the scene which is not human-friendly and makes it useless for human-robot cooperation.

## III. HYBRID METRIC-TOPOLOGICAL-SEMANTIC MAP ARCHITECTURE

In this section, we propose a new hybrid metric-topological-semantic map structure called *MTS map*. The map architecture is detailed below and illustrated in fig1.

### A. Map Architecture

*MTS-map* consists of 3-layered local sub-maps globally connected to each other in a graph structure through the use of a dense visual odometry method, first introduced in [1]. The bottom layer of each sub-map is an ego-centred RGBD spherical view of the environment acquired with a multi-cameras system [15]. As close as possible to sensor measurements, it provides a realistic representation of the local environment. The second layer presents a first level of abstraction with densely labelled spherical images, the "label layer". There is a direct correspondence between pixels of these two layers. At the upper stage lies the semantic-graph layer  $G_o$  which provides a compact and robust high-level description of the viewpoint. Nodes are the different regions semantically consistent in the labelled image and edges represent the connection between these regions. A label is

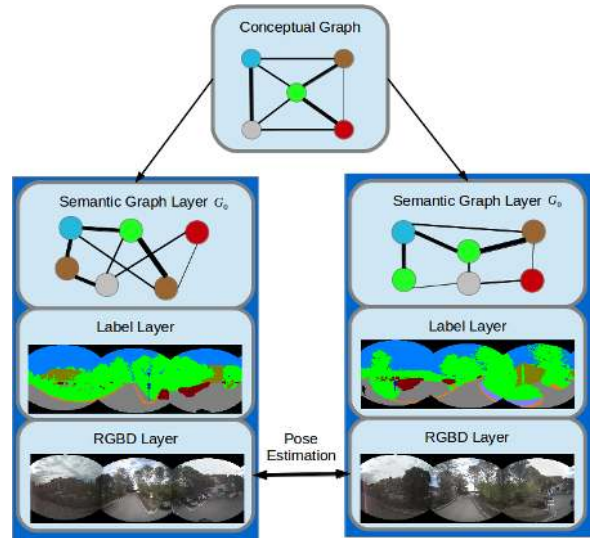


Fig. 1. MTS-map Architecture. Dark blue rectangles correspond to local sub-maps and light blue rounded rectangles to different layers.

attached to every node together with the size of the area, measured in pixels and its eccentricity represented by ratio of length and the width of the shape. Edges are weighted by the number of neighbouring pixels of two objects. All these layers constitute the RGBDL sphere, where "L" stands for Label. At the global scale every RGBDL sphere is globally referenced in a tree structure that clusters spheres according to class presence and class occurrence. Finally, atop of all sub-maps is the conceptual layer, defined as a non-spatial map which characterizes strength of relations between classes, generalized from  $G_o$  graphs.

### B. Scene Parsing

In this part we propose a framework to extract semantic information from spherical images. It should be noted that our work consists of separated building blocks and the localization step is independent of the algorithm used to label images.

1) *Local Features and Random Forest Classification*: The first step of the classification process uses *Random Forest* (RF) [16] to estimate classes distribution. A Random Forest is a set of  $T$  Decision Trees that achieves good classification rate by averaging prediction over all leaves  $L$  of all trees:

$$P(c|L) = \frac{1}{T} \sum_{i=1}^T P(c|l_i)$$

The model construction complexity is approximately of  $O(T(mn \log(n)))$  where  $n$  is the number of instances in the training data,  $m$  the vectors size. Provided they are correctly trained, RF has comparable performance with multi-class SVM with a reduced training and testing costs [17] that make them popular in computer vision. Moreover Random Forest has deeper architecture than Decision Tree or other well-known classification algorithm which makes it better able to generalize to variations not encountered in the training data [18].

Each Decision Tree is trained on a reduced subset of input data randomly chosen with replacement. Then each node is split using the best split among a subset of variables randomly chosen. The Decision Tree produces prediction by recursively branching left or right at each node until a leaf is reached. Due to classes imbalance in input data, prior preference for some classes can affect results. For that reason we weigh each training sample proportionally to the inverse class frequency.

To achieve good robustness to changes in orientation and scale, the feature vectors use SIFT descriptor computed densely on the gray scale image and augmented with color information computed on normalized RGB images.

2) *Spatio-Temporal consistency with CRF*: Random Forest produces accurate results but fails to catch contextual information at large scale. To capture global structure of the scene a common solution is to embed first stage prediction results into a probabilistic graphical model [5].

However applying classifier on single images results in practice in twinkling classification. To enforce temporal consistency large graphical models can be built among consecutive images to propagate labels [19] and [20]. The drawback of these methods is the complexity of the graph that can reach billions of edges. Other methods [21] use optical flow to propagate labels but need to previously learn similarity between pixels.

In this section, we present a way to simultaneously embed in the CRF the temporal and spatial context without the need to increase the number of edges in the CRF. We use the CRF architecture and efficient MAP inference algorithm presented in [22]. Fully connected pairwise CRF model is briefly reviewed here. Let  $X = \{x_1, \dots, x_n\}$  and  $Y = \{y_1, \dots, y_m\}$  be sets of random variables corresponding respectively to observations and labels. A CRF is an undirected graph  $G$  whose node correspond to  $X \cup Y$  and that encodes a conditional distribution as follows:

$$P(Y|X) = \frac{1}{Z(X)} \exp\left\{-\sum_{c \in \mathcal{C}_g} \phi_c(Y_c|X)\right\}$$

with  $\mathcal{C}_g$  the cliques of  $G$ ,  $\phi_c$  the induced potential and  $Z(X)$  a normalization factor.

In the fully connected pairwise CRF model the Gibbs energy [22] of a labelling  $y$  is:

$$E(y) = \sum_i \psi_u(y_i) + \sum_{i < j} \psi_c(y_i, y_j)$$

where  $\psi_c(y_i)$  denotes  $\phi(y_i|X)$ ,  $\psi_u$  is the unary potential and  $\psi_c$  the pairwise potential.

To enforce temporal consistency we accumulate Random Forest predictions from neighbours of the current view so that the unary potential  $\psi_c(y_i)$  takes the form :

$$\psi_u(y_i) = \alpha \sum_{n \in \mathcal{N}} \psi_n(y_i)$$

where  $\mathcal{N}$  is the neighbourhood of sphere  $i$  and  $\alpha$  is a normalization factor. Predictions are accumulated by projection of neighbours prediction on the current one using odometry.

### C. Scene Parsing Results

We evaluate our labelling framework on two datasets: CamVid and our INRIA dataset. Due to the lack of other dataset providing panoramic RGBD images fully annotated, we first apply our algorithm frame by frame embedding only spatial information in the CRF. Then we study the temporal consistency improvement on our dataset. CRF parameters are tuned by 2fold cross-validation on CamVid and 5fold cross validation on INRIA dataset. All experiments were performed using an Intel i7-3840QM CPU at 2.80GHz. All programs are single-threaded.

a) *INRIA Dataset*: consists of more than 13000 high resolution <sup>1</sup> panoramic images taken along a 1.6km pathway in a outdoor environment with forest and building areas. There are 9 classes corresponding to tree, sky, road, signs, sidewalk, ground signs, building, car, others. We manually label a subset of all images randomly chosen in the dataset. The training time for Random forest is 58 minutes and for CRF is 43 minutes. The mean prediction time is 2.5s for Random Forest and 3.5s for CRF.

b) *CamVid dataset*: <sup>2</sup> consists of several 960x720 video sequences taken in a highly dynamic street environment and labelled at 1Hz. There are 32 classes in the dataset. We use two sequences: 01TP sequence that lasts 2:04minutes and 06R0 sequence that lasts a 1:41minutes. We use the first half of the sequence as training set and the second for test. The training time for RF is 1h09 and for CRF is 48minutes. Prediction time is 1.5s for Random Forest and 3.1s for CRF.

c) *Performance measurement*: Two standard measures for multi-class classification are reported: the overall percentage of correctly classified pixels denoted as *global* and the unbalanced average per-class accuracy denoted as *average* and defined as  $\frac{1}{N} \sum_{k=1}^N \frac{t_k}{n_k}$  where  $N$  is the number of classes,  $t$  the number of pixels correctly classified and  $n$  the number of annotated pixels of the  $k^{\text{th}}$  class.

d) *Results*: Results for the frame-by-frame labelling are presented in table 1 and illustrated in figure 2 for INRIA dataset and in figure 3 for CamVid. As comparisons only make sense if we compare similar measures over similar datasets, we compare our results with those of [21]. Our algorithm reaches near state-of-the-art performances for global per-pixel accuracy and outperforms [21] for average per-class accuracy. Concretely, our algorithm is better in detecting each instance at the cost of a lower quality. This result is in accordance with our goal to build a discriminative semantic representation of the scene. We need to catch as much objects as possible with a lower priority on the pixelwise labelling. Despite good results some parts of labelled spherical images in INRIA dataset are particularly noisy. It is due to the stitching algorithm used to build each view from three images that change locally the light intensity (please consult the video attachment of the paper).

Results with enforced temporal consistency are presented in table II. It improves results of both global and average per

<sup>1</sup>The full resolution is 2048x665 but we use 1024x333 resolution for classification

<sup>2</sup><http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/>

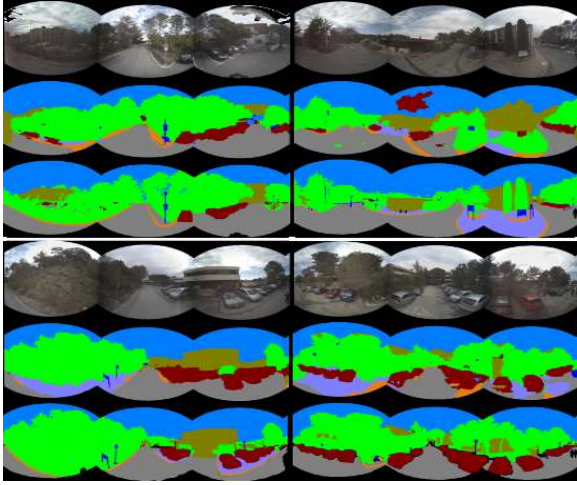


Fig. 2. Examples of frame by frame labelling results on INRIA database. Images are presented in the following order. Top: RGB image, Middle: Labelling results, Down: Ground Truth. Colors correspondence are: green: tree - blue: sky - gray: road - brown: building - light blue: signs - red: car - orange: ground signs - purple: sidewalk - black: others

class accuracy. However if the neighbourhood is too large labelling quality decreases. It comes from errors in depth estimation that project labels on wrong position. Attached video shows efficiency of temporal consistency to decrease over-illumination noise.

TABLE I  
RESULTS OF FRAME BY FRAME LABELLING

Method	Our algorithm		[21]	
	Global	Average	Global	Average
CamVid	79.2	<b>75.2</b>	<b>84.2</b>	59.5
INRIA	81.9	80.2	-	-

TABLE II  
COMPARISON OF LABELLING WITH TEMPORAL CONSISTENCY OVER DIFFERENT NEIGHBOURHOOD SIZES  $N_s$ .

Neighbourhood size	Global	Average
$N_s = 3$	82.1	81.0
$N_s = 7$	83.1	82.2
$N_s = 11$	81.3	80.4

#### IV. MAP CONTENT REQUEST

Localization is the task of retrieving the position of a map content query. It could be the position of the robot or any other content. Several methods like [23] propose a scene recognition scheme based on very low dimensional representation. Despite undeniable efficiency in scene recognition, those methods do not allow high-level content request and so are hardly extensible to tasks where human are "in the loop". At the opposite side, works like [24] propose a modelling scheme based on objects and places and use it to request high level content. These methods use the co-occurrence

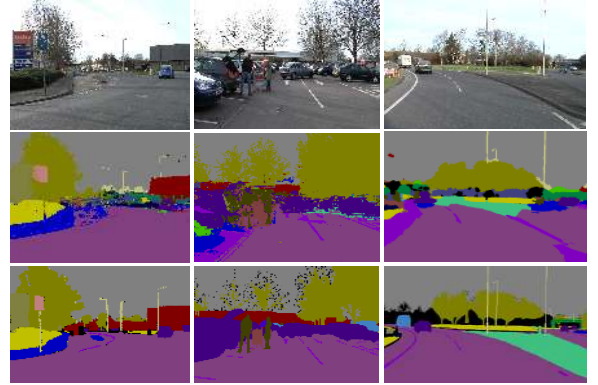


Fig. 3. Examples of frame by frame labelling results on CamVid database. Images are presented in the following order. Top: RGB image, Middle: Labelling results, Down: Ground Truth. From right to left: best to worst results.

of objects classes and places classes to predict place label or perform informed search. However, as said earlier, this strategy does not work outdoors because any object class can be present anywhere and the concept of "place" for open space is not straightforward. In this section we propose an algorithm that relies on MTS-map to efficiently realize localization of robot or any human-understandable concept like object or group of objects with given relations.

#### A. Semantic Localization

To achieve robust and efficient localization, our method relies on the proposed MTS-map structure. As explained in section III-A, local sub-maps are indexed in a tree structure encoding classes presence and occurrence. Each leaf is a set of sub-maps with similar semantic content. The first step consists in searching the tree for the leaf/leaves with corresponding content, for example, all leaves with two buildings. It allows to drastically reduce the number of sub-maps to compare with. Then semantic graphs  $G_o$  are compared to select the most probable local sub-map where for finding the needed information. Due to change in viewpoint that can possibly fuse several objects, comparing those graphs formulates as multivalent graph matching problem. This is NP hard problem but we can use structure of the graph to speed up the process. We use a variation of the interpretation tree algorithm [25] presented at Algorithm 1. Finally, when high precision relocation is needed, the visual odometry method presented in [1] is used on the RGBD layer to achieve centimetrical precision.

Our semantic graph representation presents several advantages over other ways to abstract an image: it relies on the entire image and not just on sparse local features that could be subject to noise, perceptual aliasing or occlusion. It intrinsically encodes image structure that contains an important part of the information. Graphical description allows to reconstruct approximate image structure while collection of low-level features do not. It is extremely lightweight: the size of the map with all full size images is 53Gbytes while semantic graphs representation needs only 18.5Mbytes, which

correspond to a compression ratio around 3000.

### B. Interpretation Tree Algorithm

Interpretation tree is an efficient algorithm that uses relationships between nodes of two graphs to speed up the matching process. It relies on two kinds of constraints to measure similarities called *unary constraints* and *binary constraints*. Let  $\mathcal{G}_1$  and  $\mathcal{G}_2$  be such two graphs. *Unary constraints* compare a node of  $\mathcal{G}_1$  to those of  $\mathcal{G}_2$ . If comparison succeed nodes are matched and a score is computed for the couple of nodes. Then the best pair of nodes is added to the list  $\mathcal{L}$  of matched nodes and *binary constraints* check if every two pairs of nodes in  $\mathcal{G}_1$  and  $\mathcal{G}_2$  have compatible relationships. We use the following constraints :

*Unary constraints*: they use three properties of nodes. Their label, the eccentricity and the orientation of the elliptical envelop that fits the corresponding area shape. If labels are different or the difference of shape properties is higher than a given threshold, comparison fails. Taking into account only labels, eccentricity and orientation allows to be robust to change in apparent size of semantic areas.

*Pairwise constraints*: they check relationships of two nodes. To do this they use weights  $w_i$  provided by the adjacency matrix of each semantic graph.

The interpretation tree returns the number of matched nodes. The highest score gives the most probable position.

---

**Algorithm 1** Details of our Interpretation Tree algorithm used to compare semantic graphs

---

```

INPUTS:  $\mathcal{G}_1, \mathcal{G}_2$ : graphs of the current view and a given view in the
database
OUTPUTS: Score of the matching (list of matched nodes)
for all Nodes  $n_i \in \mathcal{G}_1$  do
  for all Nodes  $n_j \in \mathcal{G}_2$  do
    if UnaryConstraint( $n_i, n_j$ ) then
      add ( $n_i, n_j$ ) to MatchNodesList
    end if
  end for
  if MatchedNodesList  $\geq 1$  then
    sort MatchNodesList
    for all ( $n_i, n_j$ ) in MatchedNodesList do
      add ( $n_i, n_j$ ) to InterpList
      if PairwiseConstraint(InterpList) == False then
        remove ( $n_i, n_j$ ) to InterpList
      end if
    end for
  end if
end for
end for

```

---

## V. LOCALIZATION FROM IMAGES RESULTS

In this section, we present our results to the problem of localizing an image in a database, which corresponds to the robot localization problem. We compare our algorithm performance with recent state-of-the-art Bag-of-Words techniques <sup>3</sup> presented in [14]. Their algorithm builds offline a tree structure that performs hierarchical clustering within the image descriptor space. Then similarity between current image and images in database is evaluated by counting

<sup>3</sup>We used the the implementation publicly available at: <http://webdiis.unizar.es/dorian/>

TABLE III  
RETRIEVAL TEST RESULTS: TIME EFFICIENCY FOR EACH ALGORITHM.

Dataset	Mean retrieval time
BoW K=10, L=5	22ms
BoW K=8, L=4	16ms
Interp	8.40ms
Interp+Index	0.12ms
Index	<b>54.20<math>\mu</math>s</b>

the number of common visual words. We have trained the vocabulary tree with two sets of branching factor and depth levels: K=10, L=5 producing 100000 visual words and K=8, L=4 producing 4096 visual words. The weighting strategy adopted between visual words is the *term frequency-inverse document frequency* tf-idf and the scoring type is L1-Norm (for details about parameters see [14]).

We evaluate several aspects of the algorithm. In subsection A we study performances for image retrieval in wide database. In subsection B we evaluate the robustness of our algorithm to wide changes in viewpoint. In subsection C we present some interesting results that cannot be attained with low-level feature-based algorithms. The dataset used for tests is the INRIA dataset presented in section III-C. CamVid is not used in this section because the dataset is too small with only 101 labelled images for sequence 06R0 and 124 for sequence 01TP.

### A. Image retrieval

*Experiments setup*: The experiment consists in retrieving successively all images in the database. We use three variations of our method: first the tree structure is used alone to search for images with same classes and same number of occurrence. It is denoted as "Index". Then the Interpretation Tree is used to discriminate between remaining results. It is denoted as "Interp+Index". Finally we use only the Interpretation Tree, denoted as "Interp". "BoW" denotes the Bag-of-words algorithm.

*Results*: Timings are presented in table III. All versions of our algorithm outperform BoW techniques in terms of time efficiency. This comes from the use of image structure that discriminate very fast between good and false candidates and the simple tests performed. Checking labels, shape properties and strength of the relation is very fast. The use of index alone is faster than all the other methods as it simply count the number of nodes of each class. However it does not encode image structure so it is subject to aliasing.

### B. Accommodation to view point change

*Experiments setup*: We run two experiments to evaluate robustness to changes in viewpoint. The first one consists in taking a subset of images from the original dataset to build a *reference database* and a different interleaved subset to build required database. We take 1 image out of 40 to build the database and 1 out of 40 shifted by 20 images to build the required set, denoted as *Distant images*. Then, we retrieve the positions of distant images in the reference database.



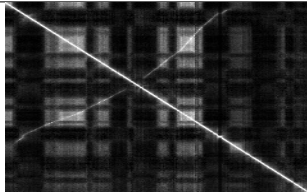
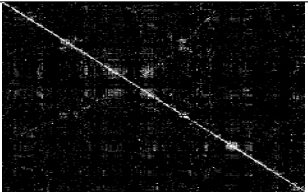
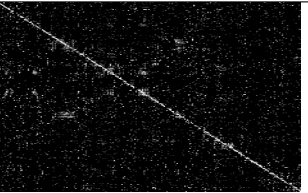
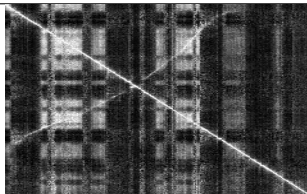

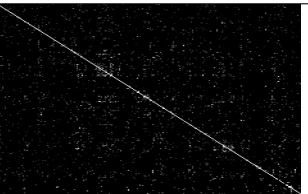
	BoW K=10, L=5	Interpretation Tree	Interp. Tree + restrictions
Distant images			
$D_p$	5.68	17.38	16.43
Monocular images			
$D_p$	2.99	13.37	22.36

Fig. 4. Distance matrix and discriminative power  $D_p$ . The brightest is the pixel the closest are the images.

In the second experiment we match monocular images with our spherical view database. Monocular images are taken from the same position as spherical views however focal and image size are different. For these experiments we compare BoW with Interpretation Tree alone. Due to large distances between images, Index is not useful since class occurrences changes significantly.

*Performance measurements:* We report two measurements to quantify the quality of matching algorithm. First we compute the distance matrix presented in figure 4. Rows correspond to database images and columns to images to retrieve. For each image to retrieve we compute distances with all images in the reference database. Ground Truth corresponds to diagonal line.

The second measurement is the discriminative power of the algorithm denoted as  $D_p$ . It is measured by first computing the mean distance between the image to retrieve and corresponding images in the dataset denoted as  $d_{true}$ . The number of corresponding images is arbitrarily set to the three closest to ground truth position. Then we compute the mean distance between image to retrieve and non corresponding images in the dataset denoted as  $d_{false}$ . We define  $D_p = d_{false}/d_{true}$ .  $d_{false}$  and  $d_{true}$  are normalized to  $[0,1]$ . The highest  $D_p$  is, the more discriminant is the algorithm.

1) *Results with large distances between images:* In this experiment our algorithm outperforms the BoW method. The discriminative power is more than three time higher with our algorithm. This comes from the fact that our algorithm use the image structure that allows to discriminate more efficiently between true and false matching. Additionally semantic areas are consistent along a wide range within images and are less sensitive to local similarities than feature-based methods.

2) *Results with monocular views:* Interpretation Tree outperforms BoW methods for monocular to spherical view matching with discriminative power more than 4 times higher. However on BoW distance matrix we can see a less pronounced cross-diagonal line corresponding to visited

TABLE IV  
TIME TO RETRIEVE HIGH LEVEL CONTENT

Request	Retrieval Time
Two buildings	56 $\mu$ s
No car	55 $\mu$ s
Car on road	0.95ms
Trees at the right of building	0.86ms

places of the robot but in the opposite direction. This structure is not detected by our algorithm. It comes from the fact that semantic graph encodes shapes of areas to match nodes. When only a part of the area is observed match is not possible. This makes our algorithm more sensitive to cases where the field of view is reduced. Moreover the number of nodes matched in graph is much lower than with spherical views (roughly 1/4). This results from the partial observation of some semantic areas that are not yet matchable because of changes in associated shape descriptors.

### C. SEMANTIC CONTENT REQUEST RESULTS

In this section we present results for tasks that need to use high-level concepts and are not possible otherwise. In the first part we study the time efficiency of localizing a given set of objects. In the second part we study the localization under constraints problem, that is without using some parts in the image.

a) *Request high level content:* The tree structure used to index sub-maps encodes classes presence and occurrence. Therefore it is easy to request from the map a list of spherical views where particular classes are observed. It is also possible to use more complex requests as for example the list of spherical views where two classes with particular relationships are observed. For this test we slightly change the previous graph structure to take into account relative positions instead of strength of connection previously expressed in term of number of pixels of different classes connected together. We do this to make it easier to specify relationships.

The possible values are: 1 = left, 2 = top left, 3 = top, 4 = top right, 5 = right, 6 = bottom right, 7 = bottom, 8 = bottom left. Table IV provides average time costs for some typical scene content requests. Notice that request times are extremely small due to the small size of graphs as we keep only those candidates where the request graph is fully matched.

*b) Relocation under constraint:* Navigation-oriented maps should provide an efficient way to deal with dynamic environments for lifelong mapping. Our map structure and the proposed relocation algorithm allow to take care of possible changes in images using semantics. For example cars are not reliable due to their dynamic nature. So we can ignore them by removing the corresponding nodes in graph.

Results of relocation without using car class are presented in figure 4, last column. Distance matrix is similar to the original one and only very slight changes occur. However for monocular and distant images discriminative power changes in opposite direction. We can explain this by the fact that in large graphs (panoramic images), each node is connected to a great number of others. Interpretation tree easily detects false node matching with binary constraints so that matched nodes are reliable. In small graphs only a few edges are allowed and false nodes matching are difficult to detect. Cars correspond most of the time to small areas in image so that the shape changes a little from one instance to another. False matching with cars is more probable than with larger semantic classes like tree or classes with more characteristic shapes. So removing cars in small graph can improve matching score while removing other classes does not. This behaviour is confirmed by tests with restrictions on other classes. Removing trees or buildings decreases matching score in all cases.

## VI. CONCLUSIONS

In this paper we have presented a new hybrid map representation well-suited for large scale outdoor environments. It combines in local sub-maps compact semantic and precise metric representations. We have also proposed an efficient query-based algorithm for coarse to fine relocation in our map. It outperforms a state-of-the-art feature-based relocation algorithm, it is able to request high level human-understandable content and easily adapts to outdoors scenes.

There are several ways to extend this work. First, the graph representation relies only on shape properties of semantic areas. It makes it difficult to match with monocular images or images taken from very different viewpoints. It could be interesting to use other properties to match nodes of the graph like color histograms in order to make matching process more robust to partial observations. Second, our algorithm entirely relies on semantics and improving labelling quality can improve relocation. Pixel-wise temporal consistency has been shown to improve labelling quality and higher level temporal consistency can be interesting to investigate. For example temporal stability of semantic areas can inform on the underlying labelling quality.

## REFERENCES

- [1] M. Meilland, A. I. Comport, and P. Rives, "A spherical robot-centered representation for urban navigation." in *IROS*. IEEE, 2010.
- [2] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Robotic exploration as graph construction," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 6, Dec 1991.
- [3] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *JOURNAL OF ROBOTICS AND AUTONOMOUS SYSTEMS*, vol. 8, 1991.
- [4] A. Chapoulie, P. Rives, and D. Filliat, "Appearance-based segmentation of indoors/outdoors sequences of spherical views," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2013*, Tokyo, Japon, 2013.
- [5] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA'12)*, Saint Paul, MN, USA, may 2012.
- [6] S. Vasudevan and R. Siegwart, "Bayesian space conceptualization and place classification for semantic maps in mobile robotics," *Robot. Auton. Syst.*, vol. 56, no. 6, jun 2008.
- [7] O. M. Mozos, P. Jensfelt, H. Zender, G.-J. M. Kruijff, and W. Burgard, "From labels to semantics: An integrated system for conceptual spatial representations of indoor environments for mobile robots," in *Proceedings of the IEEE ICRA Workshop: Semantic information in robotics*, 2007.
- [8] A. Ranganathan and F. Dellaert, "Semantic modeling of places using objects," in *rss*, atlanta, 2007.
- [9] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robot. Auton. Syst.*, vol. 56, no. 11, nov 2008.
- [10] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 0, 2013.
- [11] Y. Yagi, Y. Nishizawa, and M. Yachida, "Map-based navigation for a mobile robot with omnidirectional image sensor copis," *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 5, Oct 1995.
- [12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, 2004.
- [13] M. Cummins and P. Newman, "Highly scalable appearance-only SLAM - FAB-MAP 2.0," in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.
- [14] D. Galvez-Lopez and J. Tardos, "Bags of binary words for fast place recognition in image sequences," *Robotics, IEEE Transactions on*, vol. 28, Oct 2012.
- [15] M. Meilland, A. I. Comport, and P. Rives, "Dense visual mapping of large scale environments for real-time localisation," in *IEEE International Conference on Intelligent Robots and Systems*, sept. 2011.
- [16] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, 2006.
- [17] A. Bosch, A. Zisserman, and X. Muoz, "Image classification using random forests and ferns," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Oct 2007.
- [18] Y. Bengio, "1 learning deep architectures for ai."
- [19] J. Xiao and L. Quan, "Multiple view semantic segmentation for street view images," in *ICCV*. IEEE, 2009.
- [20] C. Wojek and B. Schiele, "A dynamic conditional random field model for joint labeling of object and scene classes," in *European Conference on Computer Vision (ECCV)*, October 2008.
- [21] O. Miksik, D. Munoz, J. A. D. Bagnell, and M. Hebert, "Efficient temporal consistency for streaming video scene analysis," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2013.
- [22] P. Krhenbhl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in Neural Information Processing Systems 24*, J. Shawe-taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011.
- [23] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vision*, vol. 42, no. 3, May 2001.
- [24] P. Viswanathan, D. Meger, T. Southey, J. Little, and A. Mackworth, "Automated spatial-semantic modeling with applications to place labeling and informed search," in *Computer and Robot Vision, 2009. CRV '09. Canadian Conference on*, May 2009.
- [25] W. E. L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*. Cambridge, MA, USA: MIT Press, 1990.