Visual Place Categorization in Maps

Ananth Ranganathan
Honda Research Institute USA, Inc
aranganathan@honda-ri.com

Jongwoo Lim
Honda Research Institute USA, Inc
jongwoo.lim@gmail.com

***Abstract*—Categorizing areas such as rooms and corridors using a discrete set of labels has been of long-standing interest to the robotics community. A map with labels such as kitchen, lab, copy room etc provides a basic amount of semantic information that can enable a robot to perform a number of tasks specified in human-centric terms rather than just map coordinates. In this work, we propose a method to label areas in a pre-built map using information from camera images. In contrast to most existing approaches, our method labels the area that is viewed in the camera image rather than just the current robot location. Place labels are generated from the image input using the PLISS system [14]. The label information on the viewed areas is integrated in a Conditional Random Field (CRF) that also considers higher level semantics such as adjacency and place boundaries. We demonstrate our technique on maps built using from laser and visual SLAM systems. We obtain the correct place categorization of a very high percentage of the map areas even when the place categorization system is trained using images only from the internet.**

I. Introduction

Robot map building has achieved a significant level of sophistication in recent years. SLAM algorithms can now build online maps of very large spaces in both natural and man-made environments [7][2]. However, metric maps by themselves are not conducive to human robot interaction since all location related commands have to be given in map coordinates. Hence the push towards semantic mapping wherein intuitive semantics are associated with locations and spaces in maps.

In this paper, we focus on categorizing areas in maps with one of a given set of labels. For example, by labeling an area of the map with the label "Kitchen", we enable interactions with the robot of the form "Bring me X from the kitchen" or "Take me to the kitchen". We believe that the categorization of places in this manner is also integral to other aspects of semantic mapping since place labels also inform us regarding the kinds of objects that could possibly exist in those places.

We perform place categorization using monocular images. At each step we label the area in the metric map of the environment corresponding to that viewed in the image. Our method works with both laser and vision generated maps, as we demonstrate in experiments. An illustration of the working of the algorithm using a laser-based map is shown in Figure 1.

A major difference of our approach to existing map labeling methods is that our method labels the viewed area rather than the location of the robot. This is important since a robot in a corridor, for example, could be looking into a meeting room through a glass window, as illustrated in Figure 2. Further,

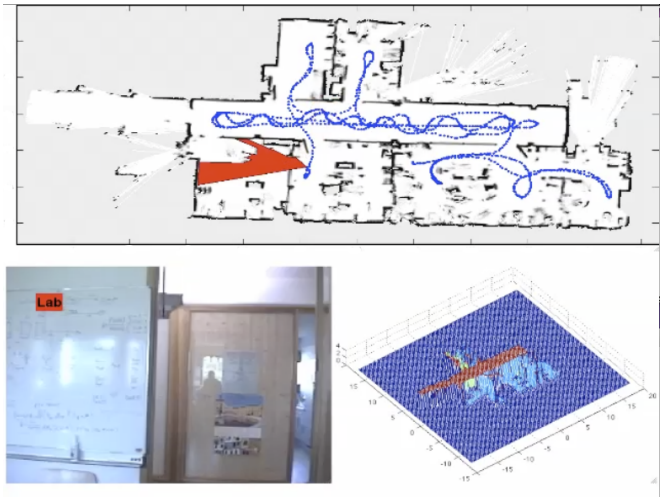

Figure 1: Place labeling in a laser-based map: The top portion shows the current area viewed by the robot (in red) and the robot trajectory from start is shown as a dotted line. The image from the robot camera used for place categorization is shown at bottom left, along with the maximum aposteriori place label from the PLISS algorithm. The label probabilities are accumulated in a grid map of the environment, and the most likely labeled map at the current step is shown at bottom right. Light blue corresponds to the place category "Lab", red to "Corridor", and green to "Copy room". Dark blue corresponds to unknown/unseen areas. A video of the map labeling process can be viewed at www.ananth.in/laser-map-output.mov and is also available as a video attachment to this paper.

our method is applicable to maps generated using both laser and visual SLAM techniques, as we demonstrate. This is in contrast to the few existing systems that use the viewed area, albeit indirectly, for place categorization [4][13], and which are exclusively based on laser-based SLAM maps. We believe that place categorization is inherently a visual task and laser range scan data, while being excellently suited for recognizing specific places, is much less useful for place category recognition. Thus, our use of visual place categorization with both laser-based and vision-based maps provides a significant contribution to the state of the art.

We use the PLISS system [14] for sequential place categorization of the images. PLISS has been shown to generalize well and takes video rather than individual images as input, thus providing a significant level of temporal consistency to

Figure 2: An instance of the robot looking into a meeting room which it does not physically enter. Classifying the robot's location as a meeting room based on this image would be incorrect in this case. Hence, we classify the area viewed by the robot rather than the robot's location itself.

the labeling. PLISS also provides probabilistic label output, including additional 'unknown' or 'transition' labels when a place label is not identifiable by the system. We demonstrate the map labeling on vision-based metric maps generated using the system of [10], which provides a feature-based 3D point-cloud map of the environment. The working on laser-based maps is demonstrated using a publicly available dataset.

Simply labeling the area viewed in each image frame consecutively does not suffice to produce a well labeled map due to errors in labeling and also localization error within the map. Thus, we accumulate the label probabilities for each area in a spatial grid representation, where each grid cell maintains a probability distribution over labels for the corresponding region in the metric map. We further construct a Conditional Random Field (CRF) [8] over the accumulated probabilities, which incorporates continuity constraints between neighboring cells except when a boundary is present. The CRF acts as a smoothing mechanism over the grid cell labels, yielding a more accurate map in which labels adhere to physical constraints.

In the following exposition, we first discuss existing approaches to place labeling in maps followed by an overview of the place categorization method we use called PLISS. Subsequently, we explain the core technique of this paper relating to labeling in maps, both vision-based and laser-based, and also our use of CRFs to enforce spatial consistency in labeling. Finally, we present experiments to justify our claims and conclude with a discussion on some limitations and unaddressed issues in our method.

II. Related Work

We start by providing an overview of the existing work on place labeling in maps. Mozos et al. [12] present a method for labeling regions in a laser-based map and subsequently converting this into a topological mapping map with labeled nodes. Map labeling is performed by simulating laser scans from each grid cell of the map, which are classified using a pre-trained classifier. Hence, in contrast to our method, this technique is only applicable to laser maps and also labels only the robot location and not the viewed area. In the case of laser range scans, we define the "viewed area" as the area of the scan. After each grid cell of the map has been labeled, an associative Markov network is used to provide spatially consistent labeling.

Voronoi Random Fields (VRF) [4] also create a labeled topological map as a final output, with the labeled metric map as the intermediate result. Instead of labeling each grid cell of the map as above, labels are generated based on the Voronoi graph of the map. Nodes on the Voronoi graph are labeled using a method similar to above [12] and these nodes are placed in a Conditional Random Field (CRF), inference on which yields a labeled map. As before, this method is specific to laser-based maps.

A method that is close to our own is [3], which is based on visual place categorization and uses Conditional Random Fields. However, the categorization is performed for the location of the robot rather than for the viewed area. Hence, only the robot trajectory is classified. Places that the robot sees but does not enter cannot be classified using this method, which is in contrast to our approach here. An explicit constraint involving a door detector ensures that the labels between doors remain the same. However, this heuristic may not hold, especially in wide open environments like offices, and can also create large cliques in the CRF, making the inference using graph cuts extremely slow.

Place categorization itself has also received a lot of attention in the literature and we provide only some recent results here. The VPC system [21] uses monocular images to learn classifiers for the various place labels. Similar classifier based systems based on global image statistics are used for image categorization in the computer vision literature [9][20]. The system by Pronobis et al [13] uses input from different sensing modalities and merges these cues to obtain the final place label. The PLISS system [14], which we use, differs from previous systems in using a Bayesian framework that inherently enforces temporal consistency and provides a probabilistic output labeling.

Recently, many systems have been proposed that detect objects in the environment and use these as semantic cues to categorize the place. Rottman et al. proposed a method in this vein that combines laser range features and visual features. They use a fast cascade object detector [18] to detect objects efficiently, and these detections are the visual features used in the system. Classifications using the laser and visual features are filtered using a Hidden Markov Model (HMM) to improve accuracy. More ambitious systems such as [5][19][17][6] use object statistics such as object counts, their co-occurrence, and their location to determine the place label. However, given that object detection is still an unsolved problem for general objects, these methods may not be robust. Further, the presence of commonplace objects such as monitors, lamps, and chairs may confuse these methods since they occur in many different place categories.

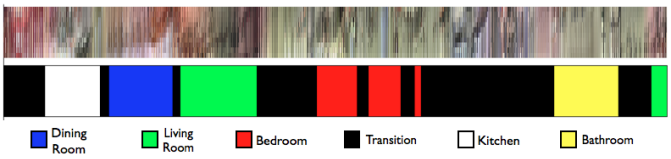

Figure 3: An image sequence (top) with images stacked up in the form of slivers. The maximum aposteriori labeled output from PLISS at the given timestep is shown below along with a legend for interpreting the colors as place labels. PLISS segments the image sequence through change-point detection. This is performed in an online manner so that the segmentation does not need to be performed starting from scratch at each step.

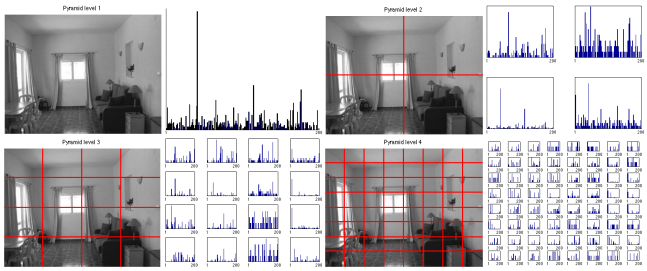

Figure 4: The Spatial Pyramid histogram: Histograms of clustered SIFT features, computed on image regions at different spatial resolutions, are concatenated to yield the representation of the image. The image regions are obtained by dividing it into successively finer grids.

III. Place Categorization Using PLISS

The first step to place labeling in a map is to generate the place labels themselves from sensor input. We use camera images as input for the place categorization task, and use the PLISS system proposed by us in previous work [14]. We now present a brief overview of PLISS.

PLISS, which stands for Place Labeling through Image Sequence Segmentation, works with video or image streams, thus intrinsically considering the strong temporal component of the place categorization problem in robotics. This strong temporal aspect arises because the robot cannot instantaneously jump from one place type to another so that the place label remains the same for large periods of time and only changes occasionally. PLISS uses online Bayesian change-point detection to segment the image streams into portions corresponding to different place categories, as shown in Figure 3. Change-point detection is the problem of detecting relatively abrupt changes to the parameters of a statistical model. We model the images in a manner such that the locations of these abrupt changes within the image stream correspond to the place boundaries where a place is entered or exited. Once the changepoints have been detected, each segment of the image stream is probabilistically classified. Thus, PLISS decomposes the place labeling problem into two subproblems - computing the changepoints and computing the place labels given the changepoints.

PLISS uses spatial pyramid histograms [9] of densely computed SIFT features, an example of which is shown in Figure 4. Each input image is converted into a multi-resolution histogram of dense SIFT features, which is modeled using a Multivariate Polya model [11]. Change-point detection is performed by monitoring the parameter values of this model as it is updated in an online fashion using histograms from the input images. Pre-trained place models are used to classify the segments of the image stream detected through changepoint detection.

PLISS is an online algorithm and operates in real-time despite being completely probabilistic and not making any hard decisions at any step. All computations are performed in a Bayesian manner to ensure that no irrevocable decisions are made, either on the changepoints or on the place labels, at any time step.

After each time step, the maximum aposteriori (MAP) changepoint distribution is used to compute the discrete distribution on place labels for the current input image. This is the place label distribution we use for labeling in the map of the environment. Further details on PLISS can be found in [14].

IV. Place Categorization in Maps

Given the sequential image label from PLISS, our task is now to label the region in the map corresponding to the area seen in the image. We approach this task by maintaining a grid map of the environment for spatially accumulating the place label probabilities. Each grid cell contains a discrete distribution over place labels. By accumulating the output probabilities from PLISS in such a grid map, we can overcome labeling errors since it is unlikely that a place will be wrongly categorized after the robot has viewed it multiple times.

Probability accumulation over the grid map is performed as follows -

1) Compute the area of the map that is visible in the current image (the viewed area)
2) Identify the grid cells corresponding to the viewed area
3) Add the output label distribution from PLISS to the distributions in each of the above grid cells

While step 3 is self-evident, we now describe steps 1 and 2.

We approximate the viewed area in the image by a polygonal shape, which is intersected with the grid map to determine the cells to be updated. The details of determining this polygonal shape for vision-based maps are described in Section IV-A and for laser maps in Section IV-B respectively.

The grid cells corresponding to the viewed area (step 2) are obtained by computing the intersection of the viewed area polygon with the grid map. In essence, we need to identify the grid cells lying inside the viewed area polygon. This is done using a point-in-polygon algorithm. In particular, we use the crossing number algorithm [16], a well-known method in computer graphics.

We apply the point in polygon test to all the grid vertices inside the bounding box of the viewed area polygon, as

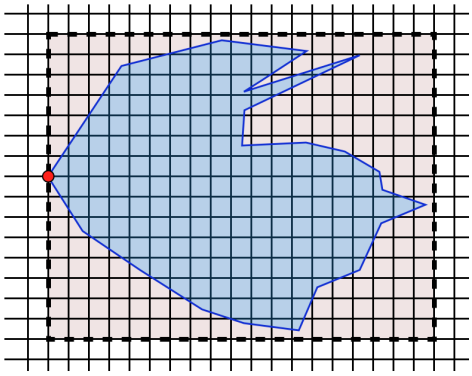

Figure 5: The area viewed by the robot (shown in red) at any given instant is represented using a polygon, called the viewed area polygon (shaded blue), which is intersected with the place label grid map to determine the grid cells to be updated using the place label probability obtained at that time instant. The place label probability is added to the grid cells inside the viewed area polygon while the grid cells on the polygon boundary are updated in proportion to the area of the cell inside the polygon. The dashed line represents the bounding box of the polygon. Only the grid vertices inside the bounding box are tested to determine if they are within the polygon.

illustrated in Figure 5. Grid cells with all four vertices inside the polygon or outside the polygon are handled in a straightforward manner. For grid cells that lie partly in the polygon, we determine the proportion of the grid cell lying inside the polygon with a detailed intersection test between the grid cell boundaries and the polygon edge or edges intersecting the grid cell. The label probability distribution is multiplied by this factor before adding to the grid cell value.

Since a large number of point-in-polygon tests are performed, we improve the efficiency of this test by first sorting the grid cell vertices in the bounding box by their y-coordinate, and subsequently using a binary search to find the first and last vertices in the sorted list that have a chance of intersecting any of the edges of the polygon[1]. Thus, only a small portion of the grid points are tested for intersection with each polygon edge.

We now describe the computation of the viewed polygon (step 1 above) at each timestep for both the laser-based and vision-based maps.

A. Labeling laser maps

Determining the viewed area in the image is relatively easy in the case of a laser map. We simply simulate the laser scan readings through ray tracing in the laser map. However, ray tracing is only performed for the laser scan readings corresponding to the viewing angle of the camera. The area of this limited laser scan corresponds to the viewed area polygon.

[1]This optimization is adapted from the fast point-in-polygon test at http://www.mathworks.com/matlabcentral/fileexchange/10391-fast-points-in-polygon-test

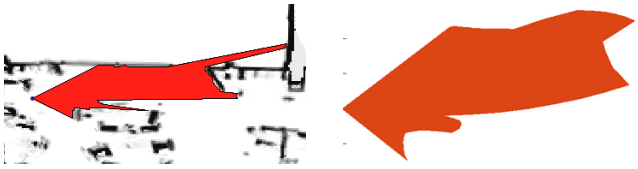

Figure 6: The area of the laser scan frequently contains narrow "peninisulas" that extend into other places (left). These can be removed by passing the scan through a smoothing filter to obtain a scan like the one on the right. Note that some of the obstacle boundaries have been lost due to smoothing. However, this is inconsequential for place labeling.

This is illustrated in Figure 1 where the viewed area polygon is shown in red.

However, as shown in Figure 6, the laser scan may have narrow "peninsulas" that extend into other places. Labeling these areas with the current place label would be erroneous. Hence, we first apply a smoothing filter to the laser scan. The width of the smoothing filter is taken to be 5 degrees, corresponding to five laser readings on either side of the center. We use a weighting vector that progressively down-weights the readings from the center of the filtering window. This operation removes the narrow peninsulas in the laser scans and results in a more semantically meaningful viewed area polygon. Note that using a smoothing filter that has a very large width is also not advisable as this causes the scan to not adhere to the obstacle boundaries so that the final labeled map will not have distinct boundaries between places.

B. Labeling vision-based maps

Determining the viewed area in a map obtained using visual SLAM (VSLAM) is harder since most existing VSLAM methods use a sparse map representation so that obstacle boundaries are not clearly visible as in a laser map. We use the system of [10] to obtain the VSLAM map, which consists of 3D locations of distinct features observed by the robot during it's map-building run. The robot can efficiently localize in the map by detecting features in the camera image and matching these against the map. The VSLAM system uses modified Harris corners limited to detection on edges [22] for computing the map representation and for localization.

Feature visibility does not provide an adequate cue for ground area visibility since features on the ceiling may be visible from a large distance away. However, this area cannot be labeled in the map since it is clearly beyond the visible ground area. At the same time, the area right in front of the robot cannot be labeled in the map if the nearest visible feature is some distance away. In this case, it is possible that the robot is looking into a room of a different place category than the location it is currently at.

We provide a few heuristics to address the above difficulties and obtain the viewed area polygon. In the case of a visual SLAM map, we use trapezoidal shapes, corresponding to the projection on the ground of the viewing frustum of the camera, to represent the viewed area polygon. The sides of the

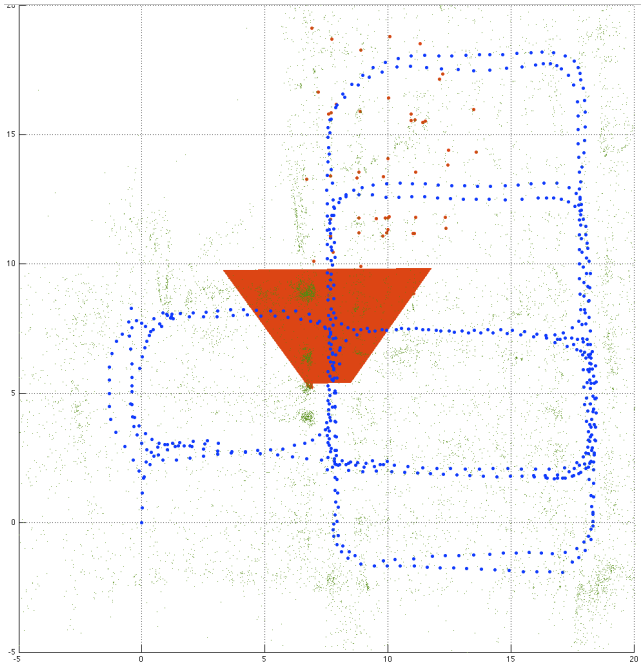

Figure 7: The viewed area polygon is a trapezium in the case of a VSLAM map. The dimensions of the trapezium (shown dark red) are determined using the features visible to the robot (red dots) from it's current location. The visible features beyond the trapezoid have been ignored since they are either beyond the distance threshold or above the height threshold. Other features in the map are shown in green. The robot trajectory is the blue dotted curve.

trapezium are physically constrained to be within the viewing angle of the camera. However, we constrain them even further to pass through the left-most and right-most features visible from the current robot location. The base of the trapezium is taken to be the line passing through the nearest feature and perpendicular to the robot's orientation. Similarly the farther side of the trapezium (the edge parallel to the base) is taken to be that passing through the farthest feature visible. If the farthest feature is beyond a certain threshold distance, this threshold is used to define the viewing area trapezium. Further, for all the above calculations, only those features below a certain threshold height are used so that the viewed area on the ground can be computed more effectively. These heuristics for computing the viewed area polygon (or trapezium) are illustrated in Figure 7.

Using a VSLAM map provides a few advantages over a laser map, which offset the disadvantage of not being able to obtain the viewed area polygon easily. First, a laser scanner is essentially a 2D sensor that registers all obstacles at it's mount height even though most of these are irrelevant for the purpose of visual place categorization. A camera, on the other hand, offers a 3D view of the environment. Second, no synchronization between sensors is required when using a VSLAM map since the mapping and place categorization modalities are the same.

V. Obtaining the Place-labelled Map using CRFs

Simply accumulating the place label probabilities using the viewed area polygon results in a noisy map that does not enforce spatial constraints on the labels. In the current setup, for instance, two grid cells next to each other having different place categories are not penalized, even though this is an unlikely case that can only occur at place boundaries. Further, no use has yet been made in our algorithm of any prior knowledge that we may have about the place labels. For instance, we might have observed from various building layouts that a kitchen is rarely next to a bathroom. Hence, the juxtaposition of a kitchen grid cell and a bathroom grid cell should be penalized in the map.

We can include our prior knowledge as well as get rid of the noise in the map through use of a Conditional Random Field (CRF). The CRF encodes a probability distribution over map labelings L given the input image stream I, a map M , and a robot trajectory T. We combine these into the set denoted by $Z = \{I, M, T\}$. The CRF probability is then given by

$$P(L|Z) \propto \prod_{s} \phi(x_s) \prod_{s,t} \phi(x_s, x_t) \quad (1)$$

where s is the set of nodes in the grid, and $\{s, t\}$ denotes the set of edges. The functions $\phi(x_s)$ and $\phi(x_s, x_t)$ denote the label likelihood and edge likelihood respectively. Note that these need not be probability distributions. We use a four-neighbor grid in the CRF to determine the neighboring nodes.

Computing the distribution over labelings from (1) is intractable for all but trivial grid sizes. Approximations such as loopy belief propagation can provide the marginal label distribution over each grid cell but are expensive to compute. Since we only require the maximum aposteriori (MAP) labeling, we use the well-known GraphCut algorithm [1].

GraphCut minimizes the energy function corresponding to the distribution in (1), which is given by

$$E(L) = \sum_{s} D_s(x_s) + \sum_{s,t} V_{s,t}(x_s, x_t) \quad (2)$$

where D_s and $V_{s,t}$ are the energy functions corresponding to $\phi(x_s)$ and $\phi(x_s, x_t)$ respectively, and are also known as the data term and the smoothness term.

The data and smoothness terms determine the output of the CRF and we now describe these. The data term encodes the cost of a grid cell taking on a specific place label. Hence, we use the negative logarithm of the output probability from PLISS directly as the data term in the CRF. If the probability of a specific label is low for the grid cell, this places a correspondingly high cost on giving that label to the grid cell in the CRF. The smoothness term, as the name suggests, is the cost given to assigning pairs of place labels to neighboring grid cells. We assign a higher cost to neighboring grid cells having different place labels compared to the cost of them having the same place label.

GraphCut minimizes the energy function (2) through the α-expansion algorithm [1]. α-expansion refers to the step wherein, in addition to any cells already labeled as α, any

Figure 8: Examples of training images obtained from Google image search used for training PLISS (a) Kitchen and (b) Office. These were obtained by giving the search terms "Kitchen" and "Office" respectively.

set of grid cells can take the label α, thereby expanding the number of cells with this label. The optimal α-labeling is computed by finding the min-cut on a specialized graph. The α-expansion algorithm iterates through the label set and attempts to expand each label in turn. The algorithm terminates when no lesser energy state can be found after cycling through all the labels. More details of the GraphCut algorithm can be found in [1].

We also assign an label adjacency cost matrix that makes the juxtaposition of certain labels more or less likely compared to others. For example, the cost of neighboring grid cells having the labels "Kitchen" and "Bathroom" may be relatively high as compared to "Lab" and "Corridor". The label adjacency cost matrix allows domain specific knowledge to be encoded in the CRF. However, note that in our formulation the cost of neighboring cells having two different labels is never less than that of them having the same label.

VI. Experiments

We tested our system on two datasets - one containing laser map data and the second containing a VSLAM-based map. In both the experiments, the PLISS system was trained using selected images obtained from a Google image search. No images from the test environments were used to learn the place categories. We used 250 images per place category for training the system, some examples of which are given in Figure 8.

The first experiment was conducted using the publicly available Albert-laser-vision dataset [15] from the Radish repository. The dataset consists of laser scans, ground truth robot poses, and camera images from a lab environment containing various labs, a corridor, and a printer room. The camera viewing angle is 65 degrees. We reconstructed the laser grid map from the dataset and used this grid map in our map labeling system. The dimensions of the environment considered in the dataset are approximately 25x15 meters. We use a grid of side 20cm for computing the labeled map. A snapshot of our system during the map labeling process is shown in Figure 1.

The place categorization result using just PLISS is shown superimposed on the robot trajectory in Figure 9. Existing place categorization approaches provide results similar to this,

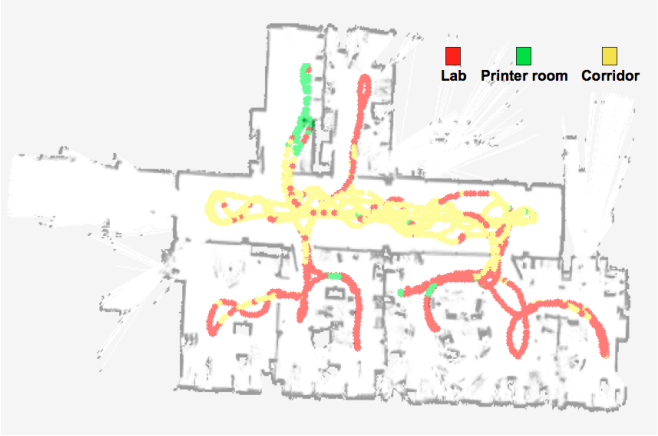

Figure 9: Robot trajectory for the AlbertB dataset with the location of each image colored according to the place recognized at that location. The misclassifications by PLISS can be removed by smoothing in a CRF framework similar to [3].

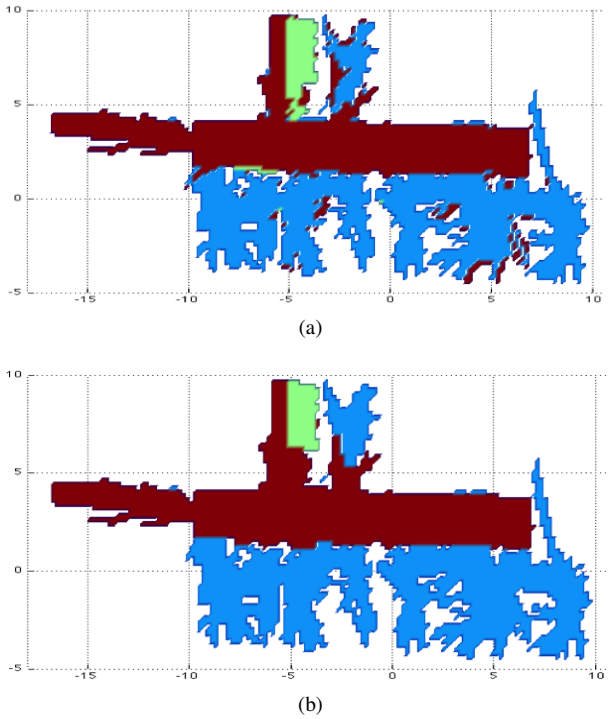

Figure 10: (a) Labeled map for the AlbertB dataset with stepwise accumulation of place label probability (b) The map with additional inference using CRF to enforce spatial and prior constraints. The error rates in both cases are 92.8% and 95.5% repectively. Blue, red, and green represent lab area, corridor, and printer room respectively.

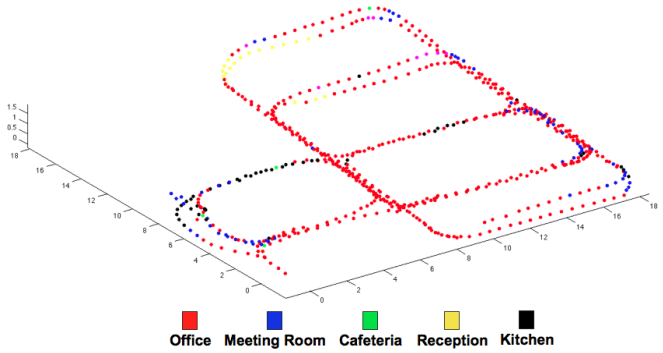

Figure 11: Robot trajectory with the location of each image colored according to the place recognized at that location. Note that the meeting room areas are marked in the wrong place since the robot never entered them but only looked into them. Compare to 13(b) for the correctly labeled meeting room areas.

in that they only categorize the robot trajectory and not the environment map [3]. In this case, PLISS alone provides a 91.3% labeling accuracy in categorizing the images. The groundtruth labels for computing accuracy were obtained through manual labeling.

Our method uses the online labeling from PLISS to obtain a labeled map of the environment. The labeled map from our method is shown in Figure 10(a). Note the noise due to incorrect categorizations from PLISS, which are also accumulated in the map using the method described in Section IV-A. The result of using the CRF to enforce label constraints and smoothness is given in Figure 10(b). In this case, we did not use a label adjacency cost since there are only three labels and any of them can occur next to each other. The accuracy of the map, as a percentage of grid cells labeled accurately in the map, is 92.8% without the use of the CRF and 95.5% upon using it. While this may seem a small difference, the CRF map is much more spatially consistent, and hence much more usable by a robot. This demonstrates clearly the advantage of using the CRF.

On a practical note, use of a laser map requires knowledge of the correspondence between the robot pose, the laser scan, and the camera image, i.e. these three information streams have to be synchronized to a tolerable accuracy. This is unavoidable since the mapping and the place labeling are being performed using different sensing modalities. The use of vision-based maps avoids this constraint.

The second dataset we use consists of images collected by us using a stereo rig in an office setting. The VSLAM system of [10] is used to construct the sparse feature map. The distance threshold for calculating the viewed area was set at 6 meters while the height threshold was 2.5 meters. The environment dimensions in this case are approximately 30x30 meters, and as before, a grid of 20cm side was used for computing the labeled map. Only the images from the left camera, which had a viewing angle of 100 degrees, were used for place categorization. In this case, PLISS alone gave a categorization accuracy of 79%. This is considerably lower

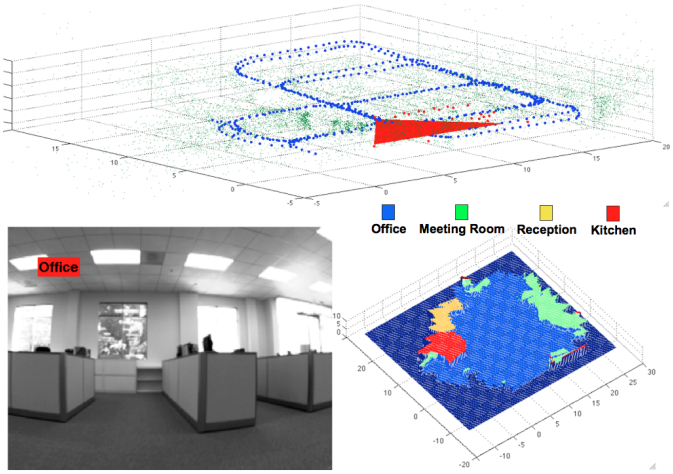

Figure 12: An snapshot from the second experiment which uses a visual SLAM map. The map with the robot trajectory is shown at the top with the area viewed by the robot in the current step (shaped as a trapezium) shown in red. This trapezium is obtained using the features in the map seen by the robot in the current step. These are shown as red dots. The input image with the most likely place label inferred by PLISS is shown at the bottom left. On the bottom right is the labeled map obtained by accumulating label probabilities.

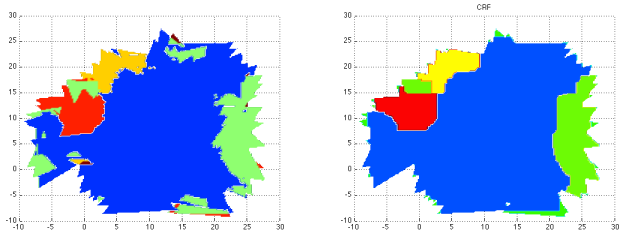

Figure 13: (a) Labeled map for the VSLAM map dataset with step-wise accumulation of place label probability (b) The map with additional inference using CRF to enforce spatial and prior constraints. See Figure 12 for the meanings of the colors used.

than in the previous experiment, the reason being that the place categories in this experiment are more similar to each other, for instance “office” and “meeting room” can be easily confused. The robot trajectory with the labels overlaid is shown in Figure 11. An illustration of the working of our system on this dataset is shown in Figure 12.

The output from our map labeling system is shown in Figure 13(a). Note that the robot only skirted the lobby area and did not traverse the meeting rooms at all. However, the robot could look into the meeting room from the office area through the glass wall (similar to Figure 2). Using existing methods, it would have been impossible to detect the meeting rooms since these methods only label the robot location, which is clearly not a meeting room. By labeling the area viewed by the robot rather than the robot’s location itself, we provide an intuitive solution to this problem.

The output from the CRF for this experiment is given in Figure 13(b). In this case, we used a label adjacency cost matrix that penalizes the "Kitchen" and "Meeting room" labels being next to each other, as well as the "Lobby" and "Kitchen" labels being adjacent. As before, the map given by the GraphCut algorithm is much cleaner, and yields a grid-wise labeling accuracy of 90.3% compared to 86.1% without it's use. To compute the labeling accuracy on the grid, we used only those grid cells that were viewed by the robot during its run. Note that the boundaries between place types are not as distinct as in the laser dataset because the VSLAM map does not indicate obstacles as does the laser map. Further, since the extent of the meeting room areas is not fully known as the robot did not venture into them, their boundaries are also uneven. However, in spite of the much lower PLISS accuracy, the final CRF map accuracy is similar to the first experiment, demonstrating the robustness of our method to errors in the place categorization itself.

VII. Conclusion

We have described a novel method for labeling places by category given a map of the environment. The place categorization itself is performed using the PLISS algorithm, which probabilistically categorizes camera images from the environment. Our map labeling algorithm consists of two main parts - first, determining the area in the map viewed by the robot using it's sensors, and second, computing the final labeled map from the labels given to these viewed areas. The first part is addressed separately for laser scanners and cameras, since the area viewed by the robot is dependent on the type of sensor used to build the map. For a laser sensor, this computation is relatively straight-forward. However, when using a visual SLAM generated map, computing the viewed area requires a set of heuristics that we provide in this work. The second part of the problem is addressed by accumulating the place category probabilities from each viewed area in a grid map and subsequently, performing inference on a CRF which places spatial constraints on the place categories.

We demonstrated our method on a publicly available dataset that contains a laser generated map, and on another dataset collected by us that contains a map generated by visual slam. The final labeled map has an accuracy of more than 90% in both cases, despite the place categorization becoming considerably more error-prone in the case of vision-based dataset. This provides validation for the robustness achieved through probability accumulation using a grid map.

Our method as presented here requires a metric map of the environment and also requires that the robot is localized in the map while the map is being labeled. These constraints can be relaxed somewhat by combining the mapping and labeling to have an online system that generates labeled metric maps directly. Such an online system can be implemented in a straight-forward manner.

In the future, we plan to extend our method to use the full geometry of the environment in the visual slam case, based on the 3D point cloud of point features. The use of semantic features, such as doors, in the labeling procedure is also to be addressed. Improvement of the PLISS algorithm itself will also help improve the map labeling accuracy.

In this work, we assume the existence of a pre-built metric map, although our method can easily be integrated into an online mapping and labeling system.

References

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(11):1222–1239, 2001.

[2] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[3] E. Fazl-Ersi and J.K. Tsotsos. Energy minimization via graph cuts for semantic place labeling. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[4] S. Friedman, H. Pasula, and D. Fox. Voronoi random fields: extracting the topological structure of indoor environments via place labeling. In *Intl. Joint Conf. on AI (IJCAI)*, 2007.

[5] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. FernÃ¡ndez-Madrigal, and J. GonzÃ¡l ez. Multi-hierarchical semantic maps for mobile robotics. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3492–3497, 2005.

[6] H. Zender H, O. Martínez Mozos, P. Jensfeltc, G. Kruijffa, and W. Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008.

[7] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. on Robotics*, 24(6):1365–1378, 2008.

[8] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Intl. Conf. on Machine Learning (ICML)*, 2001.

[9] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[10] J. Lim, J.-M. Frahm, and M. Pollefeys. Online environment mapping. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[11] T.P. Minka. Estimating a dirichlet distribution. 2003.

[12] O. Martínez Mozos and W. Burgard. Supervised learning of topological maps using semantic information extracted from range data. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2772–2777, 2006.

[13] A. Pronobis, O. M. Mozos, B. Caputo, and P. Jensfelt. Multi-modal semantic place classification. *Intl. J. of Robotics Research*, 2010. accepted.

[14] A. Ranganathan. Pliss: Detecting and labeling places using online change-point detection. In *Proceedings of Robotics: Science and Systems*, 2010.

[15] Cyrill Stachniss. The robotics data set repository (radish), 2006.

[16] I. Sutherland, R. Sproull, and R. Schumacker. A characterization of ten hidden-surface algorithms. *ACM Computing Surveys*, 6(1), 1974.

[17] S. Vasudevan, S. Gachter, M. Berger, and R. Siegwart. Cognitive maps for mobile robots — an object based approach. *Journal of Robotics and Autonomous Systems*, 55(5):359–371, May 2007.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[19] P. Viswanathan, D. Meger, T. Southey, J. J. Little, and A. K. Mackworth. Automated spatial-semantic modeling with applications to place labeling and informed search. In *Proceedings of Canadian Robot Vision*, 2009.

[20] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Learning locality-constrained linear coding for image classification. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[21] J. Wu, H. Christensen, and J. M. Rehg. Visual place categorization: Problem, dataset, and algorithm. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[22] J. Xiao and M. Shah. Two-frame wide baseline matching. In *Intl. Conf. on Computer Vision (ICCV)*, pages 603–609, 2003.