2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Visual Robot Localization Using Compact Binary Landmarks

Kouichirou Ikeda and Kanji Tanaka

Abstract— **This paper is concerned with the problem of mobile robot localization using a novel compact representation of visual landmarks. With recent progress in lifelong map-learning as well as in information sharing networks, compact representation of a large-size landmark database has become crucial. In this paper, we propose a compact binary code (e.g. 32bit code) landmark representation by employing the semantic hashing technique from web-scale image retrieval. We show how well such a binary representation achieves compactness of a landmark database while maintaining efficiency of the localization system. In our contribution, we investigate the cost-performance, the semantic gap, the saliency evaluation using the presented techniques as well as challenge to further reduce the resources (#bits) per landmark. Experiments using a high-speed car-like robot show promising results.**

I. Introduction

This paper is concerned with the problem of mobile robot localization using a novel compact representation of visual landmarks. With recent progress in lifelong map learning [1] as well as in information sharing networks [2], it has become crucial for mobile robots to obtain and use a large-size database of visual landmarks [3]. Accordingly, compact representation of landmark database attracts much interest. The motivation of this study is to enhance compactness of a landmark database while maintaining efficiency of the robot localization system.

We follow the visual retrieval approach [3] [5]- [10] to search a large-size landmark database. More formally, in our system, a visual vocabulary module maps high dimensional visual landmarks to fewer dimensional visual words, and then an information retrieval module indexes and searches relevant visual words in the landmark database. In general, space cost of such a landmark database mainly consists of

1) cost for visual vocabulary, and
2) cost for individual visual words.

Reducing the both kinds of costs is a basic requirement for a compact landmark database.

Our landmark representation is inspired by a (e.g. 32bit) compact binary code representation from web-scale image retrieval community. In CBIR (content-based image retrieval), binary codes are used for searching similar images in a database. In their approaches, a short sequence of binary codes is assigned to each image and memorized in a database. With such a binary code representation, similarity search using hash tables as well as bit count operation can be quite fast. In this line of research, a state-of-the-art is the semantic hashing technique developed by Salakhutdinov and Hinton [11]. The semantic hashing learns a deep graphical model of K-bit code vector from a huge number (e.g. 1×10^5) of training images. The binary code directly points the memory address of relevant features. Near neighbors in terms of Hamming distance are found fast by accessing only a few addresses in a Hamming ball. In [12], the semantic hashing technique achieves successful information retrieval in a web-scale image database. The visual vocabulary, deep belief network (DBN) graphical model [13] is relatively low cost (e.g. 5.3 MByte) logarithmic to the number of words.

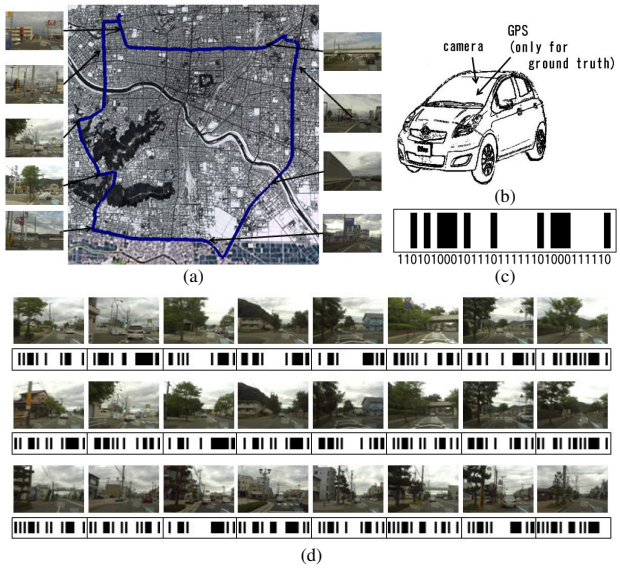

Fig. 1. How well compact binary landmark representation works in mobile robot localization? (a) Experimental environment and robot's trajectory. (b) A high-speed car-like mobile robot. (c) Binary landmark representation using the semantic hashing technique. (d) Sequences of images and binary codes. Top, Middle: Two similar locations. Bottom: A dissimilar location. Note that the codes are similar/dissimilar only at a few bits.

This paper proposes a compact binary code representation by employing the semantic hashing technique, as illustrated in Figure 1. In this study, we are particularly interested in how well such a binary landmark representation works in standard localization frameworks such as particle filter [14]. For instance, a 1-bit binary code is a very compact visual word compared to the cases of other type vocabularies such as PCA (e.g. a 10-dim float vector) [4] or the vocabulary tree (e.g. a 6-dim integer vector) [15]. In fact, individual binary measurements are not sufficiently informative for successful localization. Our approach is to integrate a sequence of binary measurements in a standard framework of incremental localization [14]. We have developed a visual localization

This work was partially supported by MECSST Grant in-Aid for Young Scientists (B) (19700192, 21700221), by KURATA grants and by TATEISI Science And Technology Foundation.

K. Ikeda and K. Tanaka are with Faculty of Engineering, University of Fukui, Japan. tnkknj@u-fukui.ac.jp

system by implementing techniques from optical flow [16] (visual odometry), GIST scene descriptor [6] (visual feature), semantic hashing [11] (visual vocabulary) as well as particle filter [14] (inference framework). In our system, individual bits i of the K-dim vector output by the semantic hashing should be viewed as K independent measurements. In the spirit of sensor fusion [17], we employ K different binary maps and record i-th bit measurements on i-th map. From the viewpoint of information sharing, the number $K' < K$ of binary maps shared and used for localization should be minimized. We experimentally show how many binary maps are actually required for successful localization.

To our knowledge, this is the first study that focuses on binary landmark representation using the semantic hashing technique. In our contribution, we investigate the cost-performance, the semantic gap, the saliency evaluation using the presented techniques, as well as challenge to further reduce resources (#bits) per landmark. Experiments using a high-speed car-like robot show promising results.

A. Relation to other works

Existing techniques for visual vocabulary have their own advantages and disadvantages in terms of the space efficiency of the vocabulary, the words and the database structure, as well as in terms of the time efficiency of database building and retrieval. For example, PCA (principal component analysis) [4] and LSH (locality sensitive hashing) [18] are advantageous in terms of the lightweight vocabulary (e.g. a few MByte) but their word is not compact (e.g. over 10 Byte). On the other hand, the vocabulary tree [15] (hierarchical k-means) is advantageous in terms of compact word (e.g. a few Byte) but its vocabulary is heavy weight (e.g. 1GByte). BOF (bag of features) [19] using such as TF-IDF (term frequency - inverse document frequency) [20] represents an image as a word count vector and could achieve a good trade-off between lightweight vocabulary (depending on the vocabulary used) and compact word. An advantage of our approach is extremely compact word (e.g. 32 bits) as well as relatively lightweight vocabulary (e.g. 5.3 MByte).

Local features such as SIFT (scale-invariant feature transform) are also used in many studies on visual localization [7] [14] [21] as well as in our previous studies [8] [9]. In their approaches, a set of local features are extracted from an image at either interest, random or dense points. In contrast, an advantage of global feature approach is that it naturally captures the semantic information of an image. Such a semantic reasoning in the case of local feature approach is an interesting topic of on-going research [10]. From the viewpoint of compactness, another advantage of global feature is that instead of representing an image by many features, it can represent an image by a single feature.

We use global features extracted by GIST scene descriptor [6]. In [22], GIST scene descriptor has been successful in the context of web-scale image retrieval. However, our approach of binary landmarks is not limited to a specific feature type, but could be applied to other type global features such as HOG (histogram of oriented gradients) [23].

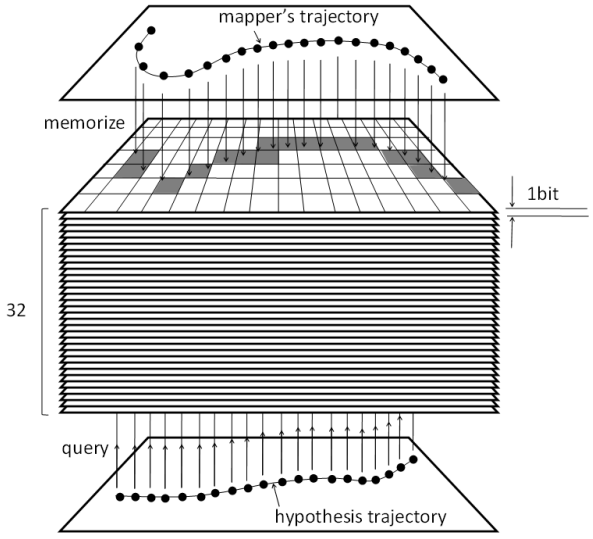

Fig. 2. Binary maps. Individual binary codes are viewed as different type independent measurements. We employ K different binary maps for K bit code and then record i-th bit measurements on i-th map. We also discuss how many $K'(< K)$ binary maps are required for successful localization.

II. Binary landmark approach

In this study, an input measurement of high dimensional visual feature is translated by a visual vocabulary to a visual word of K-bit binary vector:

$$C = [c^1, ..., c^K]. \tag{1}$$

Individual bits c^i of the vector are assumed to be independent binary measurements. Some examples of the translation are shown in Figure 1 (d). In order to deal with K different binary measurements

$$c^1, ..., c^K, \tag{2}$$

we introduce K binary maps

$$B^1, ..., B^K \tag{3}$$

as illustrated in Figure 2 and record i-th bit measurements on i-th map.

The space cost is linear to the number of binary maps we use. Each landmark consumes 1-bit per binary map. As a default setting, we use the full K bit binary maps for mapping and localization. To further enhance the compactness, we will also challenge localization with reduced $K' = K - \Delta K < K$ binary maps in experimental section IV.

Many current localization approaches (e.g. particle filters [14], multiple hypothesis tracking [21]) maintain multiple hypotheses of the 3robot pose. In implementation, the robot pose is represented as a 1DOF viewpoint ID on the map, although the approach could be easily generalized to 3DOF or 6DOF. They generate hypotheses, track each hypothesis as the robot moves, evaluate the likelihood of each hypothesis as the robot observes a landmark, and occasionally generate some new likely hypotheses to replace old ones. Our approach will use binary landmark measurements in the likelihood evaluation as well as the hypothesis generation.

A. Likelihood evaluation

The likelihood evaluation aims to evaluate how likely a robot pose hypothesis x_t is consistent with a binary landmark measurement c^i given a landmark map. It predicts the landmark pose conditioned on the hypothesized robot pose, retrieves corresponding landmark poses in the map, and then computes the observation likelihood according to the degree of similarity between the query and such a corresponding landmark. We obtain the observation likelihood for a binary measurement c^i as

$$P(c^i|x) \simeq \begin{cases} l_o c_L^{1/K} & (c^i = c_{map,x}) \\ l_o & (c^i \neq c_{map,x}) \end{cases}. \quad (4)$$

Increase in the likelihood of relevant/irrelevant hypotheses is controlled by the pre-defined constant c_L.

B. Hypothesis generation

The hypothesis generation aims to generate a new likely hypothesis of robot pose according to the latest landmark measurement. It searches a similar landmark in the landmark database, and generates a new robot pose hypothesis as a prediction from the similar landmark.

We can search similar landmarks in the database by a hash table. A binary code

$$C = [c^1, \cdots, c^K] \quad (5)$$

points an address

$$a = \sum_i^K c^i 2^{i-1} \bmod S \quad (6)$$

in the hash table. The hash table size S is set small (e.g. 8 MByte) considering available memory size. Inserting a novel landmark to the landmark database is an incremental process of inserting the pointer of the landmark to the corresponding bin at the address a. Such an incremental database building is useful and essential for incremental map learning [24]. Searching similar landmarks in the database is a process of accessing the bins in a Hamming ball centered at the address a corresponding to the query landmark and is fast by using a pre-computed lookup table.

The number of similar landmarks returned by above retrieval technique is in proportion to the number of landmarks in the database. This adds computational burden in the case of a large-size map. Usually, we are only interested in a small portion of such similar landmarks, i.e. the ones that are visible from the viewpoints of at least one hypothesis. Considering the fact, we also have developed an alternative technique that simply iteratively samples a new landmark ID until it finds a similar landmark. Finding similar landmarks by bit count operation consumes time proportional to the number of hypotheses and independent from map size.

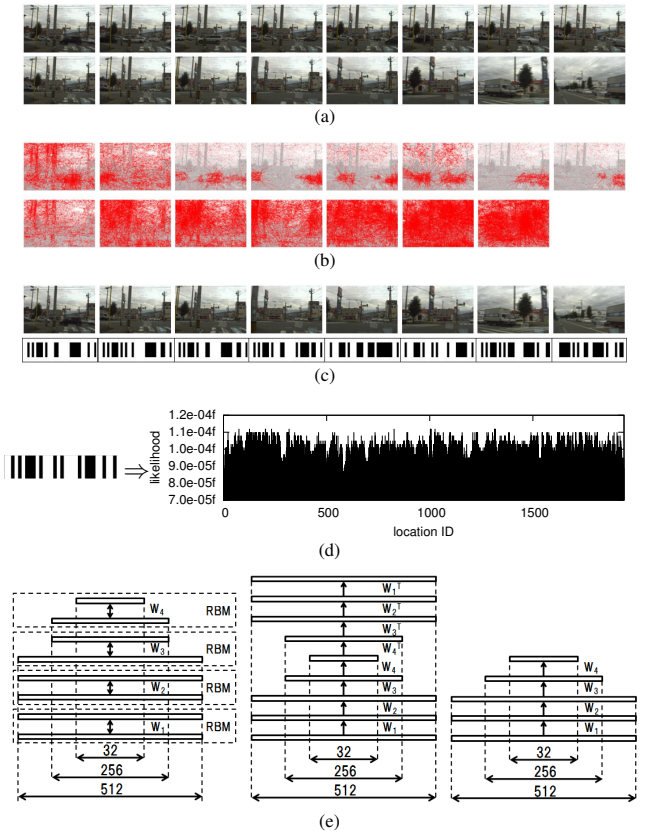

Fig. 3. Visual localization processing. (a) Input images. (b) Optical flows. (c) Binary codes. (d) Observation likelihood over the robot pose space. (e) Semantic hashing architecture. Left: Pre-training. Middle: Fine-tuning. Right: Encoding. The number of nodes for each layer is also shown in the figures.

III. Visual Localization System

This section explains an instance of our visual localization system, which is composed of visual odometry, visual feature, visual vocabulary and inference framework. Figure 3 shows an overview.

A. Visual odometry: optical flow

Visual odometry (e.g. [16]) aims to estimate the egomotion from successive images. Many visual odometry techniques such as monoSLAM [7] rely on local feature tracking technique. Unfortunately, in our application of a high-speed car-like mobile robot, we found that local feature tracking is unstable and often fails due to sparse sensing as well as large vibration of robot's body. Currently, our visual odometry aims to acquire only a simple binary measurement indicating whether the robot is moving (1) or not (0). We found that optical flow [25] is reliable to obtain such a binary measurement. Our method observes length of all the optical flow vectors appearing in the current image pair and if their median length exceeds a preset threshold then decides that the robot is moving. The binary motion measurement is simple, but in practice improves reliability of localization.

B. Visual feature: GIST scene descriptor

The GIST scene descriptor [6] aims to develop a lower dimensional representation of a scene image. It captures the information of perceptual dimensions (naturalness, openness, roughness, expansion, ruggedness) and describes the spatial structure of the scene. The descriptor extracts such a feature vector using spectral and coarsely localized information. It firstly decomposes the image using a set of multi-scale oriented filters, and then averages the output magnitude of each vector over 4x4 grid. The GIST descriptor can be viewed as an extension of local SIFT feature [26] to global feature so as to describe the entire image. We use orientation filters with 8 orientations and 4 scales and as a result, obtain (4x4)x8x4=512 dimensional feature vectors. Comparing GIST features and binary codes respectively are based on L2 distance and Hamming distance.

C. Visual vocabulary: semantic hashing

The semantic hashing [11] aims to learn compact binary codes for image retrieval. It is based on a method for training deep graphical models one layer at a time [13]. It has a network architecture that progressively reduces the number of units in each layer, and by doing so, it maps a high dimensional input vector to a far smaller binary vector (e.g. 32-bit) at the output. The first layer of visible units are modeled to have a Gaussian distribution so as to deal with GIST vectors. The real value at the top layer is binarized by a threshold learned from the entire dataset. Figure 3(e) describes the architecture of the deep graphical model used in this study.

The deep network is trained in two stages: pre-training and fine-tuning. The pre-training stage trains the network from the visible input up to the output layer in a greedy manner using a contrastive divergence. The fine-tuning aims to move weights of the network to local optimum by back-propagation on labeled data. In our current system, both the pre-training and the fine-tuning are slow, offline, batch processing.

Once the learning is complete, mapping from a feature vector to the states of the top-level variables (i.e. binary code) is fast. Mapping for each layer is performed by one matrix multiplication and by component-wise non-linearity. Each bit of the binary code represents a set containing about half the entire image set. Computing K-bit binary code vector is equivalent to a fast intersection of such K sets.

D. Inference framework: particle filter

The particle filter [14] is currently a very popular approach for probabilistic global localization as well as pose tracking. In the following, we summarize the characteristics of our approach. At the beginning of the localization task, it generates a set of robot pose samples called particles. Each sample represents a hypothesis of the robot pose, and the entire sample set represents the current belief of robot pose. During the localization, the particle filter updates the state and the likelihood (or "weight") of each hypothesis. As the robot moves, it performs so-called motion update in order to move the sample's state by simulating the robot motion and random noises. As the robot observes a novel landmark, it performs so-called perception update in order to update the likelihood of each sample according to consistency between the landmark observation and its prediction from the map. When effective sample size [27] becomes lower than a preset threshold, it performs resampling in order to duplicate samples with high weight. Occasionally, it performs sensor resetting [28] so as to replace small portion (in implementation, 1%) of samples with new likely hypotheses according to the latest landmark observation. In above processing, the motion update using motion measurement $z_t = a_t$, the perception update using perception measurement $z_t = c_t^i$ and the sensor resetting using perception measurement $z_t = C_t$ respectively are described as

$$P(x_t|z_{1:t}) = \int P(x_t|x_{t-1})P(x_{t-1}|z_{1:t-1})dx_{t-1}, \quad (7)$$

$$P(x_t|z_{1:t}) \propto P(z_t|x_t)P(x_{t-1}|z_{1:t-1}), \quad (8)$$

$$x_t \sim P(x_t|z_t), \quad (9)$$

and implemented by the techniques described in previous sections III-A, II-A and II-B.

IV. Evaluation experiments

This section investigates several use of the binary landmark maps through experiments. A high-speed car-like robot acquires experimental data in an outdoor environment as shown in Figure 1. Such car robot localization is an important and challenging scenario due to sparse sensing as well as large vibration of the robot's body. The following subsections IV-A, IV-B, IV-C report the basic performance of the proposed techniques by using the full 32 bit binary maps. The subsections IV-D, IV-E report the performance by using the reduced binary maps. The subsection IV-F compares the proposed semantic hashing approach against the LSH approach in our previous paper [8]. The subsection IV-G discusses resources used by our system.

In the experiments, we basically use two outdoor datasets (denoted as "Outdoor"), one for mapping ("mapping") and one for localization ("localization"). For each dataset, our car robot drives 20km trajectories at 0-50km/h, while acquiring images by an on-board front camera at 10fps. GPS data is used for associating each landmark with its pose in mapping task, while used for ground truth in localization task. As a default setting, #samples is 10,000, the parameter $c_L = 2$ and #bits is 32. The vocabulary is learned by using 70K images from LabelMe website ("LabelMe") [29]. The robot and the landmark poses are represented in the 1d location ID space [30]. The entire localization system is implemented in C++ language on a Linux machine with 8 GByte memory. We will see that only a small fraction of the memory is required for the scale of maps considered in the current experiment. This allows plenty of room for scalability.

A. Localization using full 32 binary maps

Figure 4 illustrates typical examples of successful and failure localization tasks. Successful task is defined as such tasks where the localization error finally becomes smaller than 200m, about 1% of the entire trajectory length. It is common to use such final localization error as a measure to quantitatively evaluate the quality of an entire localization task. The length of localization trajectory corresponds to 100 viewpoints. At the beginning of localization tasks, the localization error is very large as the robot pose is completely unknown. In success examples, the error gradually reduces and finally converges to near zero. In these cases, hypotheses and their weights are gradually updated as the robot moves and observes landmarks, and as a result of resampling, the ratio of correct hypotheses increases. In failure examples, we can see two distinguishable cases, one is case where the estimate either does not converge at all or converges to some wrong robot pose, the other is case where the estimate globally converges to the true robot pose but the resulted error exceeds the pre-defined threshold of 200m. In many cases, the primary source of errors is failure in landmark recognition as well as duplicate visual words.

We conducted 100 similar localization tasks iteratively for 100 different robot trajectories randomly sampled from the "localization" dataset. We report those results where estimate by the particle filter converges during the last 20 of the 100 viewpoint trajectories. We denote as success ratio the ratio of successful tasks. Figure 5 summarizes average success ratio over 100 different localization tasks. It can be seen that high success ratio of around 0.7 is obtained when #samples is set 10,000 or larger. We have also tested different setting of the semantic hashing. For instance, Figure 5 reports success ratio against #bits of visual word. It can be seen that stable results are obtained when #bits is set to 32. It could be concluded that localization is mostly successful even though we use compact 32 bit landmarks.

Overall, the proposed approach has proven to be advantageous in the ability of global localization as well as compactness of landmark maps. Of course, there remains room for further improvement. For instance, success ratio of around 0.7 means that the robot still often gets lost. Also, localization error that is smaller than 200m in outdoor is not so accurate, for instance, can easily be achieved with any low-cost GPS receiver. They might raise the question for the practicability of the proposed approach. However, in many cases, we are not obligated to find the best hypotheses with only a single localization system. In recent years, integrating advantages of multiple different localization systems has become quite common. We believe that in such an integrated system, our advantages of global localization as well as compact maps would play an important role.

B. Semantic gap

In general, a pre-learned vocabulary suffers from semantic gap between the dataset used for vocabulary learning (i.e. "LabelMe") and the dataset used for localization (i.e. "Outdoor"). As can be seen from Figure 6, there is a large gap

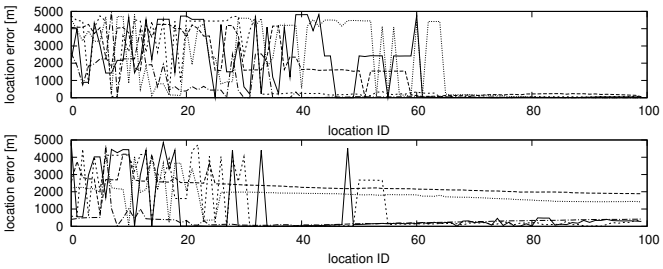

Fig. 4. Localization errors for 5 success examples (top) and for 5 failure examples (bottom), randomly sampled from the 100 localization tasks. Vertical axis: location error [m]. Horizontal axis: location ID.

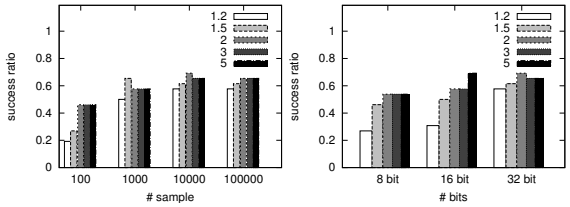

Fig. 5. Success ratio vs. #samples (left) and success ratio vs. #bits (right) over the 100 tasks. Each plot corresponds to each value of parameter c_L.

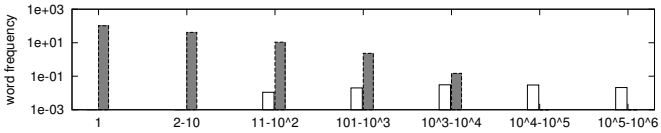

Fig. 6. Word frequency of "Outdoor" dataset (left) vs. "LabelMe" (right).

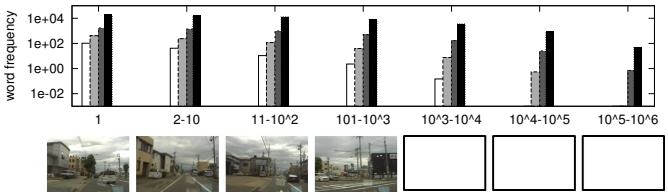

Fig. 7. Frequency of visual words. Top: Average number of landmarks per word. Words are sorted in terms of the number of landmarks they contain and then grouped into 7 groups. Each datapoint from left to right respectively corresponds to the word as well as their near neighbors in terms of Hamming distance 1, 2 and 3. Bottom: Images corresponding to the central words (ID: 1, 5, 50 and 500) of each group.

between the two datasets in terms of word frequency. For instance, none of the top 10 most frequent visual words from "Outdoor" appear in "LabelMe". Bridging the semantic gap using such as incremental learning is our future research [9]. It is noteworthy that our localization system is still successful despite of such a large semantic gap.

C. Saliency evaluation

Saliency of individual landmarks is important especially in the context of landmark selection as well as visual attention. In general, all the visual words are not equally important. In a sense, frequent words tend to be more important. Figure 7 shows the frequency of individual visual words as well as their near neighbors. There is a large difference of frequency between individual visual words. For instance, the top 100 most frequent words are 10-100 times more frequent than the other words. Small number of such frequent words correspond to the most useful landmarks in our system.

D. How many binary maps are necessary?

We now turn to the problem of localization using reduced number of $K' = 32 - \Delta K$ of binary maps, as mentioned in section II. We firstly test a random sampling strategy that randomly selects ΔK binary maps and remove them from the memory (i.e. forgets them). The robot simply ignores those binary code measurements that correspond to the removed ΔK bits. Figure 8 ("random") reports the results with reduced binary map. It can be seen that although the localization is still successful to some extent, success ratio becomes low when ΔK is 12 or larger. It is concluded that with the random sampling strategy, map reduction is not an effective way to achieve high localization performance.

E. How useful each binary map is?

Landmark selection techniques (e.g. [31]) evaluate usefulness of a landmark set through simulated experiences, a set of virtual localization tasks using a validation dataset in a computer simulation environment. We next test a planned sampling strategy that selects binary maps to be removed not randomly but according to the usefulness score evaluated through simulated experiences. For the purpose, we use additional "validation" dataset that corresponds to 20km driving in the same environment and independent from either the "mapping" and the "localization" dataset. The planned sampling strategy evaluates each candidate of the reduced binary map through 100 virtual localization tasks using 100 trajectories randomly sampled from the "validation" dataset. As a result, we obtain a database of simulated experiences. Given such an experience database, it computes the average success ratio over the 100 virtual tasks as the usefulness score, and selects as the best map the one with the highest usefulness score.

Figure 8 ("selection") illustrates the success ratio against the number ΔK of reduced binary maps. In contrast to the random sampling strategy, the decrease of success ratio is slow even when ΔK is 12-15. Moreover, the success ratio when ΔK is 5-15 is even better than the case of full 32 bit binary maps. In this sense, the planned sampling strategy is effective to remove redundant or unnecessary information existing in the original 32 binary maps, as well as to bridge the semantic gap existing in the orginal vocabulary or dialect. Our map selection reduces #bits per landmark. For example, when the number of reduced bits is 12 then each landmark is represented by 32-12=20 bits.

F. Comparison against LSH localization [9]

We also compared the proposed semantic hashing approach against LSH approach presented in our previous paper [9]. As an advantage, LSH vocabulary does not suffer from the semantic gap since it can be learned in online during the mapping task. As a downside, this requires a large amount of memory when the number of landmarks is large. The LSH localization approach used here is almost equivalent to the LSH particle filter [9] except for that it uses global features instead of local features. Figure 8(b) summarizes the comparison results. In the figure, K and L are parameters of LSH and represents the dimensionality and the number of hash tables used in LSH. It can be seen that the semantic hashing localization is comparable to the LSH localization for a wide range of the parameter values. A drawback of LSH is that its visual word is not compact (e.g. 40 Byte) even when K is small. It could be said that the proposed technique achieves extremely compact word while maintaining the efficiency of the localization system.

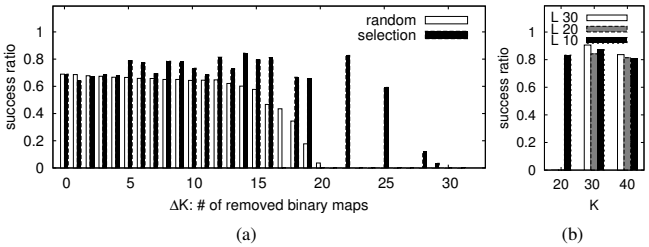

Fig. 8. Localization performance. (a) The semantic hashing localization. Left: The random sampling strategy. Right: The planned sampling strategy. Vertical axis: success ratio. Horizontal axis: the number ΔK of removed binary maps. (b) The LSH-based localization [9]. In some settings (e.g. {ΔK:22,"random"}, {L:20,K:20}), the success ratio is unreliable (due to low convergence rate) and omitted.

G. Computational costs

The resource used by our system mainly consists of the visual vocabulary (DBN) and the visual words. The cost for visual words is proportional to the number of landmarks as well as #bits per landmark. The cost for visual vocabulary is strongly dependent on the cost for DBN and quadratic in the dimensionality of input vector. The above costs respectively are 8 KB in total and 5.3 MB in this experiment. The latter cost is constant and independent of the map size. The time cost is 8.3×10^{-2} sec per viewpoint for the default setting. The result is a quite low cost visual retrieval and could scale to much larger environments and maps.

V. Conclusions & future works

We have studied the mobile robot localization from a perspective of compact binary landmarks. To our knowledge, this is the first study that focuses on binary landmark representation using the semantic hashing technique. The proposed technique maps the input high dimensional visual features to far lower dimensional binary codes and as a result, achieves compact word as well as vocabulary. In the spirit of sensor fusion, our approach treats individual bit measurements as independent measurements and employs K different binary maps. In our contribution, visual robot localization using the full 32 binary maps as well as reduced (e.g. 20-bit) binary maps are successful. In future, we plan to study our approach in the context of long-term multi-robot scenarios such as lifelong map learning [1] as well as information sharing networks [2] where compact landmark representation should play an important role.

REFERENCES

[1] P. Biber and T. Duckett. Dynamic maps for long-term operation of mobile service robots. *Proc. Robotics: Science and Systems I*, 2005.

[2] G. S. Sukhatme and W. Burgard. Robotic sensor networks: Principles and practice. *RSS07 Workshop*, 2007.

[3] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1 – 7, 2007.

[4] N. A. Vlassis, B. Terwijn, and B. ja Kroese. Auxiliary particle filter robot localization from high-dimensional sensor observations. *Proc. IEEE Int. Conf. Robotics and Automation*, pages 7–12, 2002.

[5] J. Hays and A.A. Efros. Im2gps: estimating geographic information from a single image. *IEEE Computer Vision and Pattern Recognition*, 2008.

[6] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. J. computer vision*, 42(3):145–175, 2001.

[7] B. Williams, G. Klein, and I. Reid. Real-time slam relocalisation. *Proc. IEEE 11th Int. Conf. Computer Vision*, pages 1–8, 2007.

[8] K. Tanaka and E. Kondo. A scalable algorithm for monte carlo localization using an incremental e2lsh-database of high dimensional features. *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2784–2791, 2008.

[9] K. Saeki, K. Tanaka, and T. Ueda. Lsh-ransac: An incremental scheme for scalable localization. *Proc. IEEE Int. Conf. Robotics and Automation*, pages 3523–3530, 2009.

[10] P. Newman I. Posner, M. Cummins. Fast probabilistic labeling of city maps. *Proc. Robotics: Science and Systems IV*, 2008.

[11] R. Salakhutdinov and G. Hinton. Semantic hashing. *Int. J. Approximate Reasoning*, 2008.

[12] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2008.

[13] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504–507, 2006.

[14] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1322–1328, 1999.

[15] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

[16] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *J. Field Robotics*, 23(1):3–20, 2006.

[17] M. Rosencrantz, G. Gordon, and S. Thrun. Decentralized sensor fusion with distributed particle filters. *Proc. UAI*, 2003.

[18] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. *Proc. Very Large Database Conference*, 1999.

[19] G. Csurka, C. R. D. L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. *ECCV2004 workshops on Statistical Learning in Computer Vison*, 2004.

[20] Salton. Developments in automatic text retrieval. *Science*, page 253, 1991.

[21] P. Jensfelt and S. Kristensen. Active global localisation for a mobile robot using multiple hypothesis tracking. *Trans. IEEE Robotics and Automation*, pages 13–22, 1999.

[22] M. Douze, H. Jégou, H. Singh, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. *Proc. Int. Conf. Image and Video Retrieval*, 2009.

[23] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Proc. IEEE Int. Conf. Conputer Vision and Pattern Recognition*, pages 886–893, 2005.

[24] M. Montemerlo. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Carnegie Mellon University, 2003.

[25] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[26] D. G. Lowe. Object recognition from local scale-invariant features. *Proc. Int. Conf. Computer Vision*, pages 1150–1157, 1999.

[27] A. Doucet, N. Freitas, and N. Gordon editors. Sequential monte carlo methods in practice. *Statistics for engineering and information science*, 2001.

[28] S. Lenser and M. Velose. Sensor resetting localization for poorly modeled mobile robots. *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1225–1232, 2002.

[29] B. Russell, A. Torralba, and W. T. Freeman. Labelme: The open annotation tool. *http://labelme.csail.mit.edu/*.

[30] A. Angeli, S. Doncieux, J. A. Meyer, and D. Filliat. Real-time visual loop-closure detection. *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1842–1847, 2008.

[31] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson. Landmark selection for vision-based navigation. *Trans. IEEE Robotics*, 22:334–349, 2006.